

new/usr/src/pkg/manifests/system-header.mf

1

```
*****
89662 Tue Nov 20 12:08:34 2018
new/usr/src/pkg/manifests/system-header.mf
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2012 by Delphix. All rights reserved.
25 # Copyright 2013 Saso Kiselkov. All rights reserved.
26 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
27 # Copyright 2018 Nexenta Systems, Inc.
28 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
29 #
30 #
31 set name=pkg.fmri value=pkg:/system/header@$(PKGVERS)
32 set name=pkg.description \
33     value="SunOS C/C++ header files for general development of software"
34 set name=pkg.summary value="SunOS Header Files"
35 set name=info.classification value=org.opensolaris.category.2008:System/Core
36 set name=variant.arch value=$(ARCH)
37 dir path=usr group=sys
38 dir path=usr/include
39 $(i386_ONLY)dir path=usr/include/$(ARCH64)
40 $(i386_ONLY)dir path=usr/include/$(ARCH64)/sys
41 dir path=usr/include/ads
42 dir path=usr/include/arpa
43 dir path=usr/include/asm
44 dir path=usr/include/ast
45 dir path=usr/include/bsm
46 dir path=usr/include/dat
47 dir path=usr/include/des
48 dir path=usr/include/gssapi
49 dir path=usr/include/hal
50 $(i386_ONLY)dir path=usr/include/ia32
51 $(i386_ONLY)dir path=usr/include/ia32/sys
52 dir path=usr/include/inet
53 dir path=usr/include/inet/kssl
54 dir path=usr/include/ipp
55 dir path=usr/include/ipp/ippgc
56 dir path=usr/include/iso
57 dir path=usr/include/kerberosv5
58 dir path=usr/include/libpolkit
59 dir path=usr/include/net
60 dir path=usr/include/netinet
```

new/usr/src/pkg/manifests/system-header.mf

2

```
61 dir path=usr/include/nfs
62 dir path=usr/include/protocols
63 dir path=usr/include/rpc
64 dir path=usr/include/rpcsvc
65 dir path=usr/include/sasl
66 dir path=usr/include/scsi
67 dir path=usr/include/scsi/plugins
68 dir path=usr/include/scsi/plugins/ses
69 dir path=usr/include/scsi/plugins/ses/framework
70 dir path=usr/include/scsi/plugins/ses/vendor
71 dir path=usr/include/scsi/plugins/smp
72 dir path=usr/include/scsi/plugins/smp/engine
73 dir path=usr/include/scsi/plugins/smp/framework
74 dir path=usr/include/security
75 dir path=usr/include/sharefs
76 dir path=usr/include/sys
77 dir path=usr/include/sys/av
78 dir path=usr/include/sys/contract
79 dir path=usr/include/sys/crypto
80 dir path=usr/include/sys/dktp
81 dir path=usr/include/sys/fc4
82 dir path=usr/include/sys/fm
83 dir path=usr/include/sys/fm/cpu
84 dir path=usr/include/sys/fm/fs
85 dir path=usr/include/sys/fm/io
86 $(sparc_ONLY)dir path=usr/include/sys/fpu
87 dir path=usr/include/sys/fs
88 dir path=usr/include/sys/hotplug
89 dir path=usr/include/sys/hotplug/pci
90 dir path=usr/include/sys/ib
91 dir path=usr/include/sys/ib/adapters
92 dir path=usr/include/sys/ib/adapters/hermon
93 dir path=usr/include/sys/ib/adapters/tavor
94 dir path=usr/include/sys/ib/clients
95 dir path=usr/include/sys/ib/clients/ibd
96 dir path=usr/include/sys/ib/clients/of
97 dir path=usr/include/sys/ib/clients/of/rdma
98 dir path=usr/include/sys/ib/clients/of/sol_ofs
99 dir path=usr/include/sys/ib/clients/of/sol_ucma
100 dir path=usr/include/sys/ib/clients/of/sol_umad
101 dir path=usr/include/sys/ib/clients/of/sol_uverbs
102 dir path=usr/include/sys/ib/ibnex
103 dir path=usr/include/sys/ib/ibt1
104 dir path=usr/include/sys/ib/ibt1/impl
105 dir path=usr/include/sys/ib/mgt
106 dir path=usr/include/sys/ib/mgt/ibmf
107 dir path=usr/include/sys/iso
108 dir path=usr/include/sys/proc
109 dir path=usr/include/sys/rsm
110 $(i386_ONLY)dir path=usr/include/sys/sata group=sys
111 dir path=usr/include/sys/scsi
112 dir path=usr/include/sys/scsi/adapters
113 dir path=usr/include/sys/scsi/conf
114 dir path=usr/include/sys/scsi/generic
115 dir path=usr/include/sys/scsi/impl
116 dir path=usr/include/sys/scsi/targets
117 dir path=usr/include/sys/sysevent
118 dir path=usr/include/sys/tsol
119 dir path=usr/include/tsol
120 dir path=usr/include/uuid
121 $(sparc_ONLY)dir path=usr/include/v7
122 $(sparc_ONLY)dir path=usr/include/v7/sys
123 $(sparc_ONLY)dir path=usr/include/v9
124 $(sparc_ONLY)dir path=usr/include/v9/sys
125 dir path=usr/include/vm
126 dir path=usr/platform group=sys
```

```

127 $(sparc_ONLY)dir path=usr/platform/SUNW,A70 group=sys
128 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300 group=sys
129 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP2300/include
130 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010 group=sys
131 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-CP3010/include
132 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T12 group=sys
133 $(sparc_ONLY)dir path=usr/platform/SUNW,Netra-T4 group=sys
134 $(sparc_ONLY)dir path=usr/platform/SUNW,SPARC-Enterprise group=sys
135 $(sparc_ONLY)dir path=usr/platform/SUNW,Serverbladel group=sys
136 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-100 group=sys
137 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1000 group=sys
138 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-1500 group=sys
139 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Blade-2500 group=sys
140 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire group=sys
141 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-15000 group=sys
142 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-280R group=sys
143 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-480R group=sys
144 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-880 group=sys
145 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V215 group=sys
146 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V240 group=sys
147 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V250 group=sys
148 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V440 group=sys
149 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V445 group=sys
150 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V490 group=sys
151 $(sparc_ONLY)dir path=usr/platform/SUNW,Sun-Fire-V890 group=sys
152 $(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-2 group=sys
153 $(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-250 group=sys
154 $(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-4 group=sys
155 $(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise group=sys
156 $(sparc_ONLY)dir path=usr/platform/SUNW,Ultra-Enterprise-10000 group=sys
157 $(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40 group=sys
158 $(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60 group=sys
159 $(sparc_ONLY)dir path=usr/platform/SUNW,UltraSPARC-IIi-Netract group=sys
160 $(i386_ONLY)dir path=usr/platform/i86pc group=sys
161 $(i386_ONLY)dir path=usr/platform/i86pc/include
162 $(i386_ONLY)dir path=usr/platform/i86pc/include/sys
163 $(i386_ONLY)dir path=usr/platform/i86pc/include/vm
164 $(i386_ONLY)dir path=usr/platform/i86xpv group=sys
165 $(i386_ONLY)dir path=usr/platform/i86xpv/include
166 $(i386_ONLY)dir path=usr/platform/i86xpv/include/sys
167 $(i386_ONLY)dir path=usr/platform/i86xpv/include/vm
168 $(sparc_ONLY)dir path=usr/platform/sun4u group=sys
169 $(sparc_ONLY)dir path=usr/platform/sun4u/include
170 $(sparc_ONLY)dir path=usr/platform/sun4u/include/sys
171 $(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c
172 $(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/clients
173 $(sparc_ONLY)dir path=usr/platform/sun4u/include/sys/i2c/misc
174 $(sparc_ONLY)dir path=usr/platform/sun4u/include/vm
175 $(sparc_ONLY)dir path=usr/platform/sun4v group=sys
176 $(sparc_ONLY)dir path=usr/platform/sun4v/include
177 $(sparc_ONLY)dir path=usr/platform/sun4v/include/sys
178 $(sparc_ONLY)dir path=usr/platform/sun4v/include/vm
179 dir path=usr/share
180 dir path=usr/share/man
181 dir path=usr/share/man/man3head
182 dir path=usr/share/man/man4
183 dir path=usr/share/man/man5
184 dir path=usr/share/man/man7i
185 dir path=usr/share/src group=sys
186 dir path=usr/share/src/uts
187 $(i386_ONLY)dir path=usr/share/src/uts/i86pc
188 $(i386_ONLY)dir path=usr/share/src/uts/i86xpv
189 $(sparc_ONLY)dir path=usr/share/src/uts/sun4u
190 $(sparc_ONLY)dir path=usr/share/src/uts/sun4v
191 dir path=usr/xpg4
192 dir path=usr/xpg4/include

```

```

193 $(i386_ONLY)file path=usr/include/$(ARCH64)/sys/kdi_regs.h
194 $(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privmregs.h
195 $(i386_ONLY)file path=usr/include/$(ARCH64)/sys/privregs.h
196 file path=usr/include/ads/dsgetdc.h
197 file path=usr/include/aio.h
198 file path=usr/include/alloca.h
199 file path=usr/include/apprtrace.h
200 file path=usr/include/apprtrace_impl.h
201 file path=usr/include/ar.h
202 file path=usr/include/archives.h
203 file path=usr/include/arpa/ftp.h
204 file path=usr/include/arpa/inet.h
205 file path=usr/include/arpa/nameser.h
206 file path=usr/include/arpa/nameser_compat.h
207 file path=usr/include/arpa/telnet.h
208 file path=usr/include/arpa/tftp.h
209 $(i386_ONLY)file path=usr/include/asm/atomic.h
210 $(i386_ONLY)file path=usr/include/asm/bitmap.h
211 $(i386_ONLY)file path=usr/include/asm/byteorder.h
212 $(i386_ONLY)file path=usr/include/asm/clock.h
213 $(i386_ONLY)file path=usr/include/asm/cpu.h
214 $(i386_ONLY)file path=usr/include/asm/cpudev.h
215 $(sparc_ONLY)file path=usr/include/asm/flush.h
216 $(i386_ONLY)file path=usr/include/asm/htable.h
217 $(i386_ONLY)file path=usr/include/asm/mmu.h
218 file path=usr/include/asm/sunddi.h
219 file path=usr/include/asm/thread.h
220 file path=usr/include/assert.h
221 file path=usr/include/ast/align.h
222 file path=usr/include/ast/ast.h
223 file path=usr/include/ast/ast_botch.h
224 file path=usr/include/ast/ast_ccode.h
225 file path=usr/include/ast/ast_common.h
226 file path=usr/include/ast/ast_dir.h
227 file path=usr/include/ast/ast_dirent.h
228 file path=usr/include/ast/ast_fcntl.h
229 file path=usr/include/ast/ast_float.h
230 file path=usr/include/ast/ast_fs.h
231 file path=usr/include/ast/ast_getopt.h
232 file path=usr/include/ast/ast_icnv.h
233 file path=usr/include/ast/ast_lib.h
234 file path=usr/include/ast/ast_limits.h
235 file path=usr/include/ast/ast_map.h
236 file path=usr/include/ast/ast_mmap.h
237 file path=usr/include/ast/ast_mode.h
238 file path=usr/include/ast/ast_namval.h
239 file path=usr/include/ast/ast_ndbm.h
240 file path=usr/include/ast/ast_nl_types.h
241 file path=usr/include/ast/ast_param.h
242 file path=usr/include/ast/ast_standards.h
243 file path=usr/include/ast/ast_std.h
244 file path=usr/include/ast/ast_stdio.h
245 file path=usr/include/ast/ast_sys.h
246 file path=usr/include/ast/ast_time.h
247 file path=usr/include/ast/ast_tty.h
248 file path=usr/include/ast/ast_version.h
249 file path=usr/include/ast/ast_vfork.h
250 file path=usr/include/ast/ast_wait.h
251 file path=usr/include/ast/ast_wchar.h
252 file path=usr/include/ast/ast_windows.h
253 file path=usr/include/ast/bytesex.h
254 file path=usr/include/ast/ccode.h
255 file path=usr/include/ast/cdt.h
256 file path=usr/include/ast/cmd.h
257 file path=usr/include/ast/cmdext.h
258 file path=usr/include/ast/debug.h

```

259 file path=usr/include/ast/dirent.h
260 file path=usr/include/ast/dlldefs.h
261 file path=usr/include/ast/dt.h
262 file path=usr/include/ast/ndian.h
263 file path=usr/include/ast/error.h
264 file path=usr/include/ast/find.h
265 file path=usr/include/ast/fnmatch.h
266 file path=usr/include/ast/fnv.h
267 file path=usr/include/ast/fs3d.h
268 file path=usr/include/ast/fts.h
269 file path=usr/include/ast/ftw.h
270 file path=usr/include/ast/ftwalk.h
271 file path=usr/include/ast/getopt.h
272 file path=usr/include/ast/glob.h
273 file path=usr/include/ast/hash.h
274 file path=usr/include/ast/hashkey.h
275 file path=usr/include/ast/hashpart.h
276 file path=usr/include/ast/history.h
277 file path=usr/include/ast/iconv.h
278 file path=usr/include/ast/ip6.h
279 file path=usr/include/ast/lc.h
280 file path=usr/include/ast/ls.h
281 file path=usr/include/ast/magic.h
282 file path=usr/include/ast/magicid.h
283 file path=usr/include/ast/mc.h
284 file path=usr/include/ast/mime.h
285 file path=usr/include/ast/mnt.h
286 file path=usr/include/ast/modecanon.h
287 file path=usr/include/ast/modex.h
288 file path=usr/include/ast/namval.h
289 file path=usr/include/ast/nl_types.h
290 file path=usr/include/ast/nval.h
291 file path=usr/include/ast/option.h
292 file path=usr/include/ast/preroot.h
293 file path=usr/include/ast/proc.h
294 file path=usr/include/ast/prototyped.h
295 file path=usr/include/ast/re_comp.h
296 file path=usr/include/ast/recfmt.h
297 file path=usr/include/ast/regex.h
298 file path=usr/include/ast/regexp.h
299 file path=usr/include/ast/sfdisc.h
300 file path=usr/include/ast/sfio.h
301 file path=usr/include/ast/sfio_s.h
302 file path=usr/include/ast/sfio_t.h
303 file path=usr/include/ast/shcmd.h
304 file path=usr/include/ast/shell.h
305 file path=usr/include/ast/sig.h
306 file path=usr/include/ast/stack.h
307 file path=usr/include/ast/stak.h
308 file path=usr/include/ast/stdio.h
309 file path=usr/include/ast/stk.h
310 file path=usr/include/ast/sum.h
311 file path=usr/include/ast/swap.h
312 file path=usr/include/ast/tar.h
313 file path=usr/include/ast/times.h
314 file path=usr/include/ast/tm.h
315 file path=usr/include/ast/tmx.h
316 file path=usr/include/ast/tok.h
317 file path=usr/include/ast/tv.h
318 file path=usr/include/ast/usage.h
319 file path=usr/include/ast/vdb.h
320 file path=usr/include/ast/vecargs.h
321 file path=usr/include/ast/vmalloc.h
322 file path=usr/include/ast/wait.h
323 file path=usr/include/ast/wchar.h
324 file path=usr/include/ast/wordexp.h

325 file path=usr/include/atomic.h
326 file path=usr/include/attr.h
327 file path=usr/include/auth_attr.h
328 file path=usr/include/bsm/adt.h
329 file path=usr/include/bsm/adt_event.h
330 file path=usr/include/bsm/audit.h
331 file path=usr/include/bsm/audit_kernel.h
332 file path=usr/include/bsm/audit_kevents.h
333 file path=usr/include/bsm/audit_record.h
334 file path=usr/include/bsm/audit_uevents.h
335 file path=usr/include/bsm/devices.h
336 file path=usr/include/bsm/libbsm.h
337 file path=usr/include/config_admin.h
338 file path=usr/include/cpio.h
339 file path=usr/include/crypt.h
340 file path=usr/include/cryptoutil.h
341 file path=usr/include/ctype.h
342 file path=usr/include/curses.h
343 file path=usr/include/dat/dat.h
344 file path=usr/include/dat/dat_error.h
345 file path=usr/include/dat/dat_platform_specific.h
346 file path=usr/include/dat/dat_redirection.h
347 file path=usr/include/dat/dat_registry.h
348 file path=usr/include/dat/dat_vendor_specific.h
349 file path=usr/include/dat/udat.h
350 file path=usr/include/dat/udat_config.h
351 file path=usr/include/dat/udat_redirection.h
352 file path=usr/include/dat/udat_vendor_specific.h
353 file path=usr/include/default.h
354 file path=usr/include/des/des.h
355 file path=usr/include/des/desdata.h
356 file path=usr/include/des/softdes.h
357 file path=usr/include/devfsadm.h
358 file path=usr/include/device_info.h
359 file path=usr/include/devid.h
360 file path=usr/include/devmgmt.h
361 file path=usr/include/devpoll.h
362 file path=usr/include/dial.h
363 file path=usr/include/dirent.h
364 file path=usr/include/dlfcn.h
365 file path=usr/include/door.h
366 file path=usr/include/elf.h
367 file path=usr/include/ndian.h
368 file path=usr/include/err.h
369 file path=usr/include/errno.h
370 file path=usr/include/eti.h
371 file path=usr/include/euc.h
372 file path=usr/include/exacct.h
373 file path=usr/include/exacct_impl.h
374 file path=usr/include/exec_attr.h
375 file path=usr/include/execinfo.h
376 file path=usr/include/fatal.h
377 file path=usr/include/fcntl.h
378 file path=usr/include/float.h
379 file path=usr/include/fmtmsg.h
380 file path=usr/include/fnmatch.h
381 file path=usr/include/form.h
382 file path=usr/include/fts.h
383 file path=usr/include/ftw.h
384 file path=usr/include/gelf.h
385 file path=usr/include/getopt.h
386 file path=usr/include/getwidth.h
387 file path=usr/include/glob.h
388 file path=usr/include/grp.h
389 file path=usr/include/gssapi/gssapi.h
390 file path=usr/include/gssapi/gssapi_ext.h

```
391 file path=usr/include/hal/libhal-storage.h
392 file path=usr/include/hal/libhal.h
393 $(i386_ONLY)file path=usr/include/ia32/sys/asm_linkage.h
394 $(i386_ONLY)file path=usr/include/ia32/sys/kdi_regs.h
395 $(i386_ONLY)file path=usr/include/ia32/sys/machtypes.h
396 $(i386_ONLY)file path=usr/include/ia32/sys/privregs.h
397 $(i386_ONLY)file path=usr/include/ia32/sys/privregs.h
398 $(i386_ONLY)file path=usr/include/ia32/sys/psw.h
399 $(i386_ONLY)file path=usr/include/ia32/sys/pte.h
400 $(i386_ONLY)file path=usr/include/ia32/sys/reg.h
401 $(i386_ONLY)file path=usr/include/ia32/sys/stack.h
402 $(i386_ONLY)file path=usr/include/ia32/sys/trap.h
403 $(i386_ONLY)file path=usr/include/ia32/sys/traptrace.h
404 file path=usr/include/iconv.h
405 file path=usr/include/idmap.h
406 file path=usr/include/ieeefp.h
407 file path=usr/include/ifaddrs.h
408 file path=usr/include/inet/arp.h
409 file path=usr/include/inet/common.h
410 file path=usr/include/inet/ip.h
411 file path=usr/include/inet/ip6.h
412 file path=usr/include/inet/ip6_asp.h
413 file path=usr/include/inet/ip_arp.h
414 file path=usr/include/inet/ip_ftable.h
415 file path=usr/include/inet/ip_if.h
416 file path=usr/include/inet/ip_ire.h
417 file path=usr/include/inet/ip_multi.h
418 file path=usr/include/inet/ip_netinfo.h
419 file path=usr/include/inet/ip_rts.h
420 file path=usr/include/inet/ip_stack.h
421 file path=usr/include/inet/ipclassifier.h
422 file path=usr/include/inet/ipdrop.h
423 file path=usr/include/inet/ipnet.h
424 file path=usr/include/inet/ipp_common.h
425 file path=usr/include/inet/kssl/ksslapi.h
426 file path=usr/include/inet/led.h
427 file path=usr/include/inet/mi.h
428 file path=usr/include/inet/mib2.h
429 file path=usr/include/inet/nd.h
430 file path=usr/include/inet/optcom.h
431 file path=usr/include/inet/sctp_itf.h
432 file path=usr/include/inet/snmpcom.h
433 file path=usr/include/inet/tcp.h
434 file path=usr/include/inet/tcp_sack.h
435 file path=usr/include/inet/tcp_stack.h
436 file path=usr/include/inet/tcp_stats.h
437 file path=usr/include/inet/tunables.h
438 file path=usr/include/inet/wifi_ioctl.h
439 file path=usr/include/inttypes.h
440 file path=usr/include/ipmp.h
441 file path=usr/include/ipmp_admin.h
442 file path=usr/include/ipmp_mpathd.h
443 file path=usr/include/ipmp_query.h
444 file path=usr/include/ipp/ipgpc/ipgpc.h
445 file path=usr/include/ipp/ipp.h
446 file path=usr/include/ipp/ipp_config.h
447 file path=usr/include/ipp/ipp_impl.h
448 file path=usr/include/ipp/ippctl.h
449 file path=usr/include/iso/ctype_iso.h
450 file path=usr/include/iso/limits_iso.h
451 file path=usr/include/iso/locale_iso.h
452 file path=usr/include/iso/setjmp_iso.h
453 file path=usr/include/iso/signal_iso.h
454 file path=usr/include/iso/stdarg_c99.h
455 file path=usr/include/iso/stdarg_iso.h
456 file path=usr/include/iso/stddef_iso.h
```

```
457 file path=usr/include/iso/stdio_c99.h
458 file path=usr/include/iso/stdio_iso.h
459 file path=usr/include/iso/stdlib_c11.h
460 file path=usr/include/iso/stdlib_c99.h
461 file path=usr/include/iso/stdlib_iso.h
462 file path=usr/include/iso/string_iso.h
463 file path=usr/include/iso/time_iso.h
464 file path=usr/include/iso/wchar_c99.h
465 file path=usr/include/iso/wchar_iso.h
466 file path=usr/include/iso/wctype_iso.h
467 file path=usr/include/iso646.h
468 file path=usr/include/kerberosv5/com_err.h
469 file path=usr/include/kerberosv5/krb5.h
470 file path=usr/include/kerberosv5/locate_plugin.h
471 file path=usr/include/kerberosv5/mit-sipb-copyright.h
472 file path=usr/include/kerberosv5/mit_copyright.h
473 file path=usr/include/klpd.h
474 file path=usr/include/kmfapi.h
475 file path=usr/include/kmftypes.h
476 file path=usr/include/kstat.h
477 file path=usr/include/kvm.h
478 file path=usr/include/langinfo.h
479 file path=usr/include/lastlog.h
480 file path=usr/include/lber.h
481 file path=usr/include/ldap.h
482 file path=usr/include/libcontract.h
483 file path=usr/include/libctf.h
484 file path=usr/include/libdevice.h
485 file path=usr/include/libdevinfo.h
486 file path=usr/include/libdladm.h
487 file path=usr/include/libdlbridge.h
488 file path=usr/include/libdlib.h
489 file path=usr/include/libdllink.h
490 file path=usr/include/libdmpi.h
491 file path=usr/include/libdvlvlan.h
492 file path=usr/include/libelf.h
493 $(i386_ONLY)file path=usr/include/libfdisk.h
494 file path=usr/include/libfstyp.h
495 file path=usr/include/libfstyp_module.h
496 file path=usr/include/libgen.h
497 file path=usr/include/libgrubmgmt.h
498 file path=usr/include/libintl.h
499 file path=usr/include/libipmi.h
500 file path=usr/include/libipp.h
501 file path=usr/include/libnvpair.h
502 file path=usr/include/libnwam.h
503 file path=usr/include/libpolkit/libpolkit.h
504 file path=usr/include/libproc.h
505 file path=usr/include/librcm.h
506 file path=usr/include/libscf.h
507 file path=usr/include/libscf_priv.h
508 file path=usr/include/libshare.h
509 file path=usr/include/libsysevent.h
510 file path=usr/include/libsysevent_impl.h
511 file path=usr/include/libtstnet.h
512 $(sparc_ONLY)file path=usr/include/libv12n.h
513 file path=usr/include/libw.h
514 file path=usr/include/libzfs.h
515 file path=usr/include/libzfs_core.h
516 file path=usr/include/libzoneinfo.h
517 file path=usr/include/limits.h
518 file path=usr/include/linenum.h
519 file path=usr/include/link.h
520 file path=usr/include/listen.h
521 file path=usr/include/locale.h
522 file path=usr/include/macros.h
```

```
523 file path=usr/include/maillock.h
524 file path=usr/include/malloc.h
525 file path=usr/include/md4.h
526 file path=usr/include/md5.h
527 file path=usr/include/memory.h
528 file path=usr/include/menu.h
529 file path=usr/include/mon.h
530 file path=usr/include/monetary.h
531 file path=usr/include/mp.h
532 file path=usr/include/mqueue.h
533 file path=usr/include/mtmalloc.h
534 file path=usr/include/nan.h
535 file path=usr/include/ndbm.h
536 file path=usr/include/ndpd.h
537 file path=usr/include/net/af.h
538 file path=usr/include/net/bridge.h
539 file path=usr/include/net/if.h
540 file path=usr/include/net/if_arp.h
541 file path=usr/include/net/if_dl.h
542 file path=usr/include/net/if_types.h
543 file path=usr/include/net/pfkeyv2.h
544 file path=usr/include/net/pfpolicy.h
545 file path=usr/include/net/ppp-comp.h
546 file path=usr/include/net/ppp_defs.h
547 file path=usr/include/net/pppio.h
548 file path=usr/include/net/radix.h
549 file path=usr/include/net/route.h
550 file path=usr/include/net/trill.h
551 file path=usr/include/net/vjcompress.h
552 file path=usr/include/netconfig.h
553 file path=usr/include/netdb.h
554 file path=usr/include/netdir.h
555 file path=usr/include/netinet/arp.h
556 file path=usr/include/netinet/dhcp.h
557 file path=usr/include/netinet/dhcp6.h
558 file path=usr/include/netinet/icmp6.h
559 file path=usr/include/netinet/icmp_var.h
560 file path=usr/include/netinet/if_ether.h
561 file path=usr/include/netinet/igmp.h
562 file path=usr/include/netinet/igmp_var.h
563 file path=usr/include/netinet/in.h
564 file path=usr/include/netinet/in_pcb.h
565 file path=usr/include/netinet/in_system.h
566 file path=usr/include/netinet/in_var.h
567 file path=usr/include/netinet/ip.h
568 file path=usr/include/netinet/ip6.h
569 file path=usr/include/netinet/ip_icmp.h
570 file path=usr/include/netinet/ip_mroute.h
571 file path=usr/include/netinet/ip_var.h
572 file path=usr/include/netinet/pim.h
573 file path=usr/include/netinet/sctp.h
574 file path=usr/include/netinet/tcp.h
575 file path=usr/include/netinet/tcp_debug.h
576 file path=usr/include/netinet/tcp_fsm.h
577 file path=usr/include/netinet/tcp_seq.h
578 file path=usr/include/netinet/tcp_timer.h
579 file path=usr/include/netinet/tcp_var.h
580 file path=usr/include/netinet/tcpip.h
581 file path=usr/include/netinet/udp.h
582 file path=usr/include/netinet/udp_var.h
583 file path=usr/include/netinet/vrrp.h
584 file path=usr/include/nfs/auth.h
585 file path=usr/include/nfs/export.h
586 file path=usr/include/nfs/lm.h
587 file path=usr/include/nfs/mapid.h
588 file path=usr/include/nfs/mount.h
```

```
589 file path=usr/include/nfs/nfs.h
590 file path=usr/include/nfs/nfs4.h
591 file path=usr/include/nfs/nfs4_attr.h
592 file path=usr/include/nfs/nfs4_clnt.h
593 file path=usr/include/nfs/nfs4_db_impl.h
594 file path=usr/include/nfs/nfs4_idmap_impl.h
595 file path=usr/include/nfs/nfs4_kprot.h
596 file path=usr/include/nfs/nfs_acl.h
597 file path=usr/include/nfs/nfs_clnt.h
598 file path=usr/include/nfs/nfs_cmd.h
599 file path=usr/include/nfs/nfs_log.h
600 file path=usr/include/nfs/nfs_sec.h
601 file path=usr/include/nfs/nfsid_map.h
602 file path=usr/include/nfs/nfssys.h
603 file path=usr/include/nfs/rnode.h
604 file path=usr/include/nfs/rnode4.h
605 file path=usr/include/nl_types.h
606 file path=usr/include/nlist.h
607 file path=usr/include/note.h
608 file path=usr/include/nss_common.h
609 file path=usr/include/nss_dbdefs.h
610 file path=usr/include/nss_netdir.h
611 file path=usr/include/nsswitch.h
612 file path=usr/include/panel.h
613 file path=usr/include/paths.h
614 file path=usr/include/pcsample.h
615 file path=usr/include/pfmt.h
616 file path=usr/include/pkgdev.h
617 file path=usr/include/pkginfo.h
618 file path=usr/include/pkglocs.h
619 file path=usr/include/pkgstrct.h
620 file path=usr/include/pkgtrans.h
621 file path=usr/include/poll.h
622 file path=usr/include/port.h
623 file path=usr/include/priv.h
624 file path=usr/include/proc_service.h
625 file path=usr/include/procfs.h
626 file path=usr/include/prof.h
627 file path=usr/include/prof_attr.h
628 file path=usr/include/project.h
629 file path=usr/include/protocols/dumppstore.h
630 file path=usr/include/protocols/routed.h
631 file path=usr/include/protocols/rwhod.h
632 file path=usr/include/protocols/timed.h
633 file path=usr/include/pthread.h
634 file path=usr/include/pw.h
635 file path=usr/include/pwd.h
636 file path=usr/include/rcm_module.h
637 file path=usr/include/rctl.h
638 file path=usr/include/re_comp.h
639 file path=usr/include/regex.h
640 file path=usr/include/regex.h
641 file path=usr/include/regex.h
642 file path=usr/include/resolv.h
643 file path=usr/include/rje.h
644 file path=usr/include/rp_plugin.h
645 file path=usr/include/rpc/auth.h
646 file path=usr/include/rpc/auth_des.h
647 file path=usr/include/rpc/auth_sys.h
648 file path=usr/include/rpc/auth_unix.h
649 file path=usr/include/rpc/bootparam.h
650 file path=usr/include/rpc/clnt.h
651 file path=usr/include/rpc/clnt_soc.h
652 file path=usr/include/rpc/clnt_stat.h
653 file path=usr/include/rpc/des_crypt.h
654 file path=usr/include/rpc/ib.h
```

655 file path=usr/include/rpc/key_prot.h
 656 file path=usr/include/rpc/nettype.h
 657 file path=usr/include/rpc/pmap_clnt.h
 658 file path=usr/include/rpc/pmap_prot.h
 659 file path=usr/include/rpc/pmap_prot.x
 660 file path=usr/include/rpc/pmap_rmt.h
 661 file path=usr/include/rpc/rpc.h
 662 file path=usr/include/rpc/rpc_com.h
 663 file path=usr/include/rpc/rpc_msg.h
 664 file path=usr/include/rpc/rpc_rdma.h
 665 file path=usr/include/rpc/rpc_sztypes.h
 666 file path=usr/include/rpc/rpcb_clnt.h
 667 file path=usr/include/rpc/rpcb_prot.h
 668 file path=usr/include/rpc/rpcb_prot.x
 669 file path=usr/include/rpc/rpcent.h
 670 file path=usr/include/rpc/rpcsec_gss.h
 671 file path=usr/include/rpc/rpcsys.h
 672 file path=usr/include/rpc/svc.h
 673 file path=usr/include/rpc/svc_auth.h
 674 file path=usr/include/rpc/svc_mt.h
 675 file path=usr/include/rpc/svc_soc.h
 676 file path=usr/include/rpc/types.h
 677 file path=usr/include/rpc/xdr.h
 678 file path=usr/include/rpcsvc/autofs_prot.h
 679 file path=usr/include/rpcsvc/autofs_prot.x
 680 file path=usr/include/rpcsvc/bootparam.h
 681 file path=usr/include/rpcsvc/bootparam_prot.h
 682 file path=usr/include/rpcsvc/bootparam_prot.x
 683 file path=usr/include/rpcsvc/dbm.h
 684 file path=usr/include/rpcsvc/key_prot.x
 685 file path=usr/include/rpcsvc/mount.h
 686 file path=usr/include/rpcsvc/mount.x
 687 file path=usr/include/rpcsvc/nfs4_prot.h
 688 file path=usr/include/rpcsvc/nfs4_prot.x
 689 file path=usr/include/rpcsvc/nfs_acl.h
 690 file path=usr/include/rpcsvc/nfs_acl.x
 691 file path=usr/include/rpcsvc/nfs_prot.h
 692 file path=usr/include/rpcsvc/nfs_prot.x
 693 file path=usr/include/rpcsvc/nis.h
 694 file path=usr/include/rpcsvc/nis.x
 695 file path=usr/include/rpcsvc/nis_db.h
 696 file path=usr/include/rpcsvc/nis_object.x
 697 file path=usr/include/rpcsvc/nislib.h
 698 file path=usr/include/rpcsvc/nlm_prot.h
 699 file path=usr/include/rpcsvc/nlm_prot.x
 700 file path=usr/include/rpcsvc/nsm_addr.h
 701 file path=usr/include/rpcsvc/nsm_addr.x
 702 file path=usr/include/rpcsvc/rpc_sztypes.h
 703 file path=usr/include/rpcsvc/rpc_sztypes.x
 704 file path=usr/include/rpcsvc/rquota.h
 705 file path=usr/include/rpcsvc/rquota.x
 706 file path=usr/include/rpcsvc/rstat.h
 707 file path=usr/include/rpcsvc/rstat.x
 708 file path=usr/include/rpcsvc/rusers.h
 709 file path=usr/include/rpcsvc/rusers.x
 710 file path=usr/include/rpcsvc/rwall.h
 711 file path=usr/include/rpcsvc/rwall.x
 712 file path=usr/include/rpcsvc/sm_inter.h
 713 file path=usr/include/rpcsvc/sm_inter.x
 714 file path=usr/include/rpcsvc/spray.h
 715 file path=usr/include/rpcsvc/spray.x
 716 file path=usr/include/rpcsvc/ufs_prot.h
 717 file path=usr/include/rpcsvc/ufs_prot.x
 718 file path=usr/include/rpcsvc/yp.x
 719 file path=usr/include/rpcsvc/yp_prot.h
 720 file path=usr/include/rpcsvc/ypclnt.h

721 file path=usr/include/rpcsvc/yppasswd.h
 722 file path=usr/include/rpcsvc/ypupd.h
 723 file path=usr/include/rsmapi.h
 724 file path=usr/include/rtld_db.h
 725 file path=usr/include/sac.h
 726 file path=usr/include/sasl/prop.h
 727 file path=usr/include/sasl/sasl.h
 728 file path=usr/include/sasl/saslplug.h
 729 file path=usr/include/sasl/saslutil.h
 730 file path=usr/include/sched.h
 731 file path=usr/include/schedctl.h
 732 file path=usr/include/scsi/libscsi.h
 733 file path=usr/include/scsi/libses.h
 734 file path=usr/include/scsi/libses_plugin.h
 735 file path=usr/include/scsi/libsmpt.h
 736 file path=usr/include/scsi/libsmpt_plugin.h
 737 file path=usr/include/scsi/plugins/ses/framework/libses.h
 738 file path=usr/include/scsi/plugins/ses/framework/ses2.h
 739 file path=usr/include/scsi/plugins/ses/framework/ses2_impl.h
 740 file path=usr/include/scsi/plugins/ses/vendor/sun.h
 741 file path=usr/include/sdp.h
 742 file path=usr/include/search.h
 743 file path=usr/include/secdb.h
 744 file path=usr/include/security/auditd.h
 745 file path=usr/include/security/cryptoki.h
 746 file path=usr/include/security/pam_appl.h
 747 file path=usr/include/security/pam_modules.h
 748 file path=usr/include/security/pkcs11.h
 749 file path=usr/include/security/pkcs11f.h
 750 file path=usr/include/security/pkcs11t.h
 751 file path=usr/include/semaphore.h
 752 file path=usr/include/setjmp.h
 753 file path=usr/include/sgtty.h
 754 file path=usr/include/shal.h
 755 file path=usr/include/sha2.h
 756 file path=usr/include/shadow.h
 757 file path=usr/include/sharefs/share.h
 758 file path=usr/include/sharefs/sharefs.h
 759 file path=usr/include/sharefs/sharetab.h
 760 file path=usr/include/signinfo.h
 761 file path=usr/include/signal.h
 762 file path=usr/include/sip.h
 763 file path=usr/include/skein.h
 764 file path=usr/include/smbios.h
 765 file path=usr/include/spawn.h
 766 \$(i386_ONLY)file path=usr/include/stack_unwind.h
 767 file path=usr/include/stdalign.h
 768 file path=usr/include/stdarg.h
 769 file path=usr/include/stdbool.h
 770 file path=usr/include/stddef.h
 771 file path=usr/include/stdint.h
 772 file path=usr/include/stdio.h
 773 file path=usr/include/stdio_ext.h
 774 file path=usr/include/stdio_impl.h
 775 file path=usr/include/stdio_tag.h
 776 file path=usr/include/stdlib.h
 777 file path=usr/include/stdnoreturn.h
 778 file path=usr/include/storclass.h
 779 file path=usr/include/string.h
 780 file path=usr/include/strings.h
 781 file path=usr/include/stropts.h
 782 file path=usr/include/syms.h
 783 file path=usr/include/synch.h
 784 file path=usr/include/sys/acct.h
 785 file path=usr/include/sys/acctctl.h
 786 file path=usr/include/sys/acl.h

```

787 file path=usr/include/sys/acl_impl.h
788 file path=usr/include/sys/acpi_drv.h
789 file path=usr/include/sys/aio.h
790 file path=usr/include/sys/aio_impl.h
791 file path=usr/include/sys/aio_req.h
792 file path=usr/include/sys/aiocb.h
793 file path=usr/include/sys/archsystem.h
794 file path=usr/include/sys/ascii.h
795 file path=usr/include/sys/asm_linkage.h
796 file path=usr/include/sys/asynch.h
797 file path=usr/include/sys/atomic.h
798 file path=usr/include/sys/attr.h
799 file path=usr/include/sys/autoconf.h
800 file path=usr/include/sys/auxv.h
801 file path=usr/include/sys/auxv_386.h
802 file path=usr/include/sys/auxv_SPARC.h
803 file path=usr/include/sys/av/iec61883.h
804 file path=usr/include/sys/avintr.h
805 file path=usr/include/sys/avl.h
806 file path=usr/include/sys/avl_impl.h
807 file path=usr/include/sys/bitmap.h
808 file path=usr/include/sys/bitset.h
809 file path=usr/include/sys/bl.h
810 file path=usr/include/sys/blkdev.h
811 file path=usr/include/sys/bofi.h
812 file path=usr/include/sys/bofi_impl.h
813 file path=usr/include/sys/bootconf.h
814 $(i386_ONLY)file path=usr/include/sys/bootregs.h
815 file path=usr/include/sys/bootstat.h
816 $(i386_ONLY)file path=usr/include/sys/bootsvcs.h
817 file path=usr/include/sys/bpp_io.h
818 file path=usr/include/sys/brand.h
819 file path=usr/include/sys/buf.h
820 file path=usr/include/sys/bufmod.h
821 file path=usr/include/sys/bustypes.h
822 file path=usr/include/sys/byteorder.h
823 file path=usr/include/sys/callb.h
824 file path=usr/include/sys/callo.h
825 file path=usr/include/sys/cap_util.h
826 file path=usr/include/sys/ccompile.h
827 file path=usr/include/sys/cdio.h
828 file path=usr/include/sys/cis.h
829 file path=usr/include/sys/cis_handlers.h
830 file path=usr/include/sys/cis_protos.h
831 file path=usr/include/sys/cladm.h
832 file path=usr/include/sys/class.h
833 file path=usr/include/sys/clconf.h
834 file path=usr/include/sys/cmlb.h
835 file path=usr/include/sys/cmn_err.h
836 $(sparc_ONLY)file path=usr/include/sys/cmpregs.h
837 file path=usr/include/sys/compress.h
838 file path=usr/include/sys/condvar.h
839 file path=usr/include/sys/condvar_impl.h
840 file path=usr/include/sys/conf.h
841 file path=usr/include/sys/consdev.h
842 file path=usr/include/sys/console.h
843 file path=usr/include/sys/consplat.h
844 file path=usr/include/sys/containerof.h
845 file path=usr/include/sys/contract.h
846 file path=usr/include/sys/contract/device.h
847 file path=usr/include/sys/contract/device_impl.h
848 file path=usr/include/sys/contract/process.h
849 file path=usr/include/sys/contract/process_impl.h
850 file path=usr/include/sys/contract_impl.h
851 $(i386_ONLY)file path=usr/include/sys/controlregs.h
852 file path=usr/include/sys/copyops.h

```

```

853 file path=usr/include/sys/core.h
854 file path=usr/include/sys/corectl.h
855 file path=usr/include/sys/cpc_impl.h
856 file path=usr/include/sys/cpc_pcbe.h
857 file path=usr/include/sys/cpr.h
858 file path=usr/include/sys/cpu.h
859 file path=usr/include/sys/cpu_uarray.h
860 file path=usr/include/sys/cpucaps.h
861 file path=usr/include/sys/cpucaps_impl.h
862 file path=usr/include/sys/cpupart.h
863 file path=usr/include/sys/cpuvar.h
864 file path=usr/include/sys/crc32.h
865 file path=usr/include/sys/cred.h
866 file path=usr/include/sys/cred_impl.h
867 file path=usr/include/sys/crtctl.h
868 file path=usr/include/sys/crypto/api.h
869 file path=usr/include/sys/crypto/common.h
870 file path=usr/include/sys/crypto/ioctl.h
871 file path=usr/include/sys/crypto/ioctladmin.h
872 file path=usr/include/sys/crypto/spi.h
873 file path=usr/include/sys/cs.h
874 file path=usr/include/sys/cs_priv.h
875 file path=usr/include/sys/cs_strings.h
876 file path=usr/include/sys/cs_stubs.h
877 file path=usr/include/sys/cs_types.h
878 file path=usr/include/sys/csioctl.h
879 file path=usr/include/sys/ctf.h
880 file path=usr/include/sys/ctf_api.h
881 file path=usr/include/sys/ctfs.h
882 file path=usr/include/sys/ctfs_impl.h
883 file path=usr/include/sys/ctype.h
884 file path=usr/include/sys/cyclic.h
885 file path=usr/include/sys/cyclic_impl.h
886 file path=usr/include/sys/dacf.h
887 file path=usr/include/sys/dacf_impl.h
888 file path=usr/include/sys/damap.h
889 file path=usr/include/sys/damap_impl.h
890 file path=usr/include/sys/dc_ki.h
891 file path=usr/include/sys/ddi.h
892 file path=usr/include/sys/ddi_hp.h
893 file path=usr/include/sys/ddi_hp_impl.h
894 file path=usr/include/sys/ddi_impldefs.h
895 file path=usr/include/sys/ddi_implfuncs.h
896 file path=usr/include/sys/ddi_intr.h
897 file path=usr/include/sys/ddi_intr_impl.h
898 file path=usr/include/sys/ddi_isa.h
899 file path=usr/include/sys/ddi_obsolete.h
900 file path=usr/include/sys/ddi_periodic.h
901 file path=usr/include/sys/ddidevmap.h
902 file path=usr/include/sys/ddidmareq.h
903 file path=usr/include/sys/ddifm.h
904 file path=usr/include/sys/ddifm_impl.h
905 file path=usr/include/sys/ddimapreq.h
906 file path=usr/include/sys/ddipropdefs.h
907 file path=usr/include/sys/dditypes.h
908 file path=usr/include/sys/debug.h
909 $(i386_ONLY)file path=usr/include/sys/debugreg.h
910 file path=usr/include/sys/des.h
911 file path=usr/include/sys/devcache.h
912 file path=usr/include/sys/devcache_impl.h
913 file path=usr/include/sys/devctl.h
914 file path=usr/include/sys/devfm.h
915 file path=usr/include/sys/devid_cache.h
916 file path=usr/include/sys/devinfo_impl.h
917 file path=usr/include/sys/devops.h
918 file path=usr/include/sys/devpolicy.h

```

```

919 file path=usr/include/sys/devpoll.h
920 file path=usr/include/sys/dirent.h
921 file path=usr/include/sys/disp.h
922 file path=usr/include/sys/dkbad.h
923 file path=usr/include/sys/dkio.h
924 file path=usr/include/sys/dkioc_free_util.h
925 file path=usr/include/sys/dklabel.h
926 $(sparc_ONLY)file path=usr/include/sys/dkmpio.h
927 $(i386_ONLY)file path=usr/include/sys/dktp/altctr.h
928 $(i386_ONLY)file path=usr/include/sys/dktp/cmpkt.h
929 file path=usr/include/sys/dktp/dadkio.h
930 file path=usr/include/sys/dktp/fdisk.h
931 file path=usr/include/sys/dl.h
932 file path=usr/include/sys/dld.h
933 file path=usr/include/sys/dlpi.h
934 file path=usr/include/sys/dls_mgmt.h
935 $(i386_ONLY)file path=usr/include/sys/dma_engine.h
936 file path=usr/include/sys/dma_i8237A.h
937 file path=usr/include/sys/dnlc.h
938 file path=usr/include/sys/door.h
939 file path=usr/include/sys/door_data.h
940 file path=usr/include/sys/door_impl.h
941 file path=usr/include/sys/dumphdr.h
942 file path=usr/include/sys/ecppio.h
943 file path=usr/include/sys/ecppreg.h
944 file path=usr/include/sys/ecppsys.h
945 file path=usr/include/sys/ecppvar.h
946 file path=usr/include/sys/edonr.h
947 file path=usr/include/sys/efi_partition.h
948 file path=usr/include/sys/elf.h
949 file path=usr/include/sys/elf_386.h
950 file path=usr/include/sys/elf_SPARC.h
951 file path=usr/include/sys/elf_amd64.h
952 file path=usr/include/sys/elf_notes.h
953 file path=usr/include/sys/elftypes.h
954 file path=usr/include/sys/epm.h
955 file path=usr/include/sys/epoll.h
956 file path=usr/include/sys/errno.h
957 file path=usr/include/sys/errorq.h
958 file path=usr/include/sys/errorq_impl.h
959 file path=usr/include/sys/esunddi.h
960 file path=usr/include/sys/ethernet.h
961 file path=usr/include/sys/euc.h
962 file path=usr/include/sys/eucioctl.h
963 file path=usr/include/sys/eventfd.h
964 file path=usr/include/sys/exacct.h
965 file path=usr/include/sys/exacct_catalog.h
966 file path=usr/include/sys/exacct_impl.h
967 file path=usr/include/sys/exec.h
968 file path=usr/include/sys/exechdr.h
969 file path=usr/include/sys/fault.h
970 file path=usr/include/sys/fbio.h
971 file path=usr/include/sys/fbuf.h
972 file path=usr/include/sys/fc4/fc.h
973 file path=usr/include/sys/fc4/fc_transport.h
974 file path=usr/include/sys/fc4/fcal.h
975 file path=usr/include/sys/fc4/fcal_linkapp.h
976 file path=usr/include/sys/fc4/fcal_transport.h
977 file path=usr/include/sys/fc4/fcio.h
978 file path=usr/include/sys/fc4/fcp.h
979 file path=usr/include/sys/fc4/linkapp.h
980 file path=usr/include/sys/fcntl.h
981 file path=usr/include/sys/fdbuffer.h
982 file path=usr/include/sys/fdio.h
983 $(sparc_ONLY)file path=usr/include/sys/fdreg.h
984 $(sparc_ONLY)file path=usr/include/sys/fdvar.h

```

```

985 file path=usr/include/sys/feature_tests.h
986 file path=usr/include/sys/fem.h
987 file path=usr/include/sys/file.h
988 file path=usr/include/sys/filio.h
989 $(i386_ONLY)file path=usr/include/sys/firmload.h
990 file path=usr/include/sys/flock.h
991 file path=usr/include/sys/flock_impl.h
992 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/SPARC64-VI.h
993 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-II.h
994 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-III.h
995 $(sparc_ONLY)file path=usr/include/sys/fm/cpu/UltraSPARC-T1.h
996 file path=usr/include/sys/fm/fs/zfs.h
997 file path=usr/include/sys/fm/io/ddi.h
998 file path=usr/include/sys/fm/io/disk.h
999 file path=usr/include/sys/fm/io/opl_mc_fm.h
1000 file path=usr/include/sys/fm/io/pci.h
1001 file path=usr/include/sys/fm/io/scsi.h
1002 file path=usr/include/sys/fm/io/sun4upci.h
1003 file path=usr/include/sys/fm/protocol.h
1004 file path=usr/include/sys/fm/util.h
1005 file path=usr/include/sys/fork.h
1006 $(i386_ONLY)file path=usr/include/sys/fp.h
1007 $(sparc_ONLY)file path=usr/include/sys/fpu/fpu_simulator.h
1008 $(sparc_ONLY)file path=usr/include/sys/fpu/fpusystem.h
1009 $(sparc_ONLY)file path=usr/include/sys/fpu/globals.h
1010 $(sparc_ONLY)file path=usr/include/sys/fpu/ieee.h
1011 file path=usr/include/sys/frame.h
1012 file path=usr/include/sys/fs/autofs.h
1013 file path=usr/include/sys/fs/decomp.h
1014 file path=usr/include/sys/fs/dv_node.h
1015 file path=usr/include/sys/fs/fifonode.h
1016 file path=usr/include/sys/fs/hsfs_isospec.h
1017 file path=usr/include/sys/fs/hsfs_node.h
1018 file path=usr/include/sys/fs/hsfs_rrip.h
1019 file path=usr/include/sys/fs/hsfs_spec.h
1020 file path=usr/include/sys/fs/hsfs_susp.h
1021 file path=usr/include/sys/fs/lofs_info.h
1022 file path=usr/include/sys/fs/lofs_node.h
1023 file path=usr/include/sys/fs/mntdata.h
1024 file path=usr/include/sys/fs/namenode.h
1025 file path=usr/include/sys/fs/pc_dir.h
1026 file path=usr/include/sys/fs/pc_fs.h
1027 file path=usr/include/sys/fs/pc_label.h
1028 file path=usr/include/sys/fs/pc_node.h
1029 file path=usr/include/sys/fs/pvfs_ki.h
1030 file path=usr/include/sys/fs/sdev_impl.h
1031 file path=usr/include/sys/fs/snode.h
1032 file path=usr/include/sys/fs/swapnode.h
1033 file path=usr/include/sys/fs/tmp.h
1034 file path=usr/include/sys/fs/tmpnode.h
1035 file path=usr/include/sys/fs/udf_inode.h
1036 file path=usr/include/sys/fs/udf_volume.h
1037 file path=usr/include/sys/fs/ufs_acl.h
1038 file path=usr/include/sys/fs/ufs_bio.h
1039 file path=usr/include/sys/fs/ufs_filio.h
1040 file path=usr/include/sys/fs/ufs_fs.h
1041 file path=usr/include/sys/fs/ufs_fsdir.h
1042 file path=usr/include/sys/fs/ufs_inode.h
1043 file path=usr/include/sys/fs/ufs_lockfs.h
1044 file path=usr/include/sys/fs/ufs_log.h
1045 file path=usr/include/sys/fs/ufs_mount.h
1046 file path=usr/include/sys/fs/ufs_panic.h
1047 file path=usr/include/sys/fs/ufs_prot.h
1048 file path=usr/include/sys/fs/ufs_quota.h
1049 file path=usr/include/sys/fs/ufs_snap.h
1050 file path=usr/include/sys/fs/ufs_trans.h

```


1051 file path=usr/include/sys/fs/zfs.h
 1052 file path=usr/include/sys/fs_reparse.h
 1053 file path=usr/include/sys/fs_subr.h
 1054 file path=usr/include/sys/fsid.h
 1055 \$(sparc_ONLY)file path=usr/include/sys/fsr.h
 1056 file path=usr/include/sys/fss.h
 1057 file path=usr/include/sys/fssnap.h
 1058 file path=usr/include/sys/fssnap_if.h
 1059 file path=usr/include/sys/fsspriocntl.h
 1060 file path=usr/include/sys/fstyp.h
 1061 file path=usr/include/sys/ftrace.h
 1062 file path=usr/include/sys/fx.h
 1063 file path=usr/include/sys/fxpriocntl.h
 1064 file path=usr/include/sys/gfs.h
 1065 \$(i386_ONLY)file path=usr/include/sys/gfx_private.h
 1066 file path=usr/include/sys/gld.h
 1067 file path=usr/include/sys/gldpriv.h
 1068 file path=usr/include/sys/group.h
 1069 file path=usr/include/sys/hdio.h
 1070 file path=usr/include/sys/hook.h
 1071 file path=usr/include/sys/hook_event.h
 1072 file path=usr/include/sys/hook_impl.h
 1073 file path=usr/include/sys/hotplug/hpcsvc.h
 1074 file path=usr/include/sys/hotplug/hpctrl.h
 1075 file path=usr/include/sys/hotplug/pci/pcicfg.h
 1076 file path=usr/include/sys/hotplug/pci/pcihp.h
 1077 file path=usr/include/sys/hwconf.h
 1078 \$(i386_ONLY)file path=usr/include/sys/hypervisor.h
 1079 \$(i386_ONLY)file path=usr/include/sys/i8272A.h
 1080 file path=usr/include/sys/ia.h
 1081 file path=usr/include/sys/iapriocntl.h
 1082 file path=usr/include/sys/ib/adapters/hermon/hermon_ioctl.h
 1083 file path=usr/include/sys/ib/adapters/mlnx_umap.h
 1084 file path=usr/include/sys/ib/adapters/tavor/tavor_ioctl.h
 1085 file path=usr/include/sys/ib/clients/ibd/ibd.h
 1086 file path=usr/include/sys/ib/clients/of/ofa_solaris.h
 1087 file path=usr/include/sys/ib/clients/of/ofed_kernel.h
 1088 file path=usr/include/sys/ib/clients/of/rdma/ib_addr.h
 1089 file path=usr/include/sys/ib/clients/of/rdma/ib_user_mad.h
 1090 file path=usr/include/sys/ib/clients/of/rdma/ib_user_sa.h
 1091 file path=usr/include/sys/ib/clients/of/rdma/ib_user_verbs.h
 1092 file path=usr/include/sys/ib/clients/of/rdma/ib_verbs.h
 1093 file path=usr/include/sys/ib/clients/of/rdma/rdma_cm.h
 1094 file path=usr/include/sys/ib/clients/of/rdma/rdma_user_cm.h
 1095 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_cma.h
 1096 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ib_cma.h
 1097 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_kverb_impl.h
 1098 file path=usr/include/sys/ib/clients/of/sol_ofs/sol_ofs_common.h
 1099 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_rdma_user_cm.h
 1100 file path=usr/include/sys/ib/clients/of/sol_ucma/sol_ucma.h
 1101 file path=usr/include/sys/ib/clients/of/sol_umad/sol_umad.h
 1102 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs.h
 1103 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs2ucma.h
 1104 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_comp.h
 1105 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_event.h
 1106 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_hca.h
 1107 file path=usr/include/sys/ib/clients/of/sol_uverbs/sol_uverbs_qp.h
 1108 file path=usr/include/sys/ib/ib_pkt_hdrs.h
 1109 file path=usr/include/sys/ib/ib_types.h
 1110 file path=usr/include/sys/ib/ibnex/ibnex_devctl.h
 1111 file path=usr/include/sys/ib/ibtl/ibci.h
 1112 file path=usr/include/sys/ib/ibtl/ibti.h
 1113 file path=usr/include/sys/ib/ibtl/ibti_cm.h
 1114 file path=usr/include/sys/ib/ibtl/ibti_common.h
 1115 file path=usr/include/sys/ib/ibtl/ibtl_ci_types.h
 1116 file path=usr/include/sys/ib/ibtl/ibtl_status.h

1117 file path=usr/include/sys/ib/ibtl/ibtl_types.h
 1118 file path=usr/include/sys/ib/ibtl/ibvti.h
 1119 file path=usr/include/sys/ib/ibtl/impl/ibtl_util.h
 1120 file path=usr/include/sys/ib/mgt/ib_dm_attr.h
 1121 file path=usr/include/sys/ib/mgt/ib_mad.h
 1122 file path=usr/include/sys/ib/mgt/ibmf/ibmf.h
 1123 file path=usr/include/sys/ib/mgt/ibmf/ibmf_msg.h
 1124 file path=usr/include/sys/ib/mgt/ibmf/ibmf_saa.h
 1125 file path=usr/include/sys/ib/mgt/ibmf/ibmf_utils.h
 1126 file path=usr/include/sys/ib/mgt/sa_recs.h
 1127 file path=usr/include/sys/ib/mgt/sm_attr.h
 1128 file path=usr/include/sys/ibpart.h
 1129 file path=usr/include/sys/id32.h
 1130 file path=usr/include/sys/id_space.h
 1131 file path=usr/include/sys/idmap.h
 1132 file path=usr/include/sys/inline.h
 1133 file path=usr/include/sys/instance.h
 1134 file path=usr/include/sys/int_const.h
 1135 file path=usr/include/sys/int_fmtio.h
 1136 file path=usr/include/sys/int_limits.h
 1137 file path=usr/include/sys/int_types.h
 1138 file path=usr/include/sys/inttypes.h
 1139 file path=usr/include/sys/ioccom.h
 1140 file path=usr/include/sys/ioctl.h
 1141 \$(i386_ONLY)file path=usr/include/sys/iomulib.h
 1142 file path=usr/include/sys/ipc.h
 1143 file path=usr/include/sys/ipc_impl.h
 1144 file path=usr/include/sys/ipc_rctl.h
 1145 file path=usr/include/sys/isa_defs.h
 1146 file path=usr/include/sys/iso/signal_iso.h
 1147 file path=usr/include/sys/jioctl.h
 1148 file path=usr/include/sys/kbd.h
 1149 file path=usr/include/sys/kbdreg.h
 1150 file path=usr/include/sys/kbio.h
 1151 file path=usr/include/sys/kcpc.h
 1152 file path=usr/include/sys/kd.h
 1153 file path=usr/include/sys/kdi.h
 1154 file path=usr/include/sys/kdi_impl.h
 1155 file path=usr/include/sys/kdi_machimpl.h
 1156 \$(i386_ONLY)file path=usr/include/sys/kdi_regs.h
 1157 file path=usr/include/sys/kiconv.h
 1158 file path=usr/include/sys/kidmap.h
 1159 file path=usr/include/sys/klpd.h
 1160 file path=usr/include/sys/klwp.h
 1161 file path=usr/include/sys/kmem.h
 1162 file path=usr/include/sys/kmem_impl.h
 1163 file path=usr/include/sys/kobj.h
 1164 file path=usr/include/sys/kobj_impl.h
 1165 file path=usr/include/sys/ksocket.h
 1166 file path=usr/include/sys/kstat.h
 1167 file path=usr/include/sys/kstr.h
 1168 file path=usr/include/sys/ksyms.h
 1169 file path=usr/include/sys/ksynch.h
 1170 file path=usr/include/sys/lc_core.h
 1171 file path=usr/include/sys/ldterm.h
 1172 file path=usr/include/sys/lgrp.h
 1173 file path=usr/include/sys/lgrp_user.h
 1174 file path=usr/include/sys/link.h
 1175 file path=usr/include/sys/list.h
 1176 file path=usr/include/sys/list_impl.h
 1177 file path=usr/include/sys/llcl.h
 1178 file path=usr/include/sys/loadavg.h
 1179 file path=usr/include/sys/localedef.h
 1180 file path=usr/include/sys/lock.h
 1181 file path=usr/include/sys/lockfs.h
 1182 file path=usr/include/sys/lofi.h

1183 file path=usr/include/sys/log.h
 1184 file path=usr/include/sys/logindmux.h
 1185 file path=usr/include/sys/lwp.h
 1186 file path=usr/include/sys/lwp_timer_impl.h
 1187 file path=usr/include/sys/lwp_upimutex_impl.h
 1188 file path=usr/include/sys/mac.h
 1189 file path=usr/include/sys/mac_ether.h
 1190 file path=usr/include/sys/mac_flow.h
 1191 file path=usr/include/sys/mac_provider.h
 1192 file path=usr/include/sys/machelf.h
 1193 file path=usr/include/sys/machlock.h
 1194 file path=usr/include/sys/machsig.h
 1195 file path=usr/include/sys/machtypes.h
 1196 file path=usr/include/sys/map.h
 1197 \$(i386_ONLY)file path=usr/include/sys/mc.h
 1198 \$(i386_ONLY)file path=usr/include/sys/mc_amd.h
 1199 \$(i386_ONLY)file path=usr/include/sys/mc_intel.h
 1200 \$(i386_ONLY)file path=usr/include/sys/mca_amd.h
 1201 \$(i386_ONLY)file path=usr/include/sys/mca_x86.h
 1202 file path=usr/include/sys/mcontext.h
 1203 file path=usr/include/sys/md4.h
 1204 file path=usr/include/sys/md5.h
 1205 file path=usr/include/sys/md5_consts.h
 1206 file path=usr/include/sys/mdi_impldefs.h
 1207 file path=usr/include/sys/mem.h
 1208 file path=usr/include/sys/mem_config.h
 1209 file path=usr/include/sys/memlist.h
 1210 file path=usr/include/sys/mhd.h
 1211 file path=usr/include/sys/mii.h
 1212 file path=usr/include/sys/miiregs.h
 1213 file path=usr/include/sys/mkdev.h
 1214 file path=usr/include/sys/mman.h
 1215 file path=usr/include/sys/mmapobj.h
 1216 file path=usr/include/sys/mntent.h
 1217 file path=usr/include/sys/mntio.h
 1218 file path=usr/include/sys/mnttab.h
 1219 file path=usr/include/sys/modctl.h
 1220 file path=usr/include/sys/mode.h
 1221 file path=usr/include/sys/model.h
 1222 file path=usr/include/sys/modhash.h
 1223 file path=usr/include/sys/modhash_impl.h
 1224 file path=usr/include/sys/mount.h
 1225 file path=usr/include/sys/mouse.h
 1226 file path=usr/include/sys/msacct.h
 1227 file path=usr/include/sys/msg.h
 1228 file path=usr/include/sys/msg_impl.h
 1229 file path=usr/include/sys/msio.h
 1230 file path=usr/include/sys/msreg.h
 1231 file path=usr/include/sys/mtio.h
 1232 file path=usr/include/sys/multidata.h
 1233 file path=usr/include/sys/mutex.h
 1234 \$(i386_ONLY)file path=usr/include/sys/mutex_impl.h
 1235 file path=usr/include/sys/nbmlock.h
 1236 file path=usr/include/sys/ndi_impldefs.h
 1237 file path=usr/include/sys/ndifm.h
 1238 file path=usr/include/sys/netconfig.h
 1239 file path=usr/include/sys/neti.h
 1240 file path=usr/include/sys/netstack.h
 1241 file path=usr/include/sys/nexusdefs.h
 1242 file path=usr/include/sys/note.h
 1243 file path=usr/include/sys/null.h
 1244 file path=usr/include/sys/nvpair.h
 1245 file path=usr/include/sys/nvpair_impl.h
 1246 file path=usr/include/sys/objfs.h
 1247 file path=usr/include/sys/objfs_impl.h
 1248 file path=usr/include/sys/obpdefs.h

1249 file path=usr/include/sys/old_procfs.h
 1250 file path=usr/include/sys/open.h
 1251 file path=usr/include/sys/openpromio.h
 1252 file path=usr/include/sys/panic.h
 1253 file path=usr/include/sys/param.h
 1254 file path=usr/include/sys/pathconf.h
 1255 file path=usr/include/sys/pathname.h
 1256 file path=usr/include/sys/pattr.h
 1257 file path=usr/include/sys/pbio.h
 1258 file path=usr/include/sys/pcb.h
 1259 file path=usr/include/sys/pccard.h
 1260 file path=usr/include/sys/pci.h
 1261 \$(i386_ONLY)file path=usr/include/sys/pcic_reg.h
 1262 \$(i386_ONLY)file path=usr/include/sys/pcic_var.h
 1263 file path=usr/include/sys/pcie.h
 1264 file path=usr/include/sys/pcmcia.h
 1265 file path=usr/include/sys/pctypes.h
 1266 file path=usr/include/sys/pfmod.h
 1267 file path=usr/include/sys/pg.h
 1268 file path=usr/include/sys/pghw.h
 1269 file path=usr/include/sys/phymem.h
 1270 \$(i386_ONLY)file path=usr/include/sys/pic.h
 1271 \$(i386_ONLY)file path=usr/include/sys/pit.h
 1272 file path=usr/include/sys/pkp_hash.h
 1273 file path=usr/include/sys/pm.h
 1274 \$(i386_ONLY)file path=usr/include/sys/pmem.h
 1275 file path=usr/include/sys/policy.h
 1276 file path=usr/include/sys/poll.h
 1277 file path=usr/include/sys/poll_impl.h
 1278 file path=usr/include/sys/pool.h
 1279 file path=usr/include/sys/pool_impl.h
 1280 file path=usr/include/sys/pool_pset.h
 1281 file path=usr/include/sys/port.h
 1282 file path=usr/include/sys/port_impl.h
 1283 file path=usr/include/sys/port_kernel.h
 1284 file path=usr/include/sys/ppmio.h
 1285 file path=usr/include/sys/priocntl.h
 1286 file path=usr/include/sys/priv.h
 1287 file path=usr/include/sys/priv_const.h
 1288 file path=usr/include/sys/priv_impl.h
 1289 file path=usr/include/sys/priv_names.h
 1290 \$(i386_ONLY)file path=usr/include/sys/privregs.h
 1291 \$(i386_ONLY)file path=usr/include/sys/privregs.h
 1292 file path=usr/include/sys/prnio.h
 1293 file path=usr/include/sys/proc.h
 1294 file path=usr/include/sys/proc/prdata.h
 1295 file path=usr/include/sys/processor.h
 1296 file path=usr/include/sys/procfs.h
 1297 file path=usr/include/sys/procfs_isa.h
 1298 file path=usr/include/sys/procset.h
 1299 file path=usr/include/sys/project.h
 1300 \$(i386_ONLY)file path=usr/include/sys/prom_emul.h
 1301 \$(i386_ONLY)file path=usr/include/sys/prom_isa.h
 1302 \$(i386_ONLY)file path=usr/include/sys/prom_plat.h
 1303 file path=usr/include/sys/promif.h
 1304 file path=usr/include/sys/promimpl.h
 1305 file path=usr/include/sys/protosw.h
 1306 file path=usr/include/sys/prsystem.h
 1307 file path=usr/include/sys/pset.h
 1308 file path=usr/include/sys/psw.h
 1309 \$(i386_ONLY)file path=usr/include/sys/pte.h
 1310 file path=usr/include/sys/ptem.h
 1311 file path=usr/include/sys/ptms.h
 1312 file path=usr/include/sys/ptyvar.h
 1313 file path=usr/include/sys/queue.h
 1314 file path=usr/include/sys/raidioctl.h

```

1315 file path=usr/include/sys/ramdisk.h
1316 file path=usr/include/sys/random.h
1317 file path=usr/include/sys/rctl.h
1318 file path=usr/include/sys/rctl_impl.h
1319 file path=usr/include/sys/rds.h
1320 file path=usr/include/sys/reboot.h
1321 file path=usr/include/sys/refstr.h
1322 file path=usr/include/sys/refstr_impl.h
1323 file path=usr/include/sys/reg.h
1324 file path=usr/include/sys/regset.h
1325 file path=usr/include/sys/resource.h
1326 file path=usr/include/sys/rlioctl.h
1327 file path=usr/include/sys/rsm/rsm.h
1328 file path=usr/include/sys/rsm/rsm_common.h
1329 file path=usr/include/sys/rsm/rsmapi_common.h
1330 file path=usr/include/sys/rsm/rsmka_path_int.h
1331 file path=usr/include/sys/rsm/rsmndi.h
1332 file path=usr/include/sys/rsm/rsmmpi.h
1333 file path=usr/include/sys/rsm/rsmmpi_driver.h
1334 file path=usr/include/sys/rt.h
1335 $(i386_ONLY)file path=usr/include/sys/rtc.h
1336 file path=usr/include/sys/rtpriocntl.h
1337 file path=usr/include/sys/rwlock.h
1338 file path=usr/include/sys/rwlock_impl.h
1339 file path=usr/include/sys/rwstlock.h
1340 file path=usr/include/sys/sad.h
1341 $(i386_ONLY)file path=usr/include/sys/sata/sata_defs.h
1342 $(i386_ONLY)file path=usr/include/sys/sata/sata_hba.h
1343 file path=usr/include/sys/schedctl.h
1344 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/ifpio.h
1345 file path=usr/include/sys/scsi/adapters/scsi_vhci.h
1346 $(sparc_ONLY)file path=usr/include/sys/scsi/adapters/sfvar.h
1347 file path=usr/include/sys/scsi/conf/autoconf.h
1348 file path=usr/include/sys/scsi/conf/device.h
1349 file path=usr/include/sys/scsi/generic/commands.h
1350 file path=usr/include/sys/scsi/generic/dad_mode.h
1351 file path=usr/include/sys/scsi/generic/inquiry.h
1352 file path=usr/include/sys/scsi/generic/message.h
1353 file path=usr/include/sys/scsi/generic/mode.h
1354 file path=usr/include/sys/scsi/generic/persist.h
1355 file path=usr/include/sys/scsi/generic/sense.h
1356 file path=usr/include/sys/scsi/generic/sff_frames.h
1357 file path=usr/include/sys/scsi/generic/smp_frames.h
1358 file path=usr/include/sys/scsi/generic/status.h
1359 file path=usr/include/sys/scsi/impl/commands.h
1360 file path=usr/include/sys/scsi/impl/inquiry.h
1361 file path=usr/include/sys/scsi/impl/mode.h
1362 file path=usr/include/sys/scsi/impl/scsi_reset_notify.h
1363 file path=usr/include/sys/scsi/impl/scsi_sas.h
1364 file path=usr/include/sys/scsi/impl/sense.h
1365 file path=usr/include/sys/scsi/impl/services.h
1366 file path=usr/include/sys/scsi/impl/smp_transport.h
1367 file path=usr/include/sys/scsi/impl/spc3_types.h
1368 file path=usr/include/sys/scsi/impl/status.h
1369 file path=usr/include/sys/scsi/impl/transport.h
1370 file path=usr/include/sys/scsi/impl/types.h
1371 file path=usr/include/sys/scsi/impl/uscsi.h
1372 file path=usr/include/sys/scsi/impl/usmp.h
1373 file path=usr/include/sys/scsi/scsi.h
1374 file path=usr/include/sys/scsi/scsi_address.h
1375 file path=usr/include/sys/scsi/scsi_ctl.h
1376 file path=usr/include/sys/scsi/scsi_fm.h
1377 file path=usr/include/sys/scsi/scsi_names.h
1378 file path=usr/include/sys/scsi/scsi_params.h
1379 file path=usr/include/sys/scsi/scsi_pkt.h
1380 file path=usr/include/sys/scsi/scsi_resource.h

```

```

1381 file path=usr/include/sys/scsi/scsi_types.h
1382 file path=usr/include/sys/scsi/scsi_watch.h
1383 file path=usr/include/sys/scsi/targets/sddef.h
1384 file path=usr/include/sys/scsi/targets/ses.h
1385 file path=usr/include/sys/scsi/targets/sesio.h
1386 file path=usr/include/sys/scsi/targets/sgendef.h
1387 file path=usr/include/sys/scsi/targets/smp.h
1388 $(sparc_ONLY)file path=usr/include/sys/scsi/targets/ssddef.h
1389 file path=usr/include/sys/scsi/targets/stdef.h
1390 file path=usr/include/sys/secflags.h
1391 $(i386_ONLY)file path=usr/include/sys/segment.h
1392 $(i386_ONLY)file path=usr/include/sys/segments.h
1393 file path=usr/include/sys/select.h
1394 file path=usr/include/sys/sem.h
1395 file path=usr/include/sys/sem_impl.h
1396 file path=usr/include/sys/sem_impl.h
1397 file path=usr/include/sys/semaphore.h
1398 file path=usr/include/sys/sendfile.h
1399 $(sparc_ONLY)file path=usr/include/sys/ser_async.h
1400 file path=usr/include/sys/ser_sync.h
1401 $(sparc_ONLY)file path=usr/include/sys/ser_zscc.h
1402 file path=usr/include/sys/serializer.h
1403 file path=usr/include/sys/session.h
1404 file path=usr/include/sys/sha1.h
1405 file path=usr/include/sys/sha2.h
1406 file path=usr/include/sys/share.h
1407 file path=usr/include/sys/shm.h
1408 file path=usr/include/sys/shm_impl.h
1409 file path=usr/include/sys/sid.h
1410 file path=usr/include/sys/siginfo.h
1411 file path=usr/include/sys/signal.h
1412 file path=usr/include/sys/signalfd.h
1413 file path=usr/include/sys/skein.h
1414 file path=usr/include/sys/sleepq.h
1415 file path=usr/include/sys/smbios.h
1416 file path=usr/include/sys/smbios_impl.h
1417 file path=usr/include/sys/smedia.h
1418 file path=usr/include/sys/subject.h
1419 $(sparc_ONLY)file path=usr/include/sys/social_cg_defs.h
1420 $(sparc_ONLY)file path=usr/include/sys/socialio.h
1421 $(sparc_ONLY)file path=usr/include/sys/socialmap.h
1422 $(sparc_ONLY)file path=usr/include/sys/socialreg.h
1423 $(sparc_ONLY)file path=usr/include/sys/socialvar.h
1424 file path=usr/include/sys/socket.h
1425 file path=usr/include/sys/socket_impl.h
1426 file path=usr/include/sys/socket_proto.h
1427 file path=usr/include/sys/socketvar.h
1428 file path=usr/include/sys/sockio.h
1429 file path=usr/include/sys/spl.h
1430 file path=usr/include/sys/squeue.h
1431 file path=usr/include/sys/squeue_impl.h
1432 file path=usr/include/sys/sservice.h
1433 file path=usr/include/sys/stack.h
1434 file path=usr/include/sys/stat.h
1435 file path=usr/include/sys/stat_impl.h
1436 file path=usr/include/sys/statfs.h
1437 file path=usr/include/sys/statvfs.h
1438 file path=usr/include/sys/stdbool.h
1439 file path=usr/include/sys/stddef.h
1440 file path=usr/include/sys/stdint.h
1441 file path=usr/include/sys/stermio.h
1442 file path=usr/include/sys/stream.h
1443 file path=usr/include/sys/strft.h
1444 file path=usr/include/sys/strlog.h
1445 file path=usr/include/sys/strmdep.h
1446 file path=usr/include/sys/stropts.h

```

1447 file path=usr/include/sys/strredirect.h
1448 file path=usr/include/sys/strstat.h
1449 file path=usr/include/sys/strsubr.h
1450 file path=usr/include/sys/strsun.h
1451 file path=usr/include/sys/strtty.h
1452 file path=usr/include/sys/sunddi.h
1453 file path=usr/include/sys/sunldi.h
1454 file path=usr/include/sys/sunldi_impl.h
1455 file path=usr/include/sys/sunmdi.h
1456 file path=usr/include/sys/sunndi.h
1457 file path=usr/include/sys/sunpm.h
1458 file path=usr/include/sys/suntpi.h
1459 file path=usr/include/sys/suntty.h
1460 file path=usr/include/sys/swap.h
1461 file path=usr/include/sys/synch.h
1462 file path=usr/include/sys/syscall.h
1463 file path=usr/include/sys/sysconf.h
1464 file path=usr/include/sys/sysconfig.h
1465 file path=usr/include/sys/sysconfig_impl.h
1466 file path=usr/include/sys/sysdc.h
1467 file path=usr/include/sys/sysdc_impl.h
1468 file path=usr/include/sys/sysevent.h
1469 file path=usr/include/sys/sysevent/ap_driver.h
1470 file path=usr/include/sys/sysevent/dev.h
1471 file path=usr/include/sys/sysevent/domain.h
1472 file path=usr/include/sys/sysevent/dr.h
1473 file path=usr/include/sys/sysevent/env.h
1474 file path=usr/include/sys/sysevent/eventdefs.h
1475 file path=usr/include/sys/sysevent/ipmp.h
1476 file path=usr/include/sys/sysevent/pwrctl.h
1477 file path=usr/include/sys/sysevent/vrrp.h
1478 file path=usr/include/sys/sysevent_impl.h
1479 \$(i386_ONLY)file path=usr/include/sys/sysi86.h
1480 file path=usr/include/sys/sysinfo.h
1481 file path=usr/include/sys/syslog.h
1482 file path=usr/include/sys/sysmacros.h
1483 file path=usr/include/sys/systeminfo.h
1484 file path=usr/include/sys/system.h
1485 file path=usr/include/sys/t_kuser.h
1486 file path=usr/include/sys/t_lock.h
1487 file path=usr/include/sys/task.h
1488 file path=usr/include/sys/taskq.h
1489 file path=usr/include/sys/taskq_impl.h
1490 file path=usr/include/sys/teliocctl.h
1491 file path=usr/include/sys/termio.h
1492 file path=usr/include/sys/termios.h
1493 file path=usr/include/sys/termiox.h
1494 file path=usr/include/sys/thread.h
1495 file path=usr/include/sys/ticlts.h
1496 file path=usr/include/sys/ticots.h
1497 file path=usr/include/sys/ticotsord.h
1498 file path=usr/include/sys/tihdr.h
1499 file path=usr/include/sys/time.h
1500 file path=usr/include/sys/time_impl.h
1501 file path=usr/include/sys/time_std_impl.h
1502 file path=usr/include/sys/timeb.h
1503 file path=usr/include/sys/timer.h
1504 file path=usr/include/sys/timerfd.h
1505 file path=usr/include/sys/times.h
1506 file path=usr/include/sys/timex.h
1507 file path=usr/include/sys/timod.h
1508 file path=usr/include/sys/tirdwr.h
1509 file path=usr/include/sys/tiuser.h
1510 file path=usr/include/sys/tl.h
1511 file path=usr/include/sys/tnf.h
1512 file path=usr/include/sys/tnf_com.h

1513 file path=usr/include/sys/tnf_probe.h
1514 file path=usr/include/sys/tnf_writer.h
1515 file path=usr/include/sys/todio.h
1516 file path=usr/include/sys/tpicommon.h
1517 file path=usr/include/sys/trap.h
1518 \$(i386_ONLY)file path=usr/include/sys/traptrace.h
1519 file path=usr/include/sys/ts.h
1520 file path=usr/include/sys/tsol/label.h
1521 file path=usr/include/sys/tsol/label_macro.h
1522 file path=usr/include/sys/tsol/priv.h
1523 file path=usr/include/sys/tsol/tndb.h
1524 file path=usr/include/sys/tsol/tsyscall.h
1525 file path=usr/include/sys/tpsriocntl.h
1526 \$(i386_ONLY)file path=usr/include/sys/tss.h
1527 file path=usr/include/sys/ttcompat.h
1528 file path=usr/include/sys/ttold.h
1529 file path=usr/include/sys/tty.h
1530 file path=usr/include/sys/ttychars.h
1531 file path=usr/include/sys/ttydev.h
1532 \$(sparc_ONLY)file path=usr/include/sys/ttymux.h
1533 \$(sparc_ONLY)file path=usr/include/sys/ttymuxuser.h
1534 file path=usr/include/sys/tuneable.h
1535 file path=usr/include/sys/turnstile.h
1536 file path=usr/include/sys/types.h
1537 file path=usr/include/sys/types32.h
1538 file path=usr/include/sys/tzfile.h
1539 file path=usr/include/sys/u8_textprep.h
1540 file path=usr/include/sys/uadmin.h
1541 \$(i386_ONLY)file path=usr/include/sys/ucode.h
1542 file path=usr/include/sys/ucontext.h
1543 file path=usr/include/sys/uiio.h
1544 file path=usr/include/sys/ulimit.h
1545 file path=usr/include/sys/un.h
1546 file path=usr/include/sys/unistd.h
1547 file path=usr/include/sys/user.h
1548 file path=usr/include/sys/ustat.h
1549 file path=usr/include/sys/utime.h
1550 file path=usr/include/sys/utrap.h
1551 file path=usr/include/sys/utsname.h
1552 file path=usr/include/sys/utssys.h
1553 file path=usr/include/sys/uuid.h
1554 file path=usr/include/sys/va_impl.h
1555 file path=usr/include/sys/va_list.h
1556 file path=usr/include/sys/var.h
1557 file path=usr/include/sys/varargs.h
1558 file path=usr/include/sys/vfs.h
1559 file path=usr/include/sys/vfs_opreg.h
1560 file path=usr/include/sys/vfstab.h
1561 file path=usr/include/sys/videodev2.h
1562 file path=usr/include/sys/visual_io.h
1563 file path=usr/include/sys/vm.h
1564 file path=usr/include/sys/vm_usage.h
1565 file path=usr/include/sys/vmem.h
1566 file path=usr/include/sys/vmem_impl.h
1567 file path=usr/include/sys/vmem_impl_user.h
1568 file path=usr/include/sys/vmparam.h
1569 file path=usr/include/sys/vmsystem.h
1570 file path=usr/include/sys/vnode.h
1571 file path=usr/include/sys/vt.h
1572 file path=usr/include/sys/vtdaemon.h
1573 file path=usr/include/sys/vtoc.h
1574 file path=usr/include/sys/vtrace.h
1575 file path=usr/include/sys/vuid_event.h
1576 file path=usr/include/sys/vuid_queue.h
1577 file path=usr/include/sys/vuid_state.h
1578 file path=usr/include/sys/vuid_store.h

```

1579 file path=usr/include/sys/vuid_wheel.h
1580 file path=usr/include/sys/wait.h
1581 file path=usr/include/sys/waitq.h
1582 file path=usr/include/sys/watchpoint.h
1583 $(i386_ONLY)file path=usr/include/sys/x86_archext.h
1584 $(i386_ONLY)file path=usr/include/sys/xen_errno.h
1585 file path=usr/include/sys/xti_inet.h
1586 file path=usr/include/sys/xti_osi.h
1587 file path=usr/include/sys/xti_xtiopt.h
1588 file path=usr/include/sys/zcons.h
1589 file path=usr/include/sys/zmod.h
1590 file path=usr/include/sys/zone.h
1591 $(sparc_ONLY)file path=usr/include/sys/zsdev.h
1592 file path=usr/include/sysexts.h
1593 file path=usr/include/syslog.h
1594 file path=usr/include/tar.h
1595 file path=usr/include/tcpd.h
1596 file path=usr/include/term.h
1597 file path=usr/include/termcap.h
1598 file path=usr/include/termio.h
1599 file path=usr/include/termios.h
1600 file path=usr/include/thread.h
1601 file path=usr/include/thread_db.h
1602 file path=usr/include/threads.h
1603 file path=usr/include/time.h
1604 file path=usr/include/tiuser.h
1605 file path=usr/include/tsol/label.h
1606 file path=usr/include/tzfile.h
1607 file path=usr/include/ucontext.h
1608 file path=usr/include/ucrd.h
1609 file path=usr/include/uid_stp.h
1610 file path=usr/include/ulimit.h
1611 file path=usr/include/umem.h
1612 file path=usr/include/umem_impl.h
1613 file path=usr/include/unctrl.h
1614 file path=usr/include/unistd.h
1615 file path=usr/include/user_attr.h
1616 file path=usr/include/userdefs.h
1617 file path=usr/include/ustat.h
1618 file path=usr/include/utility.h
1619 file path=usr/include/utime.h
1620 file path=usr/include/utmp.h
1621 file path=usr/include/utmpx.h
1622 file path=usr/include/uuid/uuid.h
1623 $(sparc_ONLY)file path=usr/include/v7/sys/machpcb.h
1624 $(sparc_ONLY)file path=usr/include/v7/sys/machtrap.h
1625 $(sparc_ONLY)file path=usr/include/v7/sys/mutex_impl.h
1626 $(sparc_ONLY)file path=usr/include/v7/sys/privregs.h
1627 $(sparc_ONLY)file path=usr/include/v7/sys/prom_isa.h
1628 $(sparc_ONLY)file path=usr/include/v7/sys/psr.h
1629 $(sparc_ONLY)file path=usr/include/v7/sys/traptrace.h
1630 $(sparc_ONLY)file path=usr/include/v9/sys/asi.h
1631 $(sparc_ONLY)file path=usr/include/v9/sys/machpcb.h
1632 $(sparc_ONLY)file path=usr/include/v9/sys/machtrap.h
1633 $(sparc_ONLY)file path=usr/include/v9/sys/membar.h
1634 $(sparc_ONLY)file path=usr/include/v9/sys/mutex_impl.h
1635 $(sparc_ONLY)file path=usr/include/v9/sys/privregs.h
1636 $(sparc_ONLY)file path=usr/include/v9/sys/prom_isa.h
1637 $(sparc_ONLY)file path=usr/include/v9/sys/psr_compat.h
1638 $(sparc_ONLY)file path=usr/include/v9/sys/vis_simulator.h
1639 file path=usr/include/valtools.h
1640 file path=usr/include/values.h
1641 file path=usr/include/varargs.h
1642 file path=usr/include/vm/anon.h
1643 file path=usr/include/vm/as.h
1644 file path=usr/include/vm/faultcode.h

```

```

1645 file path=usr/include/vm/hat.h
1646 file path=usr/include/vm/kpm.h
1647 file path=usr/include/vm/page.h
1648 file path=usr/include/vm/pvn.h
1649 file path=usr/include/vm/rm.h
1650 file path=usr/include/vm/seg.h
1651 file path=usr/include/vm/seg_dev.h
1652 file path=usr/include/vm/seg_enum.h
1653 file path=usr/include/vm/seg_kmem.h
1654 file path=usr/include/vm/seg_kp.h
1655 file path=usr/include/vm/seg_kpm.h
1656 file path=usr/include/vm/seg_map.h
1657 file path=usr/include/vm/seg_spt.h
1658 file path=usr/include/vm/seg_vn.h
1659 file path=usr/include/vm/vpage.h
1660 file path=usr/include/vm/vpm.h
1661 file path=usr/include/volmgt.h
1662 file path=usr/include/wait.h
1663 file path=usr/include/wchar.h
1664 file path=usr/include/wchar_impl.h
1665 file path=usr/include/wctype.h
1666 file path=usr/include/widec.h
1667 file path=usr/include/wordexp.h
1668 file path=usr/include/xlocale.h
1669 file path=usr/include/xti.h
1670 file path=usr/include/xti_inet.h
1671 file path=usr/include/zone.h
1672 file path=usr/include/zonestat.h
1673 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/acpidev.h
1674 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/amd_iommu.h
1675 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/asm_misc.h
1676 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/clock.h
1677 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/cram.h
1678 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/ddi_subdefs.h
1679 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/debug_info.h
1680 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/fastboot.h
1681 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/mach_mmu.h
1682 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machclock.h
1683 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machcpuvar.h
1684 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machparam.h
1685 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machprivregs.h
1686 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machsystem.h
1687 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/machthread.h
1688 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/memmod.h
1689 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/pmmu.h
1690 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm.h
1691 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_defs.h
1692 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_modctl.h
1693 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/psm_types.h
1694 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/rm_platter.h
1695 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/sbd_ioctl.h
1696 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/smp_impldefs.h
1697 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/vm_machparam.h
1698 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/x_call.h
1699 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xc_levels.h
1700 $(i386_ONLY)file path=usr/platform/i86pc/include/sys/xsvc.h
1701 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_i86.h
1702 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hat_pte.h
1703 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/hment.h
1704 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/htable.h
1705 $(i386_ONLY)file path=usr/platform/i86pc/include/vm/kboot_mmu.h
1706 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/balloon.h
1707 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/machprivregs.h
1708 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xen_mmu.h
1709 $(i386_ONLY)file path=usr/platform/i86xpv/include/sys/xpv_impl.h
1710 $(i386_ONLY)file path=usr/platform/i86xpv/include/vm/seg_mf.h

```

```

1711 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ac.h
1712 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/async.h
1713 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cheetahregs.h
1714 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cherryastone.h
1715 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/clock.h
1716 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cmp.h
1717 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpc_ultra.h
1718 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpr_impl.h
1719 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_impl.h
1720 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cpu_sgnblk_defs.h
1721 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/cvc.h
1722 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/daktari.h
1723 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ddi_subrdefs.h
1724 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/dvma.h
1725 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/ecc_kstat.h
1726 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/eprom.h
1727 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl.h
1728 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_gen.h
1729 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue250.h
1730 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/envctrl_ue450.h
1731 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/environ.h
1732 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/errclassify.h
1733 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/fhc.h
1734 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/gpio_87317.h
1735 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/hpc3130_events.h
1736 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/hpc3130.h
1737 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/i2c_client.h
1738 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/lm75.h
1739 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/max1617.h
1740 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/pcf8591.h
1741 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c_clients/ssc050.h
1742 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/i2c/misc/i2c_svc.h
1743 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/idprom.h
1744 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intr.h
1745 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/intreg.h
1746 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iocache.h
1747 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iommu.h
1748 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/iovintr.h
1749 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/lom_io.h
1750 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machasi.h
1751 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machclock.h
1752 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machcpuvar.h
1753 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machparam.h
1754 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machsystem.h
1755 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/machthread.h
1756 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mem_cache.h
1757 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memlist_plat.h
1758 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/memnode.h
1759 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/mmu.h
1760 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/nexusdebug.h
1761 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_hwdesc.h
1762 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/opl_module.h
1763 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_debug.h
1764 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/prom_plat.h
1765 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/pte.h
1766 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sbd_ioctl.h
1767 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scb.h
1768 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/scsb_led.h
1769 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/simpstat.h
1770 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/spitregs.h
1771 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sram.h
1772 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/starfire.h
1773 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sun4asi.h
1774 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysctrl.h
1775 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysioerr.h
1776 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/sysiosbus.h

```

```

1777 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/tod.h
1778 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/todmostek.h
1779 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/trapstat.h
1780 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/traptrace.h
1781 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vis.h
1782 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/vm_machparam.h
1783 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/x_call.h
1784 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/xc_impl.h
1785 $(sparc_ONLY)file path=usr/platform/sun4u/include/sys/zsmach.h
1786 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/hat_sfmmu.h
1787 $(sparc_ONLY)file path=usr/platform/sun4u/include/vm/mach_sfmmu.h
1788 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/clock.h
1789 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cmp.h
1790 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpc_ultra.h
1791 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/cpu_sgnblk_defs.h
1792 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ddi_subrdefs.h
1793 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_pri.h
1794 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ds_smp.h
1795 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/dvma.h
1796 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/eprom.h
1797 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/fcode.h
1798 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hsvc.h
1799 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/hypervisor_api.h
1800 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/idprom.h
1801 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intr.h
1802 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/intreg.h
1803 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ivintr.h
1804 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machasi.h
1805 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machclock.h
1806 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machcpuvar.h
1807 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machintreg.h
1808 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machparam.h
1809 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machsystem.h
1810 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/machthread.h
1811 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memlist_plat.h
1812 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/memnode.h
1813 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/mmu.h
1814 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/nexusdebug.h
1815 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagaraasi.h
1816 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/niagararegs.h
1817 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/ntwdt.h
1818 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pri.h
1819 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_debug.h
1820 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/prom_plat.h
1821 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/pte.h
1822 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/qcn.h
1823 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/scb.h
1824 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/soft_state.h
1825 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/sun4asi.h
1826 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/tod.h
1827 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/trapstat.h
1828 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/traptrace.h
1829 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vis.h
1830 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/vm_machparam.h
1831 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/x_call.h
1832 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/xc_impl.h
1833 $(sparc_ONLY)file path=usr/platform/sun4v/include/sys/zsmach.h
1834 $(sparc_ONLY)file path=usr/platform/sun4v/include/vm/hat_sfmmu.h
1835 $(sparc_ONLY)file path=usr/platform/sun4v/include/vm/mach_sfmmu.h
1836 file path=usr/share/man/man3head/acct.h.3head
1837 file path=usr/share/man/man3head/aio.h.3head
1838 file path=usr/share/man/man3head/ar.h.3head
1839 file path=usr/share/man/man3head/archives.h.3head
1840 file path=usr/share/man/man3head/assert.h.3head
1841 file path=usr/share/man/man3head/complex.h.3head
1842 file path=usr/share/man/man3head/cpio.h.3head

```

```

1843 file path=usr/share/man/man3head/dirent.h.3head
1844 file path=usr/share/man/man3head/endian.h.3head
1845 file path=usr/share/man/man3head/errno.h.3head
1846 file path=usr/share/man/man3head/fcntl.h.3head
1847 file path=usr/share/man/man3head/fenv.h.3head
1848 file path=usr/share/man/man3head/float.h.3head
1849 file path=usr/share/man/man3head/floatingpoint.h.3head
1850 file path=usr/share/man/man3head/fmtmsg.h.3head
1851 file path=usr/share/man/man3head/fnmatch.h.3head
1852 file path=usr/share/man/man3head/ftw.h.3head
1853 file path=usr/share/man/man3head/glob.h.3head
1854 file path=usr/share/man/man3head/grp.h.3head
1855 file path=usr/share/man/man3head/iconv.h.3head
1856 file path=usr/share/man/man3head/if.h.3head
1857 file path=usr/share/man/man3head/in.h.3head
1858 file path=usr/share/man/man3head/inet.h.3head
1859 file path=usr/share/man/man3head/inttypes.h.3head
1860 file path=usr/share/man/man3head/ipc.h.3head
1861 file path=usr/share/man/man3head/iso646.h.3head
1862 file path=usr/share/man/man3head/langinfo.h.3head
1863 file path=usr/share/man/man3head/libgen.h.3head
1864 file path=usr/share/man/man3head/libintl.h.3head
1865 file path=usr/share/man/man3head/limits.h.3head
1866 file path=usr/share/man/man3head/locale.h.3head
1867 file path=usr/share/man/man3head/math.h.3head
1868 file path=usr/share/man/man3head/mman.h.3head
1869 file path=usr/share/man/man3head/monetary.h.3head
1870 file path=usr/share/man/man3head/mqueue.h.3head
1871 file path=usr/share/man/man3head/msg.h.3head
1872 file path=usr/share/man/man3head/ndbm.h.3head
1873 file path=usr/share/man/man3head/netdb.h.3head
1874 file path=usr/share/man/man3head/nl_types.h.3head
1875 file path=usr/share/man/man3head/poll.h.3head
1876 file path=usr/share/man/man3head/pthread.h.3head
1877 file path=usr/share/man/man3head/pwd.h.3head
1878 file path=usr/share/man/man3head/regex.h.3head
1879 file path=usr/share/man/man3head/resource.h.3head
1880 file path=usr/share/man/man3head/sched.h.3head
1881 file path=usr/share/man/man3head/search.h.3head
1882 file path=usr/share/man/man3head/select.h.3head
1883 file path=usr/share/man/man3head/sem.h.3head
1884 file path=usr/share/man/man3head/semaphore.h.3head
1885 file path=usr/share/man/man3head/setjmp.h.3head
1886 file path=usr/share/man/man3head/shm.h.3head
1887 file path=usr/share/man/man3head/siginfo.h.3head
1888 file path=usr/share/man/man3head/signal.h.3head
1889 file path=usr/share/man/man3head/socket.h.3head
1890 file path=usr/share/man/man3head/spawn.h.3head
1891 file path=usr/share/man/man3head/stat.h.3head
1892 file path=usr/share/man/man3head/statvfs.h.3head
1893 file path=usr/share/man/man3head/stdbool.h.3head
1894 file path=usr/share/man/man3head/stddef.h.3head
1895 file path=usr/share/man/man3head/stdint.h.3head
1896 file path=usr/share/man/man3head/stdio.h.3head
1897 file path=usr/share/man/man3head/stdlib.h.3head
1898 file path=usr/share/man/man3head/string.h.3head
1899 file path=usr/share/man/man3head/strings.h.3head
1900 file path=usr/share/man/man3head/stropts.h.3head
1901 file path=usr/share/man/man3head/syslog.h.3head
1902 file path=usr/share/man/man3head/tar.h.3head
1903 file path=usr/share/man/man3head/tcp.h.3head
1904 file path=usr/share/man/man3head/termios.h.3head
1905 file path=usr/share/man/man3head/tgmath.h.3head
1906 file path=usr/share/man/man3head/time.h.3head
1907 file path=usr/share/man/man3head/timeb.h.3head
1908 file path=usr/share/man/man3head/times.h.3head

```

```

1909 file path=usr/share/man/man3head/types.h.3head
1910 file path=usr/share/man/man3head/types32.h.3head
1911 file path=usr/share/man/man3head/ucontext.h.3head
1912 file path=usr/share/man/man3head/uio.h.3head
1913 file path=usr/share/man/man3head/ulimit.h.3head
1914 file path=usr/share/man/man3head/un.h.3head
1915 file path=usr/share/man/man3head/unistd.h.3head
1916 file path=usr/share/man/man3head/utime.h.3head
1917 file path=usr/share/man/man3head/utmpx.h.3head
1918 file path=usr/share/man/man3head/utsname.h.3head
1919 file path=usr/share/man/man3head/values.h.3head
1920 file path=usr/share/man/man3head/wait.h.3head
1921 file path=usr/share/man/man3head/wchar.h.3head
1922 file path=usr/share/man/man3head/wctype.h.3head
1923 file path=usr/share/man/man3head/wordep.h.3head
1924 file path=usr/share/man/man3head/xlocale.h.3head
1925 file path=usr/share/man/man4/note.4
1926 file path=usr/share/man/man5/prof.5
1927 file path=usr/share/man/man7i/cdio.7i
1928 file path=usr/share/man/man7i/dkio.7i
1929 file path=usr/share/man/man7i/fbio.7i
1930 file path=usr/share/man/man7i/fdio.7i
1931 file path=usr/share/man/man7i/hdio.7i
1932 file path=usr/share/man/man7i/iec61883.7i
1933 file path=usr/share/man/man7i/mhd.7i
1934 file path=usr/share/man/man7i/mtio.7i
1935 file path=usr/share/man/man7i/prnio.7i
1936 file path=usr/share/man/man7i/quotactl.7i
1937 file path=usr/share/man/man7i/sesio.7i
1938 file path=usr/share/man/man7i/sockio.7i
1939 file path=usr/share/man/man7i/streamio.7i
1940 file path=usr/share/man/man7i/termio.7i
1941 file path=usr/share/man/man7i/termiox.7i
1942 file path=usr/share/man/man7i/uscsi.7i
1943 file path=usr/share/man/man7i/visual_io.7i
1944 file path=usr/share/man/man7i/vt.7i
1945 file path=usr/xpg4/include/curses.h
1946 file path=usr/xpg4/include/term.h
1947 file path=usr/xpg4/include/unctrl.h
1948 legacy pkg=SUNWhea \
1949 desc="SunOS C/C++ header files for general development of software" \
1950 name="SunOS Header Files"
1951 license cr_Sun license=cr_Sun
1952 license lic_CDDL license=lic_CDDL
1953 license license_in_headers license=license_in_headers
1954 license usr/src/lib/pkcs11/include/THIRDPARTYLICENSE \
1955 license=usr/src/lib/pkcs11/include/THIRDPARTYLICENSE
1956 license usr/src/uts/common/sys/THIRDPARTYLICENSE.firmload \
1957 license=usr/src/uts/common/sys/THIRDPARTYLICENSE.firmload
1958 link path=usr/include/iso/assert_iso.h target=./assert.h
1959 link path=usr/include/iso/errno_iso.h target=./errno.h
1960 link path=usr/include/iso/float_iso.h target=./float.h
1961 link path=usr/include/iso/iso646_iso.h target=./iso646.h
1962 $(sparc_ONLY)link path=usr/platform/SUNW,A70/include target=./sun4u/include
1963 $(sparc_ONLY)link path=usr/platform/SUNW,Netra-T12/include \
1964 target=./sun4u/include
1965 $(sparc_ONLY)link path=usr/platform/SUNW,Netra-T4/include \
1966 target=./sun4u/include
1967 $(sparc_ONLY)link path=usr/platform/SUNW,SPARC-Enterprise/include \
1968 target=./sun4u/include
1969 $(sparc_ONLY)link path=usr/platform/SUNW,Serverblad1/include \
1970 target=./sun4u/include
1971 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-100/include \
1972 target=./sun4u/include
1973 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1000/include \
1974 target=./sun4u/include

```

```

1975 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-1500/include \
1976     target=../sun4u/include
1977 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Blade-2500/include \
1978     target=../sun4u/include
1979 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-15000/include \
1980     target=../sun4u/include
1981 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-280R/include \
1982     target=../sun4u/include
1983 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-480R/include \
1984     target=../sun4u/include
1985 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-880/include \
1986     target=../sun4u/include
1987 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V215/include \
1988     target=../sun4u/include
1989 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V240/include \
1990     target=../sun4u/include
1991 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V250/include \
1992     target=../sun4u/include
1993 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V440/include \
1994     target=../sun4u/include
1995 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V445/include \
1996     target=../sun4u/include
1997 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V490/include \
1998     target=../sun4u/include
1999 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire-V890/include \
2000     target=../sun4u/include
2001 $(sparc_ONLY)link path=usr/platform/SUNW,Sun-Fire/include \
2002     target=../sun4u/include
2003 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-2/include \
2004     target=../sun4u/include
2005 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-250/include \
2006     target=../sun4u/include
2007 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-4/include \
2008     target=../sun4u/include
2009 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise-10000/include \
2010     target=../sun4u/include
2011 $(sparc_ONLY)link path=usr/platform/SUNW,Ultra-Enterprise/include \
2012     target=../sun4u/include
2013 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-40/include \
2014     target=../sun4u/include
2015 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-IIe-NetraCT-60/include \
2016     target=../sun4u/include
2017 $(sparc_ONLY)link path=usr/platform/SUNW,UltraSPARC-III-Netract/include \
2018     target=../sun4u/include
2019 link path=usr/share/man/man3head/acct.3head target=acct.h.3head
2020 link path=usr/share/man/man3head/aio.3head target=aio.h.3head
2021 link path=usr/share/man/man3head/ar.3head target=ar.h.3head
2022 link path=usr/share/man/man3head/archives.3head target=archives.h.3head
2023 link path=usr/share/man/man3head/assert.3head target=assert.h.3head
2024 link path=usr/share/man/man3head/complex.3head target=complex.h.3head
2025 link path=usr/share/man/man3head/cpio.3head target=cpio.h.3head
2026 link path=usr/share/man/man3head/dirent.3head target=dirent.h.3head
2027 link path=usr/share/man/man3head/errno.3head target=errno.h.3head
2028 link path=usr/share/man/man3head/fcntl.3head target=fcntl.h.3head
2029 link path=usr/share/man/man3head/fenv.3head target=fenv.h.3head
2030 link path=usr/share/man/man3head/float.3head target=float.h.3head
2031 link path=usr/share/man/man3head/floatingpoint.3head \
2032     target=floatingpoint.h.3head
2033 link path=usr/share/man/man3head/fmtmsg.3head target=fmtmsg.h.3head
2034 link path=usr/share/man/man3head/fnmatch.3head target=fnmatch.h.3head
2035 link path=usr/share/man/man3head/ftw.3head target=ftw.h.3head
2036 link path=usr/share/man/man3head/glob.3head target=glob.h.3head
2037 link path=usr/share/man/man3head/grp.3head target=grp.h.3head
2038 link path=usr/share/man/man3head/iconv.3head target=iconv.h.3head
2039 link path=usr/share/man/man3head/if.3head target=if.h.3head
2040 link path=usr/share/man/man3head/in.3head target=in.h.3head

```

```

2041 link path=usr/share/man/man3head/inet.3head target=inet.h.3head
2042 link path=usr/share/man/man3head/inttypes.3head target=inttypes.h.3head
2043 link path=usr/share/man/man3head/ipc.3head target=ipc.h.3head
2044 link path=usr/share/man/man3head/iso646.3head target=iso646.h.3head
2045 link path=usr/share/man/man3head/langinfo.3head target=langinfo.h.3head
2046 link path=usr/share/man/man3head/libgen.3head target=libgen.h.3head
2047 link path=usr/share/man/man3head/libintl.3head target=libintl.h.3head
2048 link path=usr/share/man/man3head/limits.3head target=limits.h.3head
2049 link path=usr/share/man/man3head/locale.3head target=locale.h.3head
2050 link path=usr/share/man/man3head/math.3head target=math.h.3head
2051 link path=usr/share/man/man3head/mman.3head target=mman.h.3head
2052 link path=usr/share/man/man3head/monetary.3head target=monetary.h.3head
2053 link path=usr/share/man/man3head/mqueue.3head target=mqueue.h.3head
2054 link path=usr/share/man/man3head/msg.3head target=msg.h.3head
2055 link path=usr/share/man/man3head/ndbm.3head target=ndbm.h.3head
2056 link path=usr/share/man/man3head/netdb.3head target=netdb.h.3head
2057 link path=usr/share/man/man3head/nl_types.3head target=nl_types.h.3head
2058 link path=usr/share/man/man3head/poll.3head target=poll.h.3head
2059 link path=usr/share/man/man3head/pthread.3head target=pthread.h.3head
2060 link path=usr/share/man/man3head/pwd.3head target=pwd.h.3head
2061 link path=usr/share/man/man3head/regex.3head target=regex.h.3head
2062 link path=usr/share/man/man3head/resource.3head target=resource.h.3head
2063 link path=usr/share/man/man3head/sched.3head target=sched.h.3head
2064 link path=usr/share/man/man3head/search.3head target=search.h.3head
2065 link path=usr/share/man/man3head/select.3head target=select.h.3head
2066 link path=usr/share/man/man3head/sem.3head target=sem.h.3head
2067 link path=usr/share/man/man3head/semaphore.3head target=semaphore.h.3head
2068 link path=usr/share/man/man3head/setjmp.3head target=setjmp.h.3head
2069 link path=usr/share/man/man3head/shm.3head target=shm.h.3head
2070 link path=usr/share/man/man3head/siginfo.3head target=siginfo.h.3head
2071 link path=usr/share/man/man3head/signal.3head target=signal.h.3head
2072 link path=usr/share/man/man3head/socket.3head target=socket.h.3head
2073 link path=usr/share/man/man3head/spawn.3head target=spawn.h.3head
2074 link path=usr/share/man/man3head/stat.3head target=stat.h.3head
2075 link path=usr/share/man/man3head/statvfs.3head target=statvfs.h.3head
2076 link path=usr/share/man/man3head/stdbool.3head target=stdbool.h.3head
2077 link path=usr/share/man/man3head/stddef.3head target=stddef.h.3head
2078 link path=usr/share/man/man3head/stdint.3head target=stdint.h.3head
2079 link path=usr/share/man/man3head/stdio.3head target=stdio.h.3head
2080 link path=usr/share/man/man3head/stdlib.3head target=stdlib.h.3head
2081 link path=usr/share/man/man3head/string.3head target=string.h.3head
2082 link path=usr/share/man/man3head/strings.3head target=strings.h.3head
2083 link path=usr/share/man/man3head/stropts.3head target=stropts.h.3head
2084 link path=usr/share/man/man3head/syslog.3head target=syslog.h.3head
2085 link path=usr/share/man/man3head/tar.3head target=tar.h.3head
2086 link path=usr/share/man/man3head/tcp.3head target=tcp.h.3head
2087 link path=usr/share/man/man3head/termios.3head target=termios.h.3head
2088 link path=usr/share/man/man3head/tgmath.3head target=tgmath.h.3head
2089 link path=usr/share/man/man3head/time.3head target=time.h.3head
2090 link path=usr/share/man/man3head/timeb.3head target=timeb.h.3head
2091 link path=usr/share/man/man3head/times.3head target=times.h.3head
2092 link path=usr/share/man/man3head/types.3head target=types.h.3head
2093 link path=usr/share/man/man3head/types32.3head target=types32.h.3head
2094 link path=usr/share/man/man3head/ucontext.3head target=ucontext.h.3head
2095 link path=usr/share/man/man3head/uoio.3head target=uoio.h.3head
2096 link path=usr/share/man/man3head/ulimit.3head target=ulimit.h.3head
2097 link path=usr/share/man/man3head/un.3head target=un.h.3head
2098 link path=usr/share/man/man3head/unistd.3head target=unistd.h.3head
2099 link path=usr/share/man/man3head/utime.3head target=utime.h.3head
2100 link path=usr/share/man/man3head/utmpx.3head target=utmpx.h.3head
2101 link path=usr/share/man/man3head/utsname.3head target=utsname.h.3head
2102 link path=usr/share/man/man3head/values.3head target=values.h.3head
2103 link path=usr/share/man/man3head/wait.3head target=wait.h.3head
2104 link path=usr/share/man/man3head/wchar.3head target=wchar.h.3head
2105 link path=usr/share/man/man3head/wctype.3head target=wctype.h.3head
2106 link path=usr/share/man/man3head/wordexp.3head target=wordexp.h.3head

```



```
2107 link path=usr/share/man/man3head/xlocale.3head target=xlocale.h.3head
2108 $(i386_ONLY)link path=usr/share/src/uts/i86pc/sys \
2109     target=../../../../platform/i86pc/include/sys
2110 $(i386_ONLY)link path=usr/share/src/uts/i86pc/vm \
2111     target=../../../../platform/i86pc/include/vm
2112 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/sys \
2113     target=../../../../platform/i86xpv/include/sys
2114 $(i386_ONLY)link path=usr/share/src/uts/i86xpv/vm \
2115     target=../../../../platform/i86xpv/include/vm
2116 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/sys \
2117     target=../../../../platform/sun4u/include/sys
2118 $(sparc_ONLY)link path=usr/share/src/uts/sun4u/vm \
2119     target=../../../../platform/sun4u/include/vm
2120 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/sys \
2121     target=../../../../platform/sun4v/include/sys
2122 $(sparc_ONLY)link path=usr/share/src/uts/sun4v/vm \
2123     target=../../../../platform/sun4v/include/vm
```

```

*****
47324 Tue Nov 20 12:08:34 2018
new/usr/src/uts/common/Makefile.files
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1991, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2011, 2014 by Delphix. All rights reserved.
25 # Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
26 # Copyright 2018 Nexenta Systems, Inc.
27 # Copyright 2016 Garrett D'Amore <garrett@damore.org>
28 # Copyright (c) 2017, Joyent, Inc.
29 # Copyright 2016 OmniTI Computer Consulting, Inc. All rights reserved.
30 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
31 #
32 #
33 #
34 # This Makefile defines all file modules for the directory uts/common
35 # and its children. These are the source files which may be considered
36 # common to all SunOS systems.
37 #
38 i386_CORE_OBJS += \
39     atomic.o      \
40     avintr.o     \
41     pic.o
42 #
43 sparc_CORE_OBJS +=
44 #
45 COMMON_CORE_OBJS += \
46     beep.o       \
47     bitset.o    \
48     bp_map.o    \
49     brand.o     \
50     cpucaps.o   \
51     cmt.o       \
52     cmt_policy.o \
53     cpu.o       \
54     cpu_uarray.o \
55     cpu_event.o \
56     cpu_intr.o  \
57     cpu_pm.o    \
58     cpupart.o  \
59     cap_util.o  \
60     disp.o

```

```

61     group.o      \
62     kstat_fr.o  \
63     iscsiboot_prop.o \
64     lgrp.o       \
65     lgrp_topo.o  \
66     mmapobj.o   \
67     mutex.o     \
68     page_lock.o \
69     page_retire.o \
70     panic.o     \
71     param.o     \
72     pg.o        \
73     pghw.o     \
74     putnext.o  \
75     rctl_proc.o \
76     rwlock.o   \
77     seg_kmem.o  \
78     softint.o   \
79     string.o    \
80     strtol.o    \
81     strtoul.o   \
82     strtoll.o   \
83     strtoull.o  \
84     thread_intr.o \
85     vm_page.o   \
86     vm_pagelist.o \
87     zlib_obj.o  \
88     clock_tick.o
89 #
90 CORE_OBJS += $(COMMON_CORE_OBJS) $(MACH)_CORE_OBJS
91 #
92 ZLIB_OBJS = zutil.o zmod.o zmod_subr.o \
93     adler32.o crc32.o deflate.o infast.o \
94     inflate.o inftrees.o trees.o
95 #
96 GENUNIX_OBJS += \
97     access.o \
98     acl.o \
99     acl_common.o \
100    adjtime.o \
101    alarm.o \
102    aio_subr.o \
103    auditsys.o \
104    audit_core.o \
105    audit_zone.o \
106    audit_memory.o \
107    autoconf.o \
108    avl.o \
109    bdev_dsort.o \
110    bio.o \
111    bitmap.o \
112    blabel.o \
113    brandsys.o \
114    bz2blocksort.o \
115    bz2compress.o \
116    bz2decompress.o \
117    bz2randtable.o \
118    bz2bzlib.o \
119    bz2crctable.o \
120    bz2huffman.o \
121    callb.o \
122    callout.o \
123    chdir.o \
124    chmod.o \
125    chown.o \
126    cladm.o

```

new/usr/src/uts/common/Makefile.files

```

127      class.o      \
128      clock.o      \
129      clock_highres.o \
130      clock_realtime.o \
131      close.o      \
132      compress.o   \
133      condvar.o    \
134      conf.o       \
135      console.o    \
136      contract.o   \
137      copyops.o    \
138      core.o       \
139      corectl.o    \
140      cred.o       \
141      cs_stubs.o   \
142      dacf.o       \
143      dacf_clnt.o  \
144      damap.o      \
145      cyclic.o     \
146      ddi.o        \
147      ddifm.o     \
148      ddi_hp_impl.o \
149      ddi_hp_ndi.o \
150      ddi_intr.o   \
151      ddi_intr_impl.o \
152      ddi_intr_irm.o \
153      ddi_nodeid.o \
154      ddi_periodic.o \
155      devcfg.o     \
156      devcache.o  \
157      device.o     \
158      devid.o      \
159      devid_cache.o \
160      devid_scsi.o \
161      devid_smp.o  \
162      devpolicy.o \
163      disp_lock.o  \
164      dkioc_free_util.o \
165      dnlc.o       \
166      driver.o     \
167      dumpsubr.o   \
168      driver_lyr.o \
169      dtrace_subr.o \
170      errorq.o     \
171      etheraddr.o  \
172      evchannels.o \
173      exacct.o     \
174      exacct_core.o \
175      exec.o       \
176      exit.o       \
177      fbio.o       \
178      fcntl.o      \
179      fdbuffer.o   \
180      fdsync.o     \
181      fem.o        \
182      ffs.o        \
183      fio.o        \
184      firmload.o   \
185      flock.o      \
186      fm.o         \
187      fork.o       \
188      vpm.o        \
189      fs_reparse.o \
190      fs_subr.o    \
191      fsflush.o    \
192      ftrace.o     \

```

3

new/usr/src/uts/common/Makefile.files

```

193      getcwd.o     \
194      getdents.o   \
195      getloadavg.o \
196      getpagesizes.o \
197      getpid.o     \
198      gfs.o        \
199      rusagesys.o  \
200      gid.o        \
201      groups.o     \
202      grow.o       \
203      hat_refmod.o \
204      id32.o       \
205      id_space.o   \
206      inet_ntop.o  \
207      instance.o   \
208      ioctl.o      \
209      ip_cksum.o   \
210      issetugid.o  \
211      ipppconf.o   \
212      kpcp.o       \
213      kdi.o        \
214      kiconv.o     \
215      klpd.o       \
216      kmem.o       \
217      ksyms_snapshot.o \
218      l_strplumb.o \
219      labelsys.o   \
220      link.o       \
221      list.o       \
222      lockstat_subr.o \
223      log_sysevent.o \
224      logsubr.o    \
225      lookup.o     \
226      lseek.o      \
227      ltos.o       \
228      lwp.o        \
229      lwp_create.o \
230      lwp_info.o   \
231      lwp_self.o   \
232      lwp_sobj.o   \
233      lwp_timer.o  \
234      lwpsys.o     \
235      main.o       \
236      mmapobjsys.o \
237      memcntl.o    \
238      memstr.o     \
239      lgrpsys.o    \
240      mkdir.o      \
241      mknod.o      \
242      mount.o      \
243      move.o       \
244      msacct.o     \
245      multidata.o  \
246      nbmlock.o   \
247      ndifm.o     \
248      nice.o       \
249      netstack.o   \
250      ntptime.o   \
251      nvpair.o     \
252      nvpair_alloc_system.o \
253      nvpair_alloc_fixed.o \
254      fnvpair.o    \
255      octet.o      \
256      open.o       \
257      p_online.o   \
258      pathconf.o  \

```

4

new/usr/src/uts/common/Makefile.files

```

259      pathname.o  \
260      pause.o    \
261      serializer.o \
262      pci_intr_lib.o \
263      pci_cap.o  \
264      pcifm.o   \
265      pgrp.o    \
266      pgrpsys.o \
267      pid.o     \
268      pkp_hash.o \
269      policy.o  \
270      poll.o    \
271      pool.o    \
272      pool_pset.o \
273      port_subr.o \
274      ppriv.o   \
275      printf.o  \
276      priocntl.o \
277      priv.o    \
278      priv_const.o \
279      proc.o    \
280      psecflags.o \
281      procset.o \
282      processor_bind.o \
283      processor_info.o \
284      profil.o  \
285      project.o \
286      qsort.o   \
287      getrandom.o \
288      rctl.o    \
289      rctlsys.o \
290      readlink.o \
291      refstr.o  \
292      rename.o  \
293      resolvepath.o \
294      retire_store.o \
295      process.o \
296      rlimit.o  \
297      rmap.o    \
298      rw.o     \
299      rwstlock.o \
300      sad_conf.o \
301      sid.o    \
302      sidsys.o \
303      sched.o  \
304      schedctl.o \
305      sctp_crc32.o \
306      secflags.o \
307      seg_dev.o \
308      seg_hole.o \
309      seg_kp.o  \
310      seg_kpm.o \
311      seg_map.o \
312      seg_vn.o  \
313      seg_spt.o \
314      seg_umap.o \
315      semaphore.o \
316      sendfile.o \
317      session.o \
318      share.o  \
319      shuttle.o \
320      sig.o    \
321      sigaction.o \
322      sigaltstack.o \
323      sigotify.o \
324      sigpending.o \

```

5

new/usr/src/uts/common/Makefile.files

```

325      sigprocmask.o \
326      sigqueue.o   \
327      sigsendset.o \
328      sigsuspend.o \
329      sigtimedwait.o \
330      sleepq.o     \
331      sock_conf.o  \
332      space.o      \
333      sscanf.o     \
334      stat.o       \
335      statfs.o    \
336      statvfs.o   \
337      stol.o      \
338      str_conf.o   \
339      strcalls.o  \
340      stream.o     \
341      streamio.o  \
342      strext.o    \
343      strsubr.o   \
344      strsun.o    \
345      subr.o      \
346      sunddi.o   \
347      sunmdi.o   \
348      sunndi.o   \
349      sunpci.o   \
350      sunpm.o    \
351      sundlpi.o  \
352      suntpi.o   \
353      swap_subr.o \
354      swap_vnops.o \
355      symlink.o   \
356      sync.o     \
357      sysclass.o \
358      sysconfig.o \
359      sysent.o   \
360      sysfs.o    \
361      systeminfo.o \
362      task.o     \
363      taskq.o   \
364      tasksys.o \
365      time.o    \
366      timer.o   \
367      times.o   \
368      timers.o  \
369      thread.o  \
370      tlabel.o  \
371      tnf_res.o \
372      turnstile.o \
373      tty_common.o \
374      u8_textprep.o \
375      uadmin.o  \
376      uconv.o   \
377      ucredsys.o \
378      uid.o     \
379      umask.o   \
380      umount.o  \
381      uname.o   \
382      unix_bb.o \
383      unlink.o  \
384      urw.o     \
385      utime.o   \
386      utssys.o  \
387      uucopy.o  \
388      vfs.o     \
389      vfs_conf.o \
390      vmem.o    \

```

6

new/usr/src/uts/common/Makefile.files

7

```

391          vm_anon.o      \
392          vm_as.o        \
393          vm_meter.o     \
394          vm_pageout.o   \
395          vm_pvn.o       \
396          vm_rm.o        \
397          vm_seg.o       \
398          vm_subr.o      \
399          vm_swap.o      \
400          vm_usage.o     \
401          vnode.o        \
402          vuid_queue.o   \
403          vuid_store.o   \
404          waitq.o        \
405          watchpoint.o   \
406          yield.o        \
407          scsi_confdata.o \
408          xattr.o        \
409          xattr_common.o \
410          xdr_mblk.o     \
411          xdr_mem.o      \
412          xdr.o          \
413          xdr_array.o    \
414          xdr_refer.o    \
415          zone.o

417 #
418 #       Stubs for the stand-alone linker/loader
419 #
420 sparc_GENSTUBS_OBJS = \
421     kobj_stubs.o

423 i386_GENSTUBS_OBJS =

425 COMMON_GENSTUBS_OBJS =

427 GENSTUBS_OBJS += $(COMMON_GENSTUBS_OBJS) $($ (MACH)_GENSTUBS_OBJS)

429 #
430 #       DTrace and DTrace Providers
431 #
432 DTRACE_OBJS += dtrace.o dtrace_isa.o dtrace_asm.o

434 SDT_OBJS += sdt_subr.o

436 PROFILE_OBJS += profile.o

438 SYSTRACE_OBJS += systrace.o

440 LOCKSTAT_OBJS += lockstat.o

442 FASTTRAP_OBJS += fasttrap.o fasttrap_isa.o

444 DCPC_OBJS += dcpc.o

446 #
447 #       Driver (pseudo-driver) Modules
448 #
449 IPP_OBJS += ippctl.o

451 AUDIO_OBJS += audio_client.o audio_ddi.o audio_engine.o \
452     audio_fltdata.o audio_format.o audio_ctrl.o \
453     audio_grc3.o audio_output.o audio_input.o \
454     audio_oss.o audio_sun.o

456 AUDIOEMU10K_OBJS += audioemu10k.o

```

new/usr/src/uts/common/Makefile.files

8

```

458 AUDIOENS_OBJS += audioens.o

460 AUDIOVIA823X_OBJS += audiovia823x.o

462 AUDIO1575_OBJS += audio1575.o

464 AUDIO810_OBJS += audio810.o

466 AUDIOCMI_OBJS += audiocmi.o

468 AUDIOCMIHD_OBJS += audiocmihd.o

470 AUDIOHD_OBJS += audiohd.o

472 AUDIOIXP_OBJS += audioixp.o

474 AUDIOLS_OBJS += audiols.o

476 AUDIOP16X_OBJS += audiop16x.o

478 AUDIOPCI_OBJS += audiopci.o

480 AUDIOSOLO_OBJS += audiosolo.o

482 AUDIOTS_OBJS += audiotls.o

484 AC97_OBJS += ac97.o ac97_ad.o ac97_alc.o ac97_cmi.o

486 BLKDEV_OBJS += blkdev.o

488 CARDBUS_OBJS += cardbus.o cardbus_hp.o cardbus_cfg.o

490 CONSKBD_OBJS += conskbd.o

492 CONSMS_OBJS += consms.o

494 OLDPTY_OBJS += tty_ptyconf.o

496 PTC_OBJS += tty_pty.o

498 PTSL_OBJS += tty_pts.o

500 PTM_OBJS += ptm.o

502 MII_OBJS += mii.o mii_cicada.o mii_natsemi.o mii_intel.o mii_qualsemi.o \
503     mii_marvell.o mii_realtek.o mii_other.o

505 PTS_OBJS += pts.o

507 PTY_OBJS += ptms_conf.o

509 SAD_OBJS += sad.o

511 MD4_OBJS += md4.o md4_mod.o

513 MD5_OBJS += md5.o md5_mod.o

515 SHA1_OBJS += sha1.o sha1_mod.o

517 SHA2_OBJS += sha2.o sha2_mod.o

519 SKEIN_OBJS += skein.o skein_block.o skein_iv.o skein_mod.o

521 EDONR_OBJS += edonr.o edonr_mod.o

```

```

523 IPGPC_OBJS += classifierddi.o classifier.o filters.o trie.o table.o \
524 ba_table.o

526 DSCPMK_OBJS += dscpmk.o dscpmkddi.o

528 DLCOSMK_OBJS += dlcosmk.o dlcosmkddi.o

530 FLOWACCT_OBJS += flowacctddi.o flowacct.o

532 TOKENMT_OBJS += tokenmt.o tokenmtddi.o

534 TSWTCL_OBJS += tswtcl.o tswtclddi.o

536 ARP_OBJS += arpddi.o

538 ICMP_OBJS += icmpddi.o

540 ICMP6_OBJS += icmp6ddi.o

542 RTS_OBJS += rtsddi.o

544 IP_ICMP_OBJS = icmp.o icmp_opt_data.o
545 IP_RTS_OBJS = rts.o rts_opt_data.o
546 IP_TCP_OBJS = tcp.o tcp_fusion.o tcp_opt_data.o tcp_sack.o tcp_stats.o \
547 tcp_misc.o tcp_timers.o tcp_time_wait.o tcp_tpi.o tcp_output.o \
548 tcp_input.o tcp_socket.o tcp_bind.o tcp_cluster.o tcp_tunables.o
549 IP_UDP_OBJS = udp.o udp_opt_data.o udp_tunables.o udp_stats.o
550 IP_SCTP_OBJS = sctp.o sctp_opt_data.o sctp_output.o \
551 sctp_init.o sctp_input.o sctp_cookie.o \
552 sctp_conn.o sctp_error.o sctp_snmp.o \
553 sctp_tunables.o sctp_shutdown.o sctp_common.o \
554 sctp_timer.o sctp_heartbeat.o sctp_hash.o \
555 sctp_bind.o sctp_notify.o sctp_asconf.o \
556 sctp_addr.o tn_ipopt.o tnet.o ip_netinfo.o \
557 sctp_misc.o
558 IP_ILB_OBJS = ilb.o ilb_nat.o ilb_conn.o ilb_alg_hash.o ilb_alg_rr.o

560 IP_OBJS += igmp.o ipmp.o ip.o ip6.o ip6_asp.o ip6_if.o ip6_ire.o \
561 ip6_rts.o ip_if.o ip_ire.o ip_listutils.o ip_mrout.o \
562 ip_multi.o ip2mac.o ip_ndp.o ip_rts.o ip_srcid.o \
563 ipddi.o ipdrop.o mi.o nd.o tunables.o optcom.o snmpcom.o \
564 ipsec_loader.o spd.o ipclassifier.o inet_common.o ip_queue.o \
565 squeue.o ip_sadb.o ip_ftable.o proto_set.o radix.o ip_dummy.o \
566 ip_helper_stream.o ip_tunables.o \
567 ip_output.o ip_input.o ip6_input.o ip6_output.o ip_arp.o \
568 conn_opt.o ip_attr.o ip_dce.o \
569 $(IP_ICMP_OBJS) \
570 $(IP_RTS_OBJS) \
571 $(IP_TCP_OBJS) \
572 $(IP_UDP_OBJS) \
573 $(IP_SCTP_OBJS) \
574 $(IP_ILB_OBJS)

576 IP6_OBJS += ip6ddi.o

578 HOOK_OBJS += hook.o

580 NETI_OBJS += neti_impl.o neti_mod.o neti_stack.o

582 KEYSOCK_OBJS += keysockddi.o keysock.o keysock_opt_data.o

584 IPNET_OBJS += ipnet.o ipnet_bpf.o

586 SPDSOCK_OBJS += spdsockddi.o spdsock.o spdsock_opt_data.o

588 IPSECESP_OBJS += ipsecespddi.o ipsecesp.o

```

```

590 IPSECAH_OBJS += ipsecahddi.o ipsecah.o sadb.o

592 SPPP_OBJS += sPPP.o sPPP_dlpi.o sPPP_mod.o s_common.o

594 SPPPTUN_OBJS += sppptun.o sppptun_mod.o

596 SPPPASYN_OBJS += spppasyn.o spppasyn_mod.o

598 SPPPCOMP_OBJS += sppppcomp.o sppppcomp_mod.o deflate.o BSD-comp.o vjcompress.o \
599 zlib.o

601 TCP_OBJS += tcpddi.o

603 TCP6_OBJS += tcp6ddi.o

605 NCA_OBJS += ncaddi.o

607 SDP SOCK_MOD_OBJS += sockmod_sdp.o socksdp.o socksdp_subr.o

609 SCTP SOCK_MOD_OBJS += sockmod_sctp.o socksctp.o socksctp_subr.o

611 PFP SOCK_MOD_OBJS += sockmod_pfp.o

613 RDS SOCK_MOD_OBJS += sockmod_rds.o

615 RDS_OBJS += rdsddi.o rdssubr.o rds_opt.o rds_ioctl.o

617 RDSIB_OBJS += rdsib.o rdsib_ib.o rdsib_cm.o rdsib_ep.o rdsib_buf.o \
618 rdsib_debug.o rdsib_sc.o

620 RDSV3_OBJS += af_rds.o rds_v3_ddi.o bind.o loop.o threads.o connection.o \
621 transport.o cong.o sysctl.o message.o rds_recv.o send.o \
622 stats.o info.o page.o rdma_transport.o ib_ring.o ib_rdma.o \
623 ib_recv.o ib.o ib_send.o ib_sysctl.o ib_stats.o ib_cm.o \
624 rds_v3_sc.o rds_v3_debug.o rds_v3_impl.o rdma.o rds_v3_af_thr.o

626 ISER_OBJS += iser.o iser_cm.o iser_cq.o iser_ib.o iser_idm.o \
627 iser_resource.o iser_xfer.o

629 UDP_OBJS += udpddi.o

631 UDP6_OBJS += udp6ddi.o

633 SY_OBJS += gentyty.o

635 TCO_OBJS += ticots.o

637 TCOO_OBJS += ticotsord.o

639 TCL_OBJS += ticlts.o

641 TL_OBJS += tl.o

643 DUMP_OBJS += dump.o

645 BPF_OBJS += bpf.o bpf_filter.o bpf_mod.o bpf_dlt.o bpf_mac.o

647 CLONE_OBJS += clone.o

649 CN_OBJS += cons.o

651 DLD_OBJS += dld_drv.o dld_proto.o dld_str.o dld_flow.o

653 DLS_OBJS += dls.o dls_link.o dls_mod.o dls_stat.o dls_mgmt.o

```

```

655 GLD_OBJS +=      gld.o gldutil.o
657 MAC_OBJS +=      mac.o mac_bcast.o mac_client.o mac_datapath_setup.o mac_flow.o
658                 mac_hio.o mac_mod.o mac_ndd.o mac_provider.o mac_sched.o \
659                 mac_protect.o mac_soft_ring.o mac_stat.o mac_util.o
661 MAC_6TO4_OBJS +=      mac_6to4.o
663 MAC_ETHER_OBJS +=      mac_ether.o
665 MAC_IPV4_OBJS +=      mac_ipv4.o
667 MAC_IPV6_OBJS +=      mac_ipv6.o
669 MAC_WIFI_OBJS +=      mac_wifi.o
671 MAC_IB_OBJS +=      mac_ib.o
673 IPTUN_OBJS +=      iptun_dev.o iptun_ctl.o iptun.o
675 AGGR_OBJS +=      aggr_dev.o aggr_ctl.o aggr_grp.o aggr_port.o \
676                 aggr_send.o aggr_recv.o aggr_lacp.o
678 SOFTMAC_OBJS +=      softmac_main.o softmac_ctl.o softmac_capab.o \
679                 softmac_dev.o softmac_stat.o softmac_pkt.o softmac_fp.o
681 NET80211_OBJS +=      net80211.o net80211_proto.o net80211_input.o \
682                 net80211_output.o net80211_node.o net80211_crypto.o \
683                 net80211_crypto_none.o net80211_crypto_wep.o net80211_ioctl.o \
684                 net80211_crypto_tkip.o net80211_crypto_ccmp.o \
685                 net80211_ht.o net80211_amrr.o
687 VNIC_OBJS +=      vnic_ctl.o vnic_dev.o
689 SIMNET_OBJS +=      simnet.o
691 IB_OBJS +=      ibnex.o ibnex_ioctl.o ibnex_hca.o
693 IBCM_OBJS +=      ibcm_impl.o ibcm_sm.o ibcm_ti.o ibcm_utils.o ibcm_path.o \
694                 ibcm_arp.o ibcm_arp_link.o
696 IBDM_OBJS +=      ibdm.o
698 IBDMA_OBJS +=      ibdma.o
700 IBMF_OBJS +=      ibmf.o ibmf_impl.o ibmf_dr.o ibmf_wqe.o ibmf_ud_dest.o ibmf_mod.o
701                 ibmf_send.o ibmf_recv.o ibmf_handlers.o ibmf_trans.o \
702                 ibmf_timers.o ibmf_msg.o ibmf_utils.o ibmf_rmpp.o \
703                 ibmf_saa.o ibmf_saa_impl.o ibmf_saa_utils.o ibmf_saa_events.o
705 IBTL_OBJS +=      ibtl_impl.o ibtl_util.o ibtl_mem.o ibtl_handlers.o ibtl_qp.o \
706                 ibtl_cq.o ibtl_wr.o ibtl_hca.o ibtl_chan.o ibtl_cm.o \
707                 ibtl_mcg.o ibtl_ibnex.o ibtl_srqp.o ibtl_part.o
709 TAVOR_OBJS +=      tavor.o tavor_agents.o tavor_cfg.o tavor_ci.o tavor_cmd.o \
710                 tavor_cq.o tavor_event.o tavor_ioctl.o tavor_misc.o \
711                 tavor_mr.o tavor_qp.o tavor_qpmod.o tavor_rsrc.o \
712                 tavor_srqp.o tavor_stats.o tavor_umap.o tavor_wr.o
714 HERMON_OBJS +=      hermon.o hermon_agents.o hermon_cfg.o hermon_ci.o hermon_cmd.o \
715                 hermon_cq.o hermon_event.o hermon_ioctl.o hermon_misc.o \
716                 hermon_mr.o hermon_qp.o hermon_qpmod.o hermon_rsrc.o \
717                 hermon_srqp.o hermon_stats.o hermon_umap.o hermon_wr.o \
718                 hermon_fcoib.o hermon_fm.o
720 DAPLT_OBJS +=      daplt.o

```

```

722 SOL_OFS_OBJS +=      sol_cma.o sol_ib_cma.o sol_uobj.o \
723                 sol_ofs_debug_util.o sol_ofs_gen_util.o \
724                 sol_kverbs.o
726 SOL_UCMA_OBJS +=      sol_ucma.o
728 SOL_UVERBS_OBJS +=      sol_uverbs.o sol_uverbs_comp.o sol_uverbs_event.o \
729                 sol_uverbs_hca.o sol_uverbs_qp.o
731 SOL_UMAD_OBJS +=      sol_umad.o
733 KSTAT_OBJS +=      kstat.o
735 KSYMS_OBJS +=      ksyms.o
737 INSTANCE_OBJS +=      inst_sync.o
739 IWSCN_OBJS +=      iwscns.o
741 LOFI_OBJS +=      lofi.o LzmaDec.o
743 FSSNAP_OBJS +=      fssnap.o
745 FSSNAPIF_OBJS +=      fssnap_if.o
747 MM_OBJS +=      mem.o
749 PHYSMEM_OBJS +=      physmem.o
751 OPTIONS_OBJS +=      options.o
753 WINLOCK_OBJS +=      winlockio.o
755 PM_OBJS +=      pm.o
756 SRN_OBJS +=      srn.o
758 PSEUDO_OBJS +=      pseudonex.o
760 RAMDISK_OBJS +=      ramdisk.o
762 LLC1_OBJS +=      llc1.o
764 USBKBM_OBJS +=      usbkbm.o
766 USBWCM_OBJS +=      usbwcm.o
768 BOFI_OBJS +=      bofi.o
770 HID_OBJS +=      hid.o
772 USBSKEL_OBJS +=      usbskel.o
774 USBVC_OBJS +=      usbvc.o usbvc_v412.o
776 HIDPARSER_OBJS +=      hidparser.o
778 USB_AC_OBJS +=      usb_ac.o
780 USB_AS_OBJS +=      usb_as.o
782 USB_AH_OBJS +=      usb_ah.o
784 USBMS_OBJS +=      usbms.o
786 USBPRN_OBJS +=      usbprn.o

```

```

788 UGEN_OBJS += ugen.o
790 USBSER_OBJS += usbser.o usbser_rseq.o
792 USBSACM_OBJS += usbasm.o
794 USBSER_KEYSPAN_OBJS += usbser_keyspan.o keyspan_dsd.o keyspan_pipe.o
796 USBS49_FW_OBJS += keyspan_49fw.o
798 USBSPRL_OBJS += usbser_pl2303.o pl2303_dsd.o
800 USBFTDI_OBJS += usbser_uftdi.o uftdi_dsd.o
802 USBECM_OBJS += usbecm.o
804 WC_OBJS += wscons.o vcons.o
806 VCONS_CONF_OBJS += vcons_conf.o
808 SCSI_OBJS +=      scsi_capabilities.o scsi_confsubr.o scsi_control.o \
809                  scsi_data.o scsi_fm.o scsi_hba.o scsi_reset_notify.o \
810                  scsi_resource.o scsi_subr.o scsi_transport.o scsi_watch.o \
811                  smp_transport.o
813 SCSI_VHCI_OBJS +=      scsi_vhci.o mpapi_impl.o scsi_vhci_tpgs.o
815 SCSI_VHCI_F_SYM_OBJS +=      sym.o
817 SCSI_VHCI_F_TPGS_OBJS +=      tpgs.o
819 SCSI_VHCI_F_ASYM_SUN_OBJS +=  asym_sun.o
821 SCSI_VHCI_F_SYM_HDS_OBJS +=   sym_hds.o
823 SCSI_VHCI_F_TAPE_OBJS +=      tape.o
825 SCSI_VHCI_F_TPGS_TAPE_OBJS += tpgs_tape.o
827 SGEN_OBJS +=      sgen.o
829 SMP_OBJS +=      smp.o
831 SATA_OBJS +=      sata.o
833 USBA_OBJS +=      hcidi.o usba.o usbai.o hubdi.o parser.o genconsole.o \
834                  usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
835                  usba_devdb.o usba10_calls.o usba_ugen.o
837 USBA10_OBJS +=    usba10.o
839 RSM_OBJS +=      rsm.o rsmka_pathmanager.o rsmka_util.o
841 RSMOPS_OBJS +=   rsmops.o
843 S1394_OBJS +=     t1394.o t1394_errmsg.o s1394.o s1394_addr.o s1394_async.o \
844                  s1394_bus_reset.o s1394_cmp.o s1394_csr.o s1394_dev_disc.o \
845                  s1394_fa.o s1394_fcp.o \
846                  s1394_hotplug.o s1394_isoch.o s1394_misc.o h1394.o nx1394.o
848 HCI1394_OBJS +=   hcil1394.o hcil1394_async.o hcil1394_attach.o hcil1394_buf.o \
849                  hcil1394_csr.o hcil1394_detach.o hcil1394_extern.o \
850                  hcil1394_ioctl.o hcil1394_isoch.o hcil1394_isr.o \
851                  hcil1394_ixl_comp.o hcil1394_ixl_isr.o hcil1394_ixl_misc.o \
852                  hcil1394_ixl_update.o hcil1394_misc.o hcil1394_ohci.o \

```

```

853                  hcil1394_q.o hcil1394_sl394if.o hcil1394_tlabel.o \
854                  hcil1394_tlist.o hcil1394_vendor.o
856 AV1394_OBJS +=    avl1394.o avl1394_as.o avl1394_async.o avl1394_cfgrom.o \
857                  avl1394_cmp.o avl1394_fcp.o avl1394_isoch.o avl1394_isoch_chan.o \
858                  avl1394_isoch_recv.o avl1394_isoch_xmit.o avl1394_list.o \
859                  avl1394_queue.o
861 DCAM1394_OBJS +=  dcam.o dcam_frame.o dcam_param.o dcam_reg.o \
862                  dcam_ring_buff.o
864 SCSA1394_OBJS +=  hba.o sbp2_driver.o sbp2_bus.o
866 SBP2_OBJS +=      cfgrom.o sbp2.o
868 PMODEM_OBJS +=    pmodem.o pmodem_cis.o cis.o cis_callout.o cis_handlers.o cis_para
870 PMCS_OBJS +=      pmcs_attach.o pmcs_ds.o pmcs_intr.o pmcs_nvram.o pmcs_sata.o \
871                  pmcs_scsa.o pmcs_smhba.o pmcs_subr.o pmcs_fwlog.o
873 PMCS8001FW_C_OBJS +=      pmcs_fw_hdr.o
874 PMCS8001FW_OBJS +=        $(PMCS8001FW_C_OBJS) SPCBoot.o ila.o firmware.o
876 #
877 #      Build up defines and paths.
879 ST_OBJS +=      st.o st_conf.o
881 EMLXS_OBJS +=    emlxs_clock.o emlxs_dfc.o emlxs_dhchap.o emlxs_diag.o \
882                  emlxs_download.o emlxs_dump.o emlxs_els.o emlxs_event.o \
883                  emlxs_fcf.o emlxs_fcp.o emlxs_fct.o emlxs_hba.o emlxs_ip.o \
884                  emlxs_mbox.o emlxs_mem.o emlxs_msg.o emlxs_node.o \
885                  emlxs_pkt.o emlxs_sli3.o emlxs_sli4.o emlxs_solaris.o \
886                  emlxs_thread.o
888 EMLXS_FW_OBJS +=      emlxs_fw.o
890 OCE_OBJS +=          oce_buf.o oce_fm.o oce_gld.o oce_hw.o oce_intr.o oce_main.o \
891                  oce_mbx.o oce_mq.o oce_queue.o oce_rx.o oce_stat.o oce_tx.o \
892                  oce_utils.o
894 FCT_OBJS +=          discovery.o fct.o
896 QLT_OBJS +=          2400.o 2500.o 8100.o qlt.o qlt_dma.o
898 SRPT_OBJS +=         srpt_mod.o srpt_ch.o srpt_cm.o srpt_ioc.o srpt_stp.o
900 FCOE_OBJS +=         fcoe.o fcoe_eth.o fcoe_fc.o
902 FCOET_OBJS +=        fcoet.o fcoet_eth.o fcoet_fc.o
904 FCOEI_OBJS +=        fcoei.o fcoei_eth.o fcoei_lv.o
906 ISCSIT_SHARED_OBJS += \
907                  iscsit_common.o
909 ISCSIT_OBJS +=       $(ISCSIT_SHARED_OBJS) \
910                  iscsit.o iscsit_tgt.o iscsit_sess.o iscsit_login.o \
911                  iscsit_text.o iscsit_isns.o iscsit_radiusauth.o \
912                  iscsit_radiuspacket.o iscsit_auth.o iscsit_authclient.o
914 PPPT_OBJS +=        alua_ic_if.o pppt.o pppt_msg.o pppt_tgt.o
916 STMF_OBJS +=        lun_map.o stmf.o
918 STMF_SBD_OBJS +=    sbd.o sbd_scsi.o sbd_pgr.o sbd_zvol.o

```



```

920 SYSMMSG_OBJS += sysmsg.o
922 SES_OBJS += ses.o ses_sen.o ses_safte.o ses_ses.o
924 TNF_OBJS += tnf_buf.o tnf_trace.o tnf_writer.o trace_init.o \
925 trace_funcs.o tnf_probe.o tnf.o
927 LOGINDMUX_OBJS += loginmux.o
929 DEVINFO_OBJS += devinfo.o
931 DEVPOLL_OBJS += devpoll.o
933 DEVPOOL_OBJS += devpool.o
935 EVENTFD_OBJS += eventfd.o
937 SIGNALFD_OBJS += signalfd.o
939 I8042_OBJS += i8042.o
941 KB8042_OBJS += \
942 at_keyprocess.o \
943 kb8042.o \
944 kb8042_keytables.o
946 MOUSE8042_OBJS += mouse8042.o
948 FDC_OBJS += fdc.o
950 ASY_OBJS += asy.o
952 ECPP_OBJS += ecpp.o
954 VUIDM3P_OBJS += vuidmice.o vuidm3p.o
956 VUIDM4P_OBJS += vuidmice.o vuidm4p.o
958 VUIDM5P_OBJS += vuidmice.o vuidm5p.o
960 VUIDPS2_OBJS += vuidmice.o vuidps2.o
962 HPCSV_C_OBJS += hpcsvc.o
964 PCIE_MISC_OBJS += pcie.o pcie_fault.o pcie_hp.o pciehpc.o pcishpc.o pcie_pwr.o p
966 PCIHPNEXUS_OBJS += pcihp.o
968 OPENEPR_OBJS += openprom.o
970 RANDOM_OBJS += random.o
972 PSHOT_OBJS += pshot.o
974 GEN_DRV_OBJS += gen_drv.o
976 TCLIENT_OBJS += tclient.o
978 TIMERFD_OBJS += timerfd.o
980 TPHCI_OBJS += tphci.o
982 TVHCI_OBJS += tvhci.o
984 EMUL64_OBJS += emul64.o emul64_bsd.o

```

```

986 FCP_OBJS += fcp.o
988 FCIP_OBJS += fcip.o
990 FCSM_OBJS += fcsm.o
992 FCTL_OBJS += fctl.o
994 FP_OBJS += fp.o
996 QLC_OBJS += ql_api.o ql_debug.o ql_hba_fru.o ql_init.o ql_iocb.o ql_ioctl.o \
997 ql_isr.o ql_mbx.o ql_nx.o ql_xioctl.o ql_fw_table.o
999 QLC_FW_2200_OBJS += ql_fw_2200.o
1001 QLC_FW_2300_OBJS += ql_fw_2300.o
1003 QLC_FW_2400_OBJS += ql_fw_2400.o
1005 QLC_FW_2500_OBJS += ql_fw_2500.o
1007 QLC_FW_6322_OBJS += ql_fw_6322.o
1009 QLC_FW_8100_OBJS += ql_fw_8100.o
1011 QLGE_OBJS += qlge.o qlge_dbg.o qlge_flash.o qlge_fm.o qlge_gld.o qlge_mpi.o
1013 ZCONS_OBJS += zcons.o
1015 NV_SATA_OBJS += nv_sata.o
1017 SI3124_OBJS += si3124.o
1019 AHCI_OBJS += ahci.o
1021 PCIIDE_OBJS += pci-ide.o
1023 PCEPP_OBJS += pcepp.o
1025 CPC_OBJS += cpc.o
1027 CPUID_OBJS += cpuid_drv.o
1029 SYSEVENT_OBJS += sysevent.o
1031 BL_OBJS += bl.o
1033 DRM_OBJS += drm_sunmod.o drm_kstat.o drm_agpsupport.o \
1034 drm_auth.o drm_bufs.o drm_context.o drm_dma.o \
1035 drm_drawable.o drm_drv.o drm_fops.o drm_ioctl.o drm_irq.o \
1036 drm_lock.o drm_memory.o drm_msg.o drm_pci.o drm_scatter.o \
1037 drm_cache.o drm_gem.o drm_mm.o ati_pcigart.o
1039 FM_OBJS += devfm.o devfm_machdep.o
1041 RTLS_OBJS += rtls.o
1043 #
1044 # exec modules
1045 #
1046 AOUTEXEC_OBJS += aout.o
1048 ELFEXEC_OBJS += elf.o elf_notes.o old_notes.o
1050 INTPEXEC_OBJS += intp.o

```

```

1052 SHBINEXEC_OBJS +=shbin.o
1054 JAVAEXEC_OBJS +=java.o

1056 #
1057 #           file system modules
1058 #
1059 AUTOFSS_OBJS += auto_vfsops.o auto_vnops.o auto_subr.o auto_xdr.o auto_sys.o

1061 DCFS_OBJS += dc_vnops.o

1063 DEVFSS_OBJS += devfs_subr.o devfs_vfsops.o devfs_vnops.o

1065 DEV_OBJS += sdev_subr.o sdev_vfsops.o sdev_vnops.o \
1066 sdev_ptsops.o sdev_zvolops.o sdev_comm.o \
1067 sdev_profile.o sdev_ncache.o sdev_netops.o \
1068 sdev_ipnetops.o \
1069 sdev_vtops.o

1071 CTFS_OBJS += ctfs_all.o ctfs_cdir.o ctfs_ctl.o ctfs_event.o \
1072 ctfs_latest.o ctfs_root.o ctfs_sym.o ctfs_tdir.o ctfs_tmpl.o

1074 OBJFSS_OBJS += objfs_vfs.o objfs_root.o objfs_common.o \
1075 objfs_odir.o objfs_data.o

1077 FDFS_OBJS += fdops.o

1079 FIFO_OBJS += fifosubr.o fifovnops.o

1081 PIPE_OBJS += pipe.o

1083 HSFS_OBJS += hsfs_node.o hsfs_subr.o hsfs_vfsops.o hsfs_vnops.o \
1084 hsfs_susp.o hsfs_rrip.o hsfs_susp_subr.o

1086 LOFS_OBJS += lofs_subr.o lofs_vfsops.o lofs_vnops.o

1088 NAMEFSS_OBJS += namevfs.o namevno.o

1090 NFS_OBJS += nfs_client.o nfs_common.o nfs_dump.o \
1091 nfs_subr.o nfs_vfsops.o nfs_vnops.o \
1092 nfs_xdr.o nfs_sys.o nfs_strerror.o \
1093 nfs3_vfsops.o nfs3_vnops.o nfs3_xdr.o \
1094 nfs_acl_vnops.o nfs_acl_xdr.o nfs4_vfsops.o \
1095 nfs4_vnops.o nfs4_xdr.o nfs4_idmap.o \
1096 nfs4_shadow.o nfs4_subr.o \
1097 nfs4_attr.o nfs4_rnode.o nfs4_client.o \
1098 nfs4_acache.o nfs4_common.o nfs4_client_state.o \
1099 nfs4_callback.o nfs4_recovery.o nfs4_client_secinfo.o \
1100 nfs4_client_debug.o nfs_stats.o \
1101 nfs4_acl.o nfs4_stub_vnops.o nfs_cmd.o

1103 NFSSRV_OBJS += nfs_server.o nfs_srv.o nfs3_srv.o \
1104 nfs_acl_srv.o nfs_auth.o nfs_auth_xdr.o \
1105 nfs_export.o nfs_log.o nfs_log_xdr.o \
1106 nfs4_srv.o nfs4_state.o nfs4_srv_attr.o \
1107 nfs4_srv_ns.o nfs4_db.o nfs4_srv_deleg.o \
1108 nfs4_deleg_ops.o nfs4_srv_readdir.o nfs4_dispatch.o

1110 SMBSRV_SHARED_OBJS += \
1111 smb_door_legacy.o \
1112 smb_inet.o \
1113 smb_match.o \
1114 smb_msgbuf.o \
1115 smb_native.o \
1116 smb_netbios_util.o \

```

```

1117 smb_oem.o \
1118 smb_sid.o \
1119 smb_status2winerr.o \
1120 smb_string.o \
1121 smb_token.o \
1122 smb_token_xdr.o \
1123 smb_utf8.o \
1124 smb_xdr.o

1126 # See also: $SRC/lib/smb/srv/libfksmb/srv/Makefile.com
1127 SMBSRV_OBJS += $(SMBSRV_SHARED_OBJS) \
1128 smb_acl.o \
1129 smb_alloc.o \
1130 smb_authenticate.o \
1131 smb_close.o \
1132 smb_cmh_rename.o \
1133 smb_cmh_setfile.o \
1134 smb_common_open.o \
1135 smb_common_transact.o \
1136 smb_create.o \
1137 smb_cred.o \
1138 smb_delete.o \
1139 smb_dfs.o \
1140 smb_directory.o \
1141 smb_dispatch.o \
1142 smb_echo.o \
1143 smb_errno.o \
1144 smb_fem.o \
1145 smb_find.o \
1146 smb_flush.o \
1147 smb_fsinfo.o \
1148 smb_fsops.o \
1149 smb_idmap.o \
1150 smb_init.o \
1151 smb_kdoor.o \
1152 smb_kshare.o \
1153 smb_kutil.o \
1154 smb_lock.o \
1155 smb_lock_byte_range.o \
1156 smb_locking_andx.o \
1157 smb_logoff_andx.o \
1158 smb_mangle_name.o \
1159 smb_mbuf_marshall.o \
1160 smb_mbuf_util.o \
1161 smb_negotiate.o \
1162 smb_net.o \
1163 smb_node.o \
1164 smb_notify.o \
1165 smb_nt_cancel.o \
1166 smb_nt_create_andx.o \
1167 smb_nt_transact_create.o \
1168 smb_nt_transact_ioctl.o \
1169 smb_nt_transact_notify_change.o \
1170 smb_nt_transact_quota.o \
1171 smb_nt_transact_security.o \
1172 smb_odir.o \
1173 smb_ofile.o \
1174 smb_open_andx.o \
1175 smb_opipe.o \
1176 smb_oplock.o \
1177 smb_pathname.o \
1178 smb_print.o \
1179 smb_process_exit.o \
1180 smb_query_fileinfo.o \
1181 smb_quota.o \
1182 smb_read.o \

```

new/usr/src/uts/common/Makefile.files

19

```

1183 smb_rename.o \
1184 smb_sd.o \
1185 smb_seek.o \
1186 smb_server.o \
1187 smb_session.o \
1188 smb_session_setup_andx.o \
1189 smb_set_fileinfo.o \
1190 smb_sign_kcf.o \
1191 smb_signing.o \
1192 smb_thread.o \
1193 smb_tree.o \
1194 smb_trans2_create_directory.o \
1195 smb_trans2_dfs.o \
1196 smb_trans2_find.o \
1197 smb_tree_connect.o \
1198 smb_unlock_byte_range.o \
1199 smb_user.o \
1200 smb_vfs.o \
1201 smb_vops.o \
1202 smb_vss.o \
1203 smb_write.o \
1204 \
1205 smb2_dispatch.o \
1206 smb2_cancel.o \
1207 smb2_change_notify.o \
1208 smb2_close.o \
1209 smb2_create.o \
1210 smb2_echo.o \
1211 smb2_flush.o \
1212 smb2_ioctl.o \
1213 smb2_lock.o \
1214 smb2_logoff.o \
1215 smb2_negotiate.o \
1216 smb2_ofile.o \
1217 smb2_oplock.o \
1218 smb2_qinfo_file.o \
1219 smb2_qinfo_fs.o \
1220 smb2_qinfo_sec.o \
1221 smb2_qinfo_quota.o \
1222 smb2_query_dir.o \
1223 smb2_query_info.o \
1224 smb2_read.o \
1225 smb2_session_setup.o \
1226 smb2_set_info.o \
1227 smb2_setinfo_file.o \
1228 smb2_setinfo_fs.o \
1229 smb2_setinfo_quota.o \
1230 smb2_setinfo_sec.o \
1231 smb2_signing.o \
1232 smb2_tree_connect.o \
1233 smb2_tree_disconn.o \
1234 smb2_write.o \
1236 PCFS_OBJJS += pc_alloc.o pc_dir.o pc_node.o pc_subr.o \
1237 pc_vfsops.o pc_vnops.o \
1239 PROC_OBJJS += prcontrol.o prioctl.o prsubr.o prusr.o \
1240 prvfsops.o prvnops.o \
1242 MNTFS_OBJJS += mntvfsops.o mntvnops.o \
1244 SHAREFS_OBJJS += sharetab.o sharefs_vfsops.o sharefs_vnops.o \
1246 SPEC_OBJJS += specsabr.o specvfsops.o specvnops.o \
1248 SOCK_OBJJS += socksubr.o sockvfsops.o sockparams.o \

```

new/usr/src/uts/common/Makefile.files

20

```

1249 socksyscalls.o socktpi.o sockstr.o \
1250 sockcommon_vnops.o sockcommon_subr.o \
1251 sockcommon_sops.o sockcommon.o \
1252 sock_notsupp.o socknotify.o \
1253 nl7c.o nl7curi.o nl7chttp.o nl7clogd.o \
1254 nl7cnca.o sodirect.o sockfilter.o \
1256 TMPFS_OBJJS += tmp_dir.o tmp_subr.o tmp_tnode.o tmp_vfsops.o \
1257 tmp_vnops.o \
1259 UDFS_OBJJS += udf_alloc.o udf_bmap.o udf_dir.o \
1260 udf_inode.o udf_subr.o udf_vfsops.o \
1261 udf_vnops.o \
1263 UFS_OBJJS += ufs_alloc.o ufs_bmap.o ufs_dir.o ufs_xattr.o \
1264 ufs_inode.o ufs_subr.o ufs_tables.o ufs_vfsops.o \
1265 ufs_vnops.o quota.o quotacalls.o quota_ufs.o \
1266 ufs_filio.o ufs_lockfs.o ufs_thread.o ufs_trans.o \
1267 ufs_acl.o ufs_panic.o ufs_directio.o ufs_log.o \
1268 ufs_extvnops.o ufs_snap.o lufs.o lufs_thread.o \
1269 lufs_log.o lufs_map.o lufs_top.o lufs_debug.o \
1270 VSCAN_OBJJS += vscan_drv.o vscan_svc.o vscan_door.o \
1272 NSMB_OBJJS += smb_conn.o smb_dev.o smb_iod.o smb_pass.o \
1273 smb_rq.o smb_sign.o smb_smb.o smb_subrs.o \
1274 smb_time.o smb_tran.o smb_trantcp.o smb_usr.o \
1275 subr_mchain.o \
1277 SMBFS_COMMON_OBJJS += smbfs_ntacl.o \
1278 SMBFS_OBJJS += smbfs_vfsops.o smbfs_vnops.o smbfs_node.o \
1279 smbfs_acl.o smbfs_client.o smbfs_smb.o \
1280 smbfs_subr.o smbfs_subr2.o \
1281 smbfs_rwlock.o smbfs_xattr.o \
1282 $(SMBFS_COMMON_OBJJS) \
1284 BOOTFS_OBJJS += bootfs_construct.o bootfs_vfsops.o bootfs_vnops.o \
1286 LUA_OBJJS += \
1287 ldo.o \
1288 lvm.o \
1289 lbitlib.o \
1290 lopcodes.o \
1291 lstring.o \
1292 ltable.o \
1293 ltm.o \
1294 lcorolib.o \
1295 lauxlib.o \
1296 ldebug.o \
1297 lstate.o \
1298 lgc.o \
1299 lmem.o \
1300 lctype.o \
1301 lfunc.o \
1302 ldump.o \
1303 lundump.o \
1304 lstrlib.o \
1305 ltablib.o \
1306 lapi.o \
1307 lobject.o \
1308 lbaselib.o \
1309 lcompat.o \
1310 lzio.o \
1311 lcode.o \
1312 lllex.o \
1313 lparser.o \

```

```

1315 ZFS_COMMON_OBJS += \
1316     abd.o \
1317     aggsum.o \
1318     arc.o \
1319     blkptr.o \
1320     bplist.o \
1321     bpobj.o \
1322     bptree.o \
1323     bqueue.o \
1324     cityhash.o \
1325     dbuf.o \
1326     ddt.o \
1327     ddt_zap.o \
1328     dmuf.o \
1329     dmuf_diff.o \
1330     dmuf_send.o \
1331     dmuf_object.o \
1332     dmuf_objset.o \
1333     dmuf_traverse.o \
1334     dmuf_tx.o \
1335     dnode.o \
1336     dnode_sync.o \
1337     dsl_bookmark.o \
1338     dsl_dir.o \
1339     dsl_dataset.o \
1340     dsl_deadlist.o \
1341     dsl_destroy.o \
1342     dsl_pool.o \
1343     dsl_synctask.o \
1344     dsl_userhold.o \
1345     dmuf_zfetch.o \
1346     dsl_deleg.o \
1347     dsl_prop.o \
1348     dsl_scan.o \
1349     zfeature.o \
1350     gzip.o \
1351     lz4.o \
1352     lzjb.o \
1353     metaslab.o \
1354     multilist.o \
1355     range_tree.o \
1356     refcount.o \
1357     rrwlock.o \
1358     sa.o \
1359     sha256.o \
1360     edonr_zfs.o \
1361     skein_zfs.o \
1362     spa.o \
1363     spa_checkpoint.o \
1364     spa_config.o \
1365     spa_errlog.o \
1366     spa_history.o \
1367     spa_misc.o \
1368     space_map.o \
1369     space_reftree.o \
1370     txg.o \
1371     uberblock.o \
1372     unique.o \
1373     vdev.o \
1374     vdev_cache.o \
1375     vdev_file.o \
1376     vdev_indirect.o \
1377     vdev_indirect_births.o \
1378     vdev_indirect_mapping.o \
1379     vdev_initialize.o \
1380     vdev_label.o \

```

```

1381     vdev_mirror.o \
1382     vdev_missing.o \
1383     vdev_queue.o \
1384     vdev_raidz.o \
1385     vdev_removal.o \
1386     vdev_root.o \
1387     zap.o \
1388     zap_leaf.o \
1389     zap_micro.o \
1390     zcp.o \
1391     zcp_get.o \
1392     zcp_global.o \
1393     zcp_iter.o \
1394     zcp_synctask.o \
1395     zfs_byteswap.o \
1396     zfs_debug.o \
1397     zfs_fm.o \
1398     zfs_fuid.o \
1399     zfs_sa.o \
1400     zfs_znode.o \
1401     zil.o \
1402     zio.o \
1403     zio_checksum.o \
1404     zio_compress.o \
1405     zio_inject.o \
1406     zle.o \
1407     zlock.o \
1408     zthr.o \
1410 ZFS_SHARED_OBJS += \
1411     zfeature_common.o \
1412     zfs_comutil.o \
1413     zfs_deleg.o \
1414     zfs_fletcher.o \
1415     zfs_namecheck.o \
1416     zfs_prop.o \
1417     zpool_prop.o \
1418     zprop_common.o \
1420 ZFS_OBJS += \
1421     $(ZFS_COMMON_OBJS) \
1422     $(ZFS_SHARED_OBJS) \
1423     vdev_disk.o \
1424     zfs_acl.o \
1425     zfs_ctldir.o \
1426     zfs_dir.o \
1427     zfs_ioctl.o \
1428     zfs_log.o \
1429     zfs_onexit.o \
1430     zfs_replay.o \
1431     zfs_rlock.o \
1432     zfs_vfsops.o \
1433     zfs_vnops.o \
1434     zvol.o \
1436 ZUT_OBJS += \
1437     zut.o \
1439 # \
1440 #     streams modules \
1441 # \
1442 BUFMOD_OBJS += bufmod.o \
1444 CONNLD_OBJS += connld.o \
1446 DEDUMP_OBJS += dedump.o \

```

```

1448 DRCOMPAT_OBJS +=      drcompat.o
1450 LDLINUX_OBJS +=      ldlinux.o
1452 LDTERM_OBJS +=      ldterm.o uwidth.o
1454 PKCT_OBJS +=        pckt.o
1456 PFMOD_OBJS +=       pfmod.o
1458 PTEM_OBJS +=        ptem.o
1460 REDIRMOD_OBJS +=    strredirm.o
1462 TIMOD_OBJS +=       timod.o
1464 TIRDWR_OBJS +=      tirdwr.o
1466 TTCOMPAT_OBJS +=    ttcompat.o
1468 LOG_OBJS +=         log.o
1470 PIPEMOD_OBJS +=    pipemod.o
1472 RPCMOD_OBJS +=      rpcmod.o      clnt_cots.o      clnt_clts.o \
1473                          clnt_gen.o      clnt_perr.o      mt_rpcinit.o      rpc_calmsg.o \
1474                          rpc_prot.o      rpc_sztypes.o   rpc_subr.o      rpcb_prot.o \
1475                          svc.o          svc_clts.o      svc_gen.o      svc_cots.o \
1476                          rpcsys.o      xdr_sizeof.o   clnt_rdma.o     svc_rdma.o \
1477                          xdr_rdma.o      rdma_subr.o     xdrdma_sizeof.o
1479 KLMMOD_OBJS +=      klmmod.o \
1480                          nlm_impl.o \
1481                          nlm_rpc_handle.o \
1482                          nlm_dispatch.o \
1483                          nlm_rpc_svc.o \
1484                          nlm_client.o \
1485                          nlm_service.o \
1486                          nlm_prot_clnt.o \
1487                          nlm_prot_xdr.o \
1488                          nlm_rpc_clnt.o \
1489                          nsm_addr_clnt.o \
1490                          nsm_addr_xdr.o \
1491                          sm_inter_clnt.o \
1492                          sm_inter_xdr.o
1494 KLMOPS_OBJS +=      klmops.o
1496 TLIMOD_OBJS +=      tlimod.o      t_kalloc.o      t_kbind.o      t_kclose.o \
1497                          t_kconnect.o   t_kfree.o      t_kgtstate.o   t_kopen.o \
1498                          t_koptmgmt.o   t_krcvudat.o  t_ksndudat.o  t_kspoll.o \
1499                          t_kunbind.o    t_kutil.o
1501 RLMOD_OBJS +=      rlmmod.o
1503 TELMOD_OBJS +=      telmod.o
1505 CRYPTMOD_OBJS +=    cryptmod.o
1507 KB_OBJS +=          kbd.o          keytables.o
1509 #
1510 #                      ID mapping module
1511 #
1512 IDMAP_OBJS +=        idmap_mod.o     idmap_kapi.o   idmap_xdr.o    idmap_cache.o

```

```

1514 #
1515 #                      scheduling class modules
1516 #
1517 SDC_OBJS +=          sysdc.o
1519 RT_OBJS +=           rt.o
1520 RT_DPTBL_OBJS +=     rt_dptbl.o
1522 TS_OBJS +=           ts.o
1523 TS_DPTBL_OBJS +=     ts_dptbl.o
1525 IA_OBJS +=           ia.o
1527 FSS_OBJS +=         fss.o
1529 FX_OBJS +=           fx.o
1530 FX_DPTBL_OBJS +=     fx_dptbl.o
1532 #
1533 #                      Inter-Process Communication (IPC) modules
1534 #
1535 IPC_OBJS +=          ipc.o
1537 IPCMSG_OBJS +=      msg.o
1539 IPCSEM_OBJS +=      sem.o
1541 IPCSHM_OBJS +=      shm.o
1543 #
1544 #                      bignum module
1545 #
1546 COMMON_BIGNUM_OBJS += bignum_mod.o bignumimpl.o
1548 BIGNUM_OBJS +=      $(COMMON_BIGNUM_OBJS) $(BIGNUM_PSR_OBJS)
1550 #
1551 #                      kernel cryptographic framework
1552 #
1553 KCF_OBJS +=          kcf.o kcf_callprov.o kcf_cbufoall.o kcf_cipher.o kcf_crypto.o \
1554                          kcf_cryptoadm.o kcf_ctxops.o kcf_digest.o kcf_dual.o \
1555                          kcf_keys.o kcf_mac.o kcf_mech_tabs.o kcf_miscapi.o \
1556                          kcf_object.o kcf_policy.o kcf_prov_lib.o kcf_prov_tabs.o \
1557                          kcf_sched.o kcf_session.o kcf_sign.o kcf_spi.o kcf_verify.o \
1558                          kcf_random.o modes.o ecb.o cbc.o ctr.o ccm.o gcm.o \
1559                          fips_random.o
1561 CRYPTOADM_OBJS +=    cryptoadm.o
1563 CRYPTO_OBJS +=       crypto.o
1565 DPROV_OBJS +=        dprov.o
1567 DCA_OBJS +=          dca.o dca_3des.o dca_debug.o dca_dsa.o dca_kstat.o dca_rng.o \
1568                          dca_rsa.o
1570 AESPROV_OBJS +=      aes.o aes_impl.o aes_modes.o
1572 ARCFOURPROV_OBJS +=  arcfour.o arcfour_crypt.o
1574 BLOWFISHPROV_OBJS += blowfish.o blowfish_impl.o
1576 ECCPROV_OBJS +=      ecc.o ec.o ec2_163.o ec2_mont.o ecdecode.o ecl_mult.o \
1577                          ecp_384.o ecp_jac.o ec2_193.o ecl.o ecp_192.o ecp_521.o \
1578                          ecp_jm.o ec2_233.o ecl_curve.o ecp_224.o ecp_aff.o \

```

new/usr/src/uts/common/Makefile.files

25

```

1579         ecp_mont.o ec2_aff.o ec_naf.o ecl_gf.o ecp_256.o mp_gf2m.o \
1580         mpi.o mplogic.o mpmontg.o mpprime.o oid.o \
1581         secitem.o ec2_test.o ecp_test.o

1583 RSAPROV_OBJS += rsa.o rsa_impl.o pkcs1.o

1585 SWRANDPROV_OBJS += swrand.o

1587 #
1588 #             kernel SSL
1589 #
1590 KSSL_OBJS +=      kssl.o ksslioctl.o

1592 KSSL_SOCKFIL_MOD_OBJS += ksslfilter.o ksslapi.o ksslrec.o

1594 #
1595 #             misc. modules
1596 #

1598 C2AUDIT_OBJS +=  adr.o audit.o audit_event.o audit_io.o \
1599                 audit_path.o audit_start.o audit_syscalls.o audit_token.o \
1600                 audit_mem.o

1602 PCIC_OBJS +=    pcic.o

1604 RPCSEC_OBJS +=  secmod.o      sec_clnt.o      sec_svc.o      sec_gen.o \
1605                 auth_des.o    auth_kern.o    auth_none.o    auth_loopb.o \
1606                 authdesprt.o  authdesubr.o  auth_prot.o \
1607                 key_call.o    key_prot.o    svc_authu.o    svcauthdes.o

1609 RPCSEC_GSS_OBJS +=      rpcsec_gssmod.o rpcsec_gss.o rpcsec_gss_misc.o \
1610                         rpcsec_gss_utils.o svc_rpcsec_gss.o

1612 CONSCONFIG_OBJS += consconfig.o

1614 CONSCONFIG_DACF_OBJS += consconfig_dacf.o consplat.o

1616 TEM_OBJS += tem.o tem_safe.o

1618 #
1619 #       Font data for generated 8x16 font
1620 #
1621 VGATEXT_FONT      = 8x16
1622 VGATEXT_FONT_SRC= 8859-1
1623 VGATEXT_FONT_DIR= $(UTSBASE)/common/font
1624 FONT_OBJS += font.o 6x10.o 7x14.o 8x16.o 12x22.o

1626 KBTRANS_OBJS +=      \
1627                 kbtrans.o          \
1628                 kbtrans_keytables.o \
1629                 kbtrans_polled.o    \
1630                 kbtrans_streams.o   \
1631                 usb_keytables.o

1633 KGSSD_OBJS +=      gssd_clnt_stubs.o gssd_handle.o gssd_prot.o \
1634                 gss_display_name.o gss_release_name.o gss_import_name.o \
1635                 gss_release_buffer.o gss_release_oid_set.o gen_oids.o gssdmod.o

1637 KGSSD_DERIVED_OBJS = gssd_xdr.o

1639 KGSS_DUMMY_OBJS += dmech.o

1641 KSOCKET_OBJS += ksocket.o ksocket_mod.o

1643 CRYPTO= cksumtypes.o decrypt.o encrypt.o encrypt_length.o etypes.o \
1644         nfold.o verify_checksum.o prng.o block_size.o make_checksum.o \

```

new/usr/src/uts/common/Makefile.files

26

```

1645         checksum_length.o hmac.o default_state.o mandatory_sumtype.o

1647 # crypto/des
1648 CRYPTO_DES= f_cbc.o f_cksum.o f_parity.o weak_key.o d3_cbc.o ef_crypto.o

1650 CRYPTO_DK= checksum.o derive.o dk_decrypt.o dk_encrypt.o

1652 CRYPTO_ARCFOUR= k5_arcfour.o

1654 # crypto/enc_provider
1655 CRYPTO_ENC= des.o des3.o arcfour_provider.o aes_provider.o

1657 # crypto/hash_provider
1658 CRYPTO_HASH= hash_kef_generic.o hash_kmd5.o hash_crc32.o hash_kshal.o

1660 # crypto/keyhash_provider
1661 CRYPTO_KEYHASH= descbc.o k5_kmd5des.o k_hmac_md5.o

1663 # crypto/crc32
1664 CRYPTO_CRC32= crc32.o

1666 # crypto/old
1667 CRYPTO_OLD= old_decrypt.o old_encrypt.o

1669 # crypto/raw
1670 CRYPTO_RAW= raw_decrypt.o raw_encrypt.o

1672 K5_KRB= kfree.o copy_key.o \
1673         parse.o init_ctx.o \
1674         ser_adata.o ser_addr.o \
1675         ser_auth.o ser_cksum.o \
1676         ser_key.o ser_princ.o \
1677         serialize.o unparse.o \
1678         ser_actx.o

1680 K5_OS= timeofday.o toffset.o \
1681         init_os_ctx.o c_ustime.o

1683 SEAL= seal.o unseal.o

1685 MECH= delete_sec_context.o \
1686         import_sec_context.o \
1687         gssapi_krb5.o \
1688         k5seal.o k5unseal.o k5sealv3.o \
1689         ser_sctx.o \
1690         sign.o \
1691         util_crypt.o \
1692         util_validate.o util_ordering.o \
1693         util_seqnum.o util_set.o util_seed.o \
1694         wrap_size_limit.o verify.o

1698 MECH_GEN= util_token.o

1701 KGSS_KRB5_OBJS += krb5mech.o \
1702                 $(MECH) $(SEAL) $(MECH_GEN) \
1703                 $(CRYPTO) $(CRYPTO_DES) $(CRYPTO_DK) $(CRYPTO_ARCFOUR) \
1704                 $(CRYPTO_ENC) $(CRYPTO_HASH) \
1705                 $(CRYPTO_KEYHASH) $(CRYPTO_CRC32) \
1706                 $(CRYPTO_OLD) \
1707                 $(CRYPTO_RAW) $(K5_KRB) $(K5_OS)

1709 DES_OBJS +=      des_crypt.o des_impl.o des_ks.o des_soft.o

```

```

1711 DLBOOT_OBJS += bootparam_xdr.o nfs_dlnet.o scan.o
1713 KRTLD_OBJS += kobj_bootflags.o getoptstr.o \
1714 kobj.o kobj_kdi.o kobj_lm.o kobj_subr.o
1716 MOD_OBJS += modctl.o modsubr.o modsysfile.o modconf.o modhash.o
1718 STRPLUMB_OBJS += strplumb.o
1720 CPR_OBJS += cpr_driver.o cpr_dump.o \
1721 cpr_main.o cpr_misc.o cpr_mod.o cpr_stat.o \
1722 cpr_uthread.o
1724 PROF_OBJS += prf.o
1726 SE_OBJS += se_driver.o
1728 SYSACCT_OBJS += acct.o
1730 ACCTCTL_OBJS += acctctl.o
1732 EXACTSYS_OBJS += exactsys.o
1734 KAIO_OBJS += aio.o
1736 PCMCIA_OBJS += pcmcia.o cs.o cis.o cis_callout.o cis_handlers.o cis_params.o
1738 BUSRA_OBJS += busra.o
1740 PCS_OBJS += pcs.o
1742 PSET_OBJS += pset.o
1744 OHCI_OBJS += ohci.o ohci_hub.o ohci_polled.o
1746 UHCI_OBJS += uhci.o uhciutil.o uhci_tgt.o uhcihub.o uhci_polled.o
1748 EHCI_OBJS += ehci.o ehci_hub.o ehci_xfer.o ehci_intr.o ehci_util.o ehci_polled.o
1750 XHCI_OBJS += xhci.o xhci_quirks.o xhci_dma.o xhci_context.o xhci_intr.o \
1751 xhci_ring.o xhci_command.o xhci_event.o xhci_usba.o \
1752 xhci_endpoint.o xhci_hub.o
1754 HUBD_OBJS += hubd.o
1756 USB_MID_OBJS += usb_mid.o
1758 USB_IA_OBJS += usb_ia.o
1760 SCSA2USB_OBJS += scsa2usb.o usb_ms_bulkonly.o usb_ms_cbi.o
1762 IPF_OBJS += ip_fil_solaris.o fil.o solaris.o ip_state.o ip_frag.o ip_nat.o \
1763 ip_proxy.o ip_auth.o ip_pool.o ip_hstable.o ip_lookup.o \
1764 ip_log.o misc.o ip_compat.o ip_nat6.o drand48.o
1766 IPD_OBJS += ipd.o
1768 IBD_OBJS += ibd.o ibd_cm.o
1770 EIBNX_OBJS += enx_main.o enx_hdlrs.o enx_ibt.o enx_log.o enx_fip.o \
1771 enx_misc.o enx_q.o enx_ctl.o
1773 EOIB_OBJS += eib_adm.o eib_chan.o eib_cm.o eib_ctl.o eib_data.o \
1774 eib_fip.o eib_ibt.o eib_log.o eib_mac.o eib_main.o \
1775 eib_rsrc.o eib_svc.o eib_vnic.o

```

```

1777 DLPSTUB_OBJS += dlpistub.o
1779 SDP_OBJS += sdpddi.o
1781 TRILL_OBJS += trill.o
1783 CTF_OBJS += ctf_create.o ctf_decl.o ctf_error.o ctf_hash.o ctf_labels.o \
1784 ctf_lookup.o ctf_open.o ctf_types.o ctf_util.o ctf_subr.o ctf_mod.o
1786 SMBIOS_OBJS += smb_error.o smb_info.o smb_open.o smb_subr.o smb_dev.o
1788 RPCIB_OBJS += rpcib.o
1790 KMDB_OBJS += kdrv.o
1792 AFE_OBJS += afe.o
1794 BGE_OBJS += bge_main2.o bge_chip2.o bge_kstats.o bge_log.o bge_ndd.o \
1795 bge_atomic.o bge_mii.o bge_send.o bge_recv2.o bge_mii_5906.o
1797 DMFE_OBJS += dmfe_log.o dmfe_main.o dmfe_mii.o
1799 EFE_OBJS += efe.o
1801 ELXL_OBJS += elxl.o
1803 HME_OBJS += hme.o
1805 IXGB_OBJS += ixgb.o ixgb_atomic.o ixgb_chip.o ixgb_gld.o ixgb_kstats.o \
1806 ixgb_log.o ixgb_ndd.o ixgb_rx.o ixgb_tx.o ixgb_xmii.o
1808 NGE_OBJS += nge_main.o nge_atomic.o nge_chip.o nge_ndd.o nge_kstats.o \
1809 nge_log.o nge_rx.o nge_tx.o nge_xmii.o
1811 PCN_OBJS += pcn.o
1813 RGE_OBJS += rge_main.o rge_chip.o rge_ndd.o rge_kstats.o rge_log.o rge_rxtx.o
1815 URTW_OBJS += urtw.o
1817 ARN_OBJS += arn_hw.o arn_eeprom.o arn_mac.o arn_calib.o arn_ani.o arn_phy.o arn_
1818 arn_main.o arn_recv.o arn_xmit.o arn_rc.o
1820 ATH_OBJS += ath_aux.o ath_main.o ath_osdep.o ath_rate.o
1822 ATU_OBJS += atu.o
1824 IPW_OBJS += ipw2100_hw.o ipw2100.o
1826 IWI_OBJS += ipw2200_hw.o ipw2200.o
1828 IWH_OBJS += iwh.o
1830 IWK_OBJS += iw2.o
1832 IWN_OBJS += if_iwn.o
1834 IWP_OBJS += iwp.o
1836 MWL_OBJS += mw1.o
1838 MWLFW_OBJS += mw1fw_mode.o
1840 WPI_OBJS += wpi.o
1842 RAL_OBJS += rt2560.o ral_rate.o

```

```

1844 RUM_OBJS += rum.o
1846 RWD_OBJS += rt2661.o
1848 RWN_OBJS += rt2860.o
1850 UATH_OBJS += uath.o
1852 UATHFW_OBJS += uathfw_mod.o
1854 URAL_OBJS += ural.o
1856 RTW_OBJS += rtw.o smc93cx6.o rtwphy.o rtwphyio.o
1858 ZYD_OBJS += zyd.o zyd_usb.o zyd_hw.o zyd_fw.o
1860 MXFE_OBJS += mxfe.o
1862 MPTSAS_OBJS += mptsas.o mptsas_hash.o mptsas_impl.o mptsas_init.o \
1863 mptsas_raid.o mptsas_smhba.o
1865 SFE_OBJS += sfe.o sfe_util.o
1867 BFE_OBJS += bfe.o
1869 BRIDGE_OBJS += bridge.o
1871 IDM_SHARED_OBJS += base64.o
1873 IDM_OBJS += $(IDM_SHARED_OBJS) \
1874 idm.o idm_impl.o idm_text.o idm_conn_sm.o idm_so.o
1876 VR_OBJS += vr.o
1878 ATGE_OBJS += atge_main.o atge_lle.o atge_mii.o atge_ll.o atge_llc.o
1880 YGE_OBJS = yge.o
1882 SKD_OBJS = skd.o
1884 NVME_OBJS = nvme.o
1886 #
1887 # Build up defines and paths.
1888 #
1889 LINT_DEFS += -Dunix
1891 #
1892 # This duality can be removed when the native and target compilers
1893 # are the same (or at least recognize the same command line syntax!)
1894 # It is a bug in the current compilation system that the assembler
1895 # can't process the -Y I, flag.
1896 #
1897 NATIVE_INC_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1898 AS_INC_PATH += $(INC_PATH) -I$(UTSBASE)/common
1899 INCLUDE_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1901 PCIEB_OBJS += pcieb.o
1903 # Chelsio N110 10G NIC driver module
1904 #
1905 CH_OBJS = ch.o glue.o pe.o sge.o
1907 CH_COM_OBJS = ch_mac.o ch_subr.o csapi.o espi.o ixfl1010.o mc3.o mc4.o mc5.o \
1908 mv88elxxx.o mv88x20lx.o my3126.o pm3393.o tp.o ulp.o \

```

```

1909 vsc7321.o vsc7326.o xpak.o
1911 #
1912 # Chelsio Terminator 4 10G NIC nexus driver module
1913 #
1914 CXGBE_FW_OBJS = t4_fw.o t4_cfg.o t5_cfg.o t5_fw.o t6_cfg.o t6_fw.o
1915 CXGBE_COM_OBJS = t4_hw.o
1916 CXGBE_NEX_OBJS = t4_nexus.o t4_sge.o t4_mac.o t4_ioctl.o shared.o \
1917 t4_l2t.o osdep.o
1919 #
1920 # Chelsio Terminator 4 10G NIC driver module
1921 #
1922 CXGBE_OBJS = cxgbe.o
1924 #
1925 # PCI strings file
1926 #
1927 PCI_STRING_OBJS = pci_strings.o
1929 NET_DACF_OBJS += net_dacf.o
1931 #
1932 # Xframe 10G NIC driver module
1933 #
1934 XGE_OBJS = xge.o xgell.o
1936 XGE_HAL_OBJS = xgehal-channel.o xgehal-fifo.o xgehal-ring.o xgehal-config.o \
1937 xgehal-driver.o xgehal-mm.o xgehal-stats.o xgehal-device.o \
1938 xge-queue.o xgehal-mgmt.o xgehal-mgmtaux.o
1940 #
1941 # e1000/igb common objs
1942 #
1943 # Historically e1000g and igb had separate copies of all of the common
1944 # code. At this time while they are now sharing the same copy of it, they
1945 # are building it into their own modules which is due to the differences
1946 # in the osdep and debug portions of their code.
1947 #
1948 E1000API_OBJS += e1000_80003es2lan.o e1000_82540.o e1000_82541.o e1000_82542.o \
1949 e1000_82543.o e1000_82571.o e1000_api.o e1000_ich8lan.o \
1950 e1000_mac.o e1000_manage.o e1000_nvmm.o e1000_phy.o \
1951 e1000_82575.o e1000_i210.o e1000_mbx.o e1000_vf.o
1953 #
1954 # e1000g module
1955 #
1956 E1000G_OBJS += e1000g_debug.o e1000g_main.o e1000g_alloc.o \
1957 e1000g_tx.o e1000g_rx.o e1000g_stat.o \
1958 e1000g_osdep.o e1000g_workarounds.o
1961 #
1962 # Intel 82575 1G NIC driver module
1963 #
1964 IGB_OBJS = igb_buf.o igb_debug.o igb_gld.o igb_log.o igb_main.o \
1965 igb_rx.o igb_stat.o igb_tx.o igb_osdep.o
1967 #
1968 # Intel Pro/100 NIC driver module
1969 #
1970 IPRB_OBJS = iprb.o
1972 #
1973 # Intel 10GbE PCIE NIC driver module
1974 #

```



```

1976 # illumos-written source files
1977 IXGBE_OBJS = ixgbe_buf.o ixgbe_debug.o ixgbe_gld.o ixgbe_log.o ixgbe_main.o \
1978 ixgbe_osdep.o ixgbe_rx.o ixgbe_stat.o ixgbe_transceiver.o \
1979 ixgbe_tx.o

1981 # Intel-written source files
1982 IXGBE_INTC_OBJS = ixgbe_82598.o ixgbe_82599.o ixgbe_api.o ixgbe_common.o \
1983 ixgbe_phy.o ixgbe_dcb.o ixgbe_dcb_82598.o \
1984 ixgbe_dcb_82599.o ixgbe_mbx.o ixgbe_vf.o ixgbe_x540.o \
1985 ixgbe_x550.o

1987 #
1988 # Intel 40GbE PCIe NIC driver module
1989 #

1991 # illumos-written ones.
1992 I40E_OBJS = i40e_main.o i40e_osdep.o i40e_intr.o i40e_transceiver.o \
1993 i40e_stats.o i40e_gld.o
1994 # Intel-written ones.
1995 I40E_INTC_OBJS = i40e_adminq.o i40e_common.o i40e_hmc.o i40e_lan_hmc.o \
1996 i40e_nvmm.o

1998 #
1999 # Solarflare 1/10/40GbE NIC driver module
2000 #
2001 # NB: The illumos specific sources are listed first, with the
2002 # common (OS-independent) sources afterwards.
2003 #
2004 SFXGE_OBJS = sfxge_err.o sfxge_ev.o sfxge_hash.o sfxge_intr.o sfxge_mac.o \
2005 sfxge_gld_v3.o sfxge_mon.o sfxge_phy.o \
2006 sfxge_sram.o sfxge_bar.o sfxge_pci.o sfxge_nvram.o \
2007 sfxge_rx.o sfxge_tcp.o sfxge_tx.o sfxge_mcdi.o sfxge_vpd.o \
2008 sfxge.o sfxge_dma.o
2009 SFXGE_SF_OBJS = efx_bootcfg.o efx_crc32.o efx_ev.o efx_filter.o \
2010 efx_hash.o efx_intr.o efx_mac.o efx_mcdi.o efx_mon.o \
2011 efx_nic.o efx_nvram.o efx_phy.o efx_port.o efx_rx.o \
2012 efx_sram.o efx_tx.o efx_vpd.o efx_wol.o mcdi_mon.o \
2013 siena_mac.o siena_mcdi.o siena_nic.o siena_nvram.o \
2014 siena_phy.o siena_sram.o siena_vpd.o \
2015 ef10_ev.o ef10_filter.o ef10_intr.o ef10_mac.o ef10_mcdi.o \
2016 ef10_nic.o ef10_nvram.o ef10_phy.o ef10_rx.o ef10_tx.o \
2017 ef10_vpd.o hunt_nic.o hunt_phy.o

2019 #
2020 # NIU 10G/1G driver module
2021 #
2022 NXGE_OBJS = nxge_mac.o nxge_ipp.o nxge_rxdma.o \
2023 nxge_txdma.o nxge_txc.o nxge_main.o \
2024 nxge_hw.o nxge_fzc.o nxge_virtual.o \
2025 nxge_send.o nxge_classify.o nxge_fflp.o \
2026 nxge_fflp_hash.o nxge_ndd.o nxge_kstats.o \
2027 nxge_zcp.o nxge_fm.o nxge_espc.o nxge_hv.o \
2028 nxge_hio.o nxge_hio_guest.o nxge_intr.o

2030 NXGE_NPI_OBJS = \
2031 npi.o npi_mac.o npi_ipp.o \
2032 npi_txdma.o npi_rxdma.o npi_txc.o \
2033 npi_zcp.o npi_espc.o npi_fflp.o \
2034 npi_vir.o

2036 NXGE_HCALL_OBJS = \
2037 nxge_hcall.o

2039 #
2040 # Virtio modules

```

```

2041 #

2043 # Virtio core
2044 VIRTIO_OBJS = virtio.o

2046 # Virtio block driver
2047 VIOBLK_OBJS = vioblk.o

2049 # Virtio network driver
2050 VIOIF_OBJS = vioif.o

2052 #
2053 # kiconv modules
2054 #
2055 KICONV_EMEA_OBJS += kiconv_emea.o

2057 KICONV_JA_OBJS += kiconv_ja.o

2059 KICONV_KO_OBJS += kiconv_cck_common.o kiconv_ko.o

2061 KICONV_SC_OBJS += kiconv_cck_common.o kiconv_sc.o

2063 KICONV_TC_OBJS += kiconv_cck_common.o kiconv_tc.o

2065 #
2066 # AAC module
2067 #
2068 AAC_OBJS = aac.o aac_ioctl.o

2070 #
2071 # sdcards modules
2072 #
2073 SDA_OBJS = sda_cmd.o sda_host.o sda_init.o sda_mem.o sda_mod.o sda_slot.o
2074 SDHOST_OBJS = sdhost.o

2076 #
2077 # hxge 10G driver module
2078 #
2079 HXGE_OBJS = hxge_main.o hxge_vmac.o hxge_send.o \
2080 hxge_txdma.o hxge_rxdma.o hxge_virtual.o \
2081 hxge_fm.o hxge_fzc.o hxge_hw.o hxge_kstats.o \
2082 hxge_ndd.o hxge_pfc.o \
2083 hpi.o hpi_vmac.o hpi_rxdma.o hpi_txdma.o \
2084 hpi_vir.o hpi_pfc.o

2086 #
2087 # MEGARAID_SAS module
2088 #
2089 MEGA_SAS_OBJS = megaraid_sas.o

2091 #
2092 # MR_SAS module
2093 #
2094 MR_SAS_OBJS = ld_pd_map.o mr_sas.o mr_sas_tbolt.o mr_sas_list.o

2096 #
2097 # CPQARY3 module
2098 #
2099 CPQARY3_OBJS = cpqary3.o cpqary3_noe.o cpqary3_talk2ctrl.o \
2100 cpqary3_isr.o cpqary3_transport.o cpqary3_mem.o \
2101 cpqary3_scsi.o cpqary3_util.o cpqary3_ioctl.o \
2102 cpqary3_bd.o

2104 #
2105 # ISCSI_INITIATOR module
2106 #

```

new/usr/src/uts/common/Makefile.files

```

2107 ISCSI_INITIATOR_OBJS = chap.o iscsi_io.o iscsi_thread.o \
2108 iscsi_ioctl.o iscsid.o iscsi.o \
2109 iscsi_login.o isns_client.o iscsiAuthClient.o \
2110 iscsi_lun.o iscsiAuthClientGlue.o \
2111 iscsi_net.o nvfile.o iscsi_cmd.o \
2112 iscsi_queue.o persistent.o iscsi_conn.o \
2113 iscsi_sess.o radius_auth.o iscsi_crc.o \
2114 iscsi_stats.o radius_packet.o iscsi_doorclt.o \
2115 iscsi_targetparam.o utils.o kifconf.o

2117 #
2118 # ntxn 10Gb/1Gb NIC driver module
2119 #
2120 NTXN_OBJS = unm_nic_init.o unm_gem.o unm_nic_hw.o unm_ndd.o \
2121 unm_nic_main.o unm_nic_isr.o unm_nic_ctx.o niu.o

2123 #
2124 # Myricom 10Gb NIC driver module
2125 #
2126 MYRI10GE_OBJS = myri10ge.o myri10ge_lro.o

2128 # nulldriver module
2129 #
2130 NULLDRIVER_OBJS = nulldriver.o

2132 TPM_OBJS = tpm.o tpm_hcall.o

2134 #
2135 # USB Fast ethernet drivers
2136 #
2137 USBGEM_OBJS = usbgem.o
2138 AXF_OBJS = axf_usbgem.o
2139 UDMF_OBJS = udmf_usbgem.o
2140 URF_OBJS = urf_usbgem.o
2141 UPF_OBJS = upf_usbgem.o

2143 #
2144 # BNXE objects
2145 #
2146 BNXE_OBJS += bnxe_cfg.o \
2147 bnxe_fcoe.o \
2148 bnxe_debug.o \
2149 bnxe_gld.o \
2150 bnxe_hw.o \
2151 bnxe_intr.o \
2152 bnxe_kstat.o \
2153 bnxe_lock.o \
2154 bnxe_main.o \
2155 bnxe_mm.o \
2156 bnxe_mm_14.o \
2157 bnxe_mm_15.o \
2158 bnxe_rr.o \
2159 bnxe_rx.o \
2160 bnxe_timer.o \
2161 bnxe_tx.o \
2162 bnxe_workq.o \
2163 bnxe_clc.o \
2164 bnxe_illumos.o \
2165 ecore_sp_verbs.o \
2166 bnxe_context.o \
2167 57710_init_values.o \
2168 57711_init_values.o \
2169 57712_init_values.o \
2170 bnxe_fw_funcs.o \
2171 bnxe_hw_debug.o \
2172 lm_14fp.o

```

33

new/usr/src/uts/common/Makefile.files

```

2173 lm_14rx.o \
2174 lm_14sp.o \
2175 lm_14tx.o \
2176 lm_15.o \
2177 lm_15sp.o \
2178 lm_dcbx.o \
2179 lm_devinfo.o \
2180 lm_dmae.o \
2181 lm_er.o \
2182 lm_hw_access.o \
2183 lm_hw_attn.o \
2184 lm_hw_init_reset.o \
2185 lm_main.o \
2186 lm_mcp.o \
2187 lm_niv.o \
2188 lm_nvram.o \
2189 lm_phy.o \
2190 lm_power.o \
2191 lm_recv.o \
2192 lm_resc.o \
2193 lm_sb.o \
2194 lm_send.o \
2195 lm_sp.o \
2196 lm_dcbx_mp.o \
2197 lm_sp_req_mgr.o \
2198 lm_stats.o \
2199 lm_util.o

2201 #
2202 # The QEDE_EN_OBJS list are general objects that are shared across
2203 # multiple different operating systems. QEDE_OBJS are illumos-specific
2204 # objects.
2205 #
2206 QEDE_OBJS += qede_cfg.o qede_dbg.o qede_fp.o qede_gld.o \
2207 qede_kstat.o qede_main.o qede_misc.o qede_osal.o
2208 QEDE_GEN_OBJS += ecore_hw.o ecore_cxt.o ecore_selftest.o ecore_init_ops.o \
2209 ecore_init_fw_funcs.o ecore_sp_commands.o ecore_dcbx.o \
2210 ecore_dbg_fw_funcs.o ecore_mcp.o ecore_spq.o ecore_phy.o \
2211 ecore_dev.o ecore_l2.o ecore_int.o

```

34

```

*****
1664 Tue Nov 20 12:08:34 2018
new/usr/src/uts/common/os/cpu_uarray.c
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 /*
2  * This file and its contents are supplied under the terms of the
3  * Common Development and Distribution License ("CDDL"), version 1.0.
4  * You may only use this file in accordance with the terms of version
5  * 1.0 of the CDDL.
6  *
7  * A full copy of the text of the CDDL should have accompanied this
8  * source. A copy of the CDDL is also available via the Internet at
9  * http://www.illumos.org/license/CDDL.
10 */

12 /*
13  * Copyright (c) 2018, Joyent, Inc.
14  */

16 #include <sys/cpu_uarray.h>
17 #include <sys/sysmacros.h>
18 #include <sys/cpuvar.h>
19 #include <sys/debug.h>
20 #include <sys/kmem.h>

22 static size_t
23 cpu_uarray_size(size_t nr_items)
24 {
25     size_t size = P2ROUNDUP(nr_items * sizeof (uint64_t), CUA_ALIGN);
26     size *= NCPU;
27     return (sizeof (cpu_uarray_t) + size);
28 }

30 cpu_uarray_t *
31 cpu_uarray_zalloc(size_t nr_items, int kmflags)
32 {
33     cpu_uarray_t *cua;

35     cua = kmem_zalloc(cpu_uarray_size(nr_items), kmflags);

37     if (cua != NULL) {
38         VERIFY(IS_P2ALIGNED(cua->cu_vals, CUA_ALIGN));
39         cua->cu_nr_items = nr_items;
40     }

42     return (cua);
43 }

45 void
46 cpu_uarray_free(cpu_uarray_t *cua)
47 {
48     kmem_free(cua, cpu_uarray_size(cua->cu_nr_items));
49 }

51 uint64_t
52 cpu_uarray_sum(cpu_uarray_t *cua, size_t index)
53 {
54     uint64_t sum = 0;

56     VERIFY3U(index, <, cua->cu_nr_items);

58     for (size_t c = 0; c < ncpus; c++) {
59         uint64_t addend = CPU_UARRAY_VAL(cua, c, index);
60         sum = UINT64_OVERFLOW_ADD(sum, addend);

```

```

61     }
62     return (sum);
63 }

66 uint64_t
67 cpu_uarray_sum_all(cpu_uarray_t *cua)
68 {
69     uint64_t sum = 0;

71     for (size_t c = 0; c < ncpus; c++) {
72         for (size_t i = 0; i < cua->cu_nr_items; i++) {
73             uint64_t addend = CPU_UARRAY_VAL(cua, c, i);
74             sum = UINT64_OVERFLOW_ADD(sum, addend);
75         }
76     }

78     return (sum);
79 }

```

```

*****
22975 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/os/msacct.c
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright 2009 Sun Microsystems, Inc. All rights reserved.
23 * Use is subject to license terms.
24 * Copyright (c) 2018, Joyent, Inc.
24 * Copyright 2012 Joyent, Inc. All rights reserved.
25 */

27 #include <sys/types.h>
28 #include <sys/param.h>
29 #include <sys/system.h>
30 #include <sys/user.h>
31 #include <sys/proc.h>
32 #include <sys/cpuvar.h>
33 #include <sys/thread.h>
34 #include <sys/debug.h>
35 #include <sys/msacct.h>
36 #include <sys/time.h>
37 #include <sys/zone.h>

39 /*
40  * Mega-theory block comment:
41  *
42  * Microstate accounting uses finite states and the transitions between these
43  * states to measure timing and accounting information. The state information
44  * is presently tracked for threads (via microstate accounting) and cpus (via
45  * cpu microstate accounting). In each case, these accounting mechanisms use
46  * states and transitions to measure time spent in each state instead of
47  * clock-based sampling methodologies.
48  *
49  * For microstate accounting:
50  * state transitions are accomplished by calling new_mstate() to switch between
51  * states. Transitions from a sleeping state (LMS_SLEEP and LMS_STOPPED) occur
52  * by calling restore_mstate() which restores a thread to its previously running
53  * state. This code is primarily executed by the dispatcher in disp() before
54  * running a process that was put to sleep. If the thread was not in a sleeping
55  * state, this call has little effect other than to update the count of time the
56  * thread has spent waiting on run-queues in its lifetime.
57  *
58  * For cpu microstate accounting:
59  * Cpu microstate accounting is similar to the microstate accounting for threads

```

```

60 * but it tracks user, system, and idle time for cpus. Cpu microstate
61 * accounting does not track interrupt times as there is a pre-existing
62 * interrupt accounting mechanism for this purpose. Cpu microstate accounting
63 * tracks time that user threads have spent active, idle, or in the system on a
64 * given cpu. Cpu microstate accounting has fewer states which allows it to
65 * have better defined transitions. The states transition in the following
66 * order:
67 *
68 * CMS_USER <-> CMS_SYSTEM <-> CMS_IDLE
69 *
70 * In order to get to the idle state, the cpu microstate must first go through
71 * the system state, and vice-versa for the user state from idle. The switching
72 * of the microstates from user to system is done as part of the regular thread
73 * microstate accounting code, except for the idle state which is switched by
74 * the dispatcher before it runs the idle loop.
75 *
76 * Cpu percentages:
77 * Cpu percentages are now handled by and based upon microstate accounting
78 * information (the same is true for load averages). The routines which handle
79 * the growing/shrinking and exponentiation of cpu percentages have been moved
80 * here as it now makes more sense for them to be generated from the microstate
81 * code. Cpu percentages are generated similarly to the way they were before;
82 * however, now they are based upon high-resolution timestamps and the
83 * timestamps are modified at various state changes instead of during a clock()
84 * interrupt. This allows us to generate more accurate cpu percentages which
85 * are also in-sync with microstate data.
86 */

88 /*
89  * Initialize the microstate level and the
90  * associated accounting information for an LWP.
91  */
92 void
93 init_mstate(
94     kthread_t *t,
95     int init_state)
96 {
97     struct mstate *ms;
98     klpw_t *lwp;
99     hrtime_t curtime;

101     ASSERT(init_state != LMS_WAIT_CPU);
102     ASSERT((unsigned)init_state < NMSTATES);

104     if ((lwp = ttolwp(t)) != NULL) {
105         ms = &lwp->lwp_mstate;
106         curtime = gethrtime_unscaled();
107         ms->ms_prev = LMS_SYSTEM;
108         ms->ms_start = curtime;
109         ms->ms_term = 0;
110         ms->ms_state_start = curtime;
111         t->t_mstate = init_state;
112         t->t_waitrq = 0;
113         t->t_hrtime = curtime;
114         if ((t->t_proc_flag & TP_MSACCT) == 0)
115             t->t_proc_flag |= TP_MSACCT;
116         bzero((caddr_t)&ms->ms_acct[0], sizeof (ms->ms_acct));
117     }
118 }
unchanged_portion_omitted

391 void
392 syscall_mstate(int fromms, int toms)
393 {
394     kthread_t *t = curthread;

```

```

395     zone_t *z = ttozone(t);
396     struct mstate *ms;
397     hrtime_t *mstimep;
398     hrtime_t curtime;
399     klwp_t *lwp;
400     hrtime_t newtime;
401     cpu_t *cpu;
402     uint16_t gen;

404     if ((lwp = ttolwp(t)) == NULL)
405         return;

407     ASSERT(fromms < NMSTATES);
408     ASSERT(toms < NMSTATES);

410     ms = &lwp->lwp_mstate;
411     mstimep = &ms->ms_acct[fromms];
412     curtime = gethrtime_unscaled();
413     newtime = curtime - ms->ms_state_start;
414     while (newtime < 0) {
415         curtime = gethrtime_unscaled();
416         newtime = curtime - ms->ms_state_start;
417     }
418     *mstimep += newtime;
419     if (fromms == LMS_USER)
420         atomic_add_64(&z->zone_utime, newtime);
421     else if (fromms == LMS_SYSTEM)
422         atomic_add_64(&z->zone_stime, newtime);
419     t->t_mstate = toms;
420     ms->ms_state_start = curtime;
421     ms->ms_prev = fromms;
422     kpreempt_disable(); /* don't change CPU while changing CPU's state */
423     cpu = CPU;
424     ASSERT(cpu == t->t_cpu);

426     if (fromms == LMS_USER) {
427         CPU_UARRAY_VAL(z->zone_ustate, cpu->cpu_id,
428             ZONE_USTATE_UTIME) += newtime;
429     } else if (fromms == LMS_SYSTEM) {
430         CPU_UARRAY_VAL(z->zone_ustate, cpu->cpu_id,
431             ZONE_USTATE_STIME) += newtime;
432     }

434     if ((toms != LMS_USER) && (cpu->cpu_mstate != CMS_SYSTEM)) {
435         NEW_CPU_MSTATE(CMS_SYSTEM);
436     } else if ((toms == LMS_USER) && (cpu->cpu_mstate != CMS_USER)) {
437         NEW_CPU_MSTATE(CMS_USER);
438     }
439     kpreempt_enable();
440 }

```

unchanged portion omitted

```

594 /*
595  * Change the microstate level for the LWP and update the
596  * associated accounting information. Return the previous
597  * LWP state.
598  */
599 int
600 new_mstate(kthread_t *t, int new_state)
601 {
602     struct mstate *ms;
603     unsigned state;
604     hrtime_t *mstimep;
605     hrtime_t curtime;
606     hrtime_t newtime;
607     hrtime_t oldtime;

```

```

608     hrtime_t ztime;
609     hrtime_t origstart;
610     klwp_t *lwp;
611     zone_t *z;

613     ASSERT(new_state != LMS_WAIT_CPU);
614     ASSERT((unsigned)new_state < NMSTATES);
615     ASSERT(t == curthread || THREAD_LOCK_HELD(t));

617     /*
618     * Don't do microstate processing for threads without a lwp (kernel
619     * threads). Also, if we're an interrupt thread that is pinning another
620     * thread, our t_mstate hasn't been initialized. We'd be modifying the
621     * microstate of the underlying lwp which doesn't realize that it's
622     * pinned. In this case, also don't change the microstate.
623     */
624     if (((lwp = ttolwp(t)) == NULL) || t->t_intr)
625         return (LMS_SYSTEM);

627     curtime = gethrtime_unscaled();

629     /* adjust cpu percentages before we go any further */
630     (void) cpu_update_pct(t, curtime);

632     ms = &lwp->lwp_mstate;
633     state = t->t_mstate;
634     origstart = ms->ms_state_start;
635     do {
636         switch (state) {
637             case LMS_TFAULT:
638             case LMS_DFAULT:
639             case LMS_KFAULT:
640             case LMS_USER_LOCK:
641                 mstimep = &ms->ms_acct[LMS_SYSTEM];
642                 break;
643             default:
644                 mstimep = &ms->ms_acct[state];
645                 break;
646         }
647         ztime = newtime = curtime - ms->ms_state_start;
648         if (newtime < 0) {
649             curtime = gethrtime_unscaled();
650             oldtime = *mstimep - 1; /* force CAS to fail */
651             continue;
652         }
653         oldtime = *mstimep;
654         newtime += oldtime;
655         t->t_mstate = new_state;
656         ms->ms_state_start = curtime;
657     } while (atomic_cas_64((uint64_t *)mstimep, oldtime, newtime) !=
658         oldtime);

660     /*
661     * When the system boots the initial startup thread will have a
662     * ms_state_start of 0 which would add a huge system time to the global
663     * zone. We want to skip aggregating that initial bit of work.
664     */
665     if (origstart != 0) {
666         z = ttozone(t);
667         if (state == LMS_USER)
668             atomic_add_64(&z->zone_utime, ztime);
669         else if (state == LMS_SYSTEM)
670             atomic_add_64(&z->zone_stime, ztime);
671     }

672     /*

```

```

661      * Remember the previous running microstate.
662      */
663      if (state != LMS_SLEEP && state != LMS_STOPPED)
664          ms->ms_prev = state;

666      /*
667      * Switch CPU microstate if appropriate
668      */

670      kpreempt_disable(); /* MUST disable kpreempt before touching t->cpu */

672      ASSERT(t->t_cpu == CPU);

674      /*
675      * When the system boots the initial startup thread will have a
676      * ms_state_start of 0 which would add a huge system time to the global
677      * zone. We want to skip aggregating that initial bit of work.
678      */
679      if (origstart != 0) {
680          z = ttozone(t);
681          if (state == LMS_USER) {
682              CPU_UARRAY_VAL(z->zone_ustate, t->t_cpu->cpu_id,
683                  ZONE_USTATE_UTIME) += ztime;
684          } else if (state == LMS_SYSTEM) {
685              CPU_UARRAY_VAL(z->zone_ustate, t->t_cpu->cpu_id,
686                  ZONE_USTATE_STIME) += ztime;
687          }
688      }

690      if (!CPU_ON_INTR(t->t_cpu) && curthread->t_intr == NULL) {
691          if (new_state == LMS_USER && t->t_cpu->cpu_mstate != CMS_USER)
692              new_cpu_mstate(CMS_USER, curtime);
693          else if (new_state != LMS_USER &&
694                  t->t_cpu->cpu_mstate != CMS_SYSTEM)
695              new_cpu_mstate(CMS_SYSTEM, curtime);
696      }
697      kpreempt_enable();

699      return (ms->ms_prev);
700 }

702 /*
703 * Restore the LWP microstate to the previous runnable state.
704 * Called from disp() with the newly selected lwp.
705 */
706 void
707 restore_mstate(kthread_t *t)
708 {
709     struct mstate *ms;
710     hrtime_t *mstimep;
711     klwp_t *lwp;
712     hrtime_t curtime;
713     hrtime_t waitrq;
714     hrtime_t newtime;
715     hrtime_t oldtime;
716     hrtime_t waittime;
717     zone_t *z;

719     /*
720     * Don't call restore mstate of threads without lwps. (Kernel threads)
721     *
722     * threads with t_intr set shouldn't be in the dispatcher, so assert
723     * that nobody here has t_intr.
724     */
725     ASSERT(t->t_intr == NULL);

```

```

727         if ((lwp = ttolwp(t)) == NULL)
728             return;

730         curtime = gethrtime_unscaled();
731         (void) cpu_update_pct(t, curtime);
732         ms = &lwp->lwp_mstate;
733         ASSERT((unsigned)t->t_mstate < NMSTATES);
734         do {
735             switch (t->t_mstate) {
736                 case LMS_SLEEP:
737                     /*
738                      * Update the timer for the current sleep state.
739                      */
740                     ASSERT((unsigned)ms->ms_prev < NMSTATES);
741                     switch (ms->ms_prev) {
742                         case LMS_TFAULT:
743                         case LMS_DFAULT:
744                         case LMS_KFAULT:
745                         case LMS_USER_LOCK:
746                             mstimep = &ms->ms_acct[ms->ms_prev];
747                             break;
748                         default:
749                             mstimep = &ms->ms_acct[LMS_SLEEP];
750                             break;
751                     }
752                     /*
753                      * Return to the previous run state.
754                      */
755                     t->t_mstate = ms->ms_prev;
756                     break;
757                 case LMS_STOPPED:
758                     mstimep = &ms->ms_acct[LMS_STOPPED];
759                     /*
760                      * Return to the previous run state.
761                      */
762                     t->t_mstate = ms->ms_prev;
763                     break;
764                 case LMS_TFAULT:
765                 case LMS_DFAULT:
766                 case LMS_KFAULT:
767                 case LMS_USER_LOCK:
768                     mstimep = &ms->ms_acct[LMS_SYSTEM];
769                     break;
770                 default:
771                     mstimep = &ms->ms_acct[t->t_mstate];
772                     break;
773             }
774             waitrq = t->t_waitrq; /* hopefully atomic */
775             if (waitrq == 0) {
776                 waitrq = curtime;
777             }
778             t->t_waitrq = 0;
779             newtime = waitrq - ms->ms_state_start;
780             if (newtime < 0) {
781                 curtime = gethrtime_unscaled();
782                 oldtime = *mstimep - 1; /* force CAS to fail */
783                 continue;
784             }
785             oldtime = *mstimep;
786             newtime += oldtime;
787         } while (atomic_cas_64((uint64_t *)mstimep, oldtime, newtime) !=
788                 oldtime);

790     /*
791     * Update the WAIT_CPU timer and per-cpu waitrq total.
792     */

```

```
793     z = ttozone(t);
794     waittime = curtime - waitrq;
795     ms->ms_acct[LMS_WAIT_CPU] += waittime;

797     /*
798     * We are in a disp context where we're not going to migrate CPUs.
799     */
800     CPU_UARRAY_VAL(z->zone_ustate, CPU->cpu_id,
801     ZONE_USTATE_WTIME) += waittime;

786     atomic_add_64(&z->zone_wtime, waittime);
803     CPU->cpu_waitrq += waittime;
804     ms->ms_state_start = curtime;
805 }
_____unchanged_portion_omitted_____
```

```

*****
200174 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/os/zone.c
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
_____unchanged_portion_omitted_____

1910 static int
1911 zone_misc_kstat_update(kstat_t *ksp, int rw)
1912 {
1913     zone_t *zone = ksp->ks_private;
1914     zone_misc_kstat_t *zmp = ksp->ks_data;
1915     hrtime_t hrtime;
1916     uint64_t tmp;
1915     hrtime_t tmp;

1918     if (rw == KSTAT_WRITE)
1919         return (EACCES);

1921     tmp = cpu_uarray_sum(zone->zone_ustate, ZONE_USTATE_STIME);
1922     hrtime = UINT64_OVERFLOW_TO_INT64(tmp);
1923     scalehrtime(&hrtime);
1924     zmp->zm_stime.value.ui64 = hrtime;
1920     tmp = zone->zone_utime;
1921     scalehrtime(&tmp);
1922     zmp->zm_utime.value.ui64 = tmp;
1923     tmp = zone->zone_wtime;
1924     scalehrtime(&tmp);
1925     zmp->zm_wtime.value.ui64 = tmp;
1926     tmp = zone->zone_wtime;
1927     scalehrtime(&tmp);
1928     zmp->zm_wtime.value.ui64 = tmp;

1926     tmp = cpu_uarray_sum(zone->zone_ustate, ZONE_USTATE_UTIME);
1927     hrtime = UINT64_OVERFLOW_TO_INT64(tmp);
1928     scalehrtime(&hrtime);
1929     zmp->zm_utime.value.ui64 = hrtime;

1931     tmp = cpu_uarray_sum(zone->zone_ustate, ZONE_USTATE_WTIME);
1932     hrtime = UINT64_OVERFLOW_TO_INT64(tmp);
1933     scalehrtime(&hrtime);
1934     zmp->zm_wtime.value.ui64 = hrtime;

1936     zmp->zm_avenrun1.value.ui32 = zone->zone_avenrun[0];
1937     zmp->zm_avenrun5.value.ui32 = zone->zone_avenrun[1];
1938     zmp->zm_avenrun15.value.ui32 = zone->zone_avenrun[2];

1940     zmp->zm_ffcap.value.ui32 = zone->zone_ffcap;
1941     zmp->zm_ffnproc.value.ui32 = zone->zone_ffnproc;
1942     zmp->zm_ffnomem.value.ui32 = zone->zone_ffnomem;
1943     zmp->zm_ffmisc.value.ui32 = zone->zone_ffmisc;

1945     zmp->zm_nested_intp.value.ui32 = zone->zone_nested_intp;

1947     zmp->zm_init_pid.value.ui32 = zone->zone_proc_initpid;
1948     zmp->zm_boot_time.value.ui64 = (uint64_t)zone->zone_boot_time;

1950     return (0);
1951 }
_____unchanged_portion_omitted_____

2049 /*
2050 * Called very early on in boot to initialize the ZSD list so that
2051 * zone_key_create() can be called before zone_init(). It also initializes
2052 * portions of zone0 which may be used before zone_init() is called. The

```

```

2053 * variable "global_zone" will be set when zone0 is fully initialized by
2054 * zone_init().
2055 */
2056 void
2057 zone_zsd_init(void)
2058 {
2059     mutex_init(&zonehash_lock, NULL, MUTEX_DEFAULT, NULL);
2060     mutex_init(&zsd_key_lock, NULL, MUTEX_DEFAULT, NULL);
2061     list_create(&zsd_registered_keys, sizeof (struct zsd_entry),
2062         offsetof(struct zsd_entry, zsd_linkage));
2063     list_create(&zone_active, sizeof (zone_t),
2064         offsetof(zone_t, zone_linkage));
2065     list_create(&zone_deathrow, sizeof (zone_t),
2066         offsetof(zone_t, zone_linkage));

2068     mutex_init(&zone0.zone_lock, NULL, MUTEX_DEFAULT, NULL);
2069     mutex_init(&zone0.zone_nlwps_lock, NULL, MUTEX_DEFAULT, NULL);
2070     mutex_init(&zone0.zone_mem_lock, NULL, MUTEX_DEFAULT, NULL);
2071     zone0.zone_shares = 1;
2072     zone0.zone_nlwps = 0;
2073     zone0.zone_nlwps_ctl = INT_MAX;
2074     zone0.zone_nprocs = 0;
2075     zone0.zone_nprocs_ctl = INT_MAX;
2076     zone0.zone_locked_mem = 0;
2077     zone0.zone_locked_mem_ctl = UINT64_MAX;
2078     ASSERT(zone0.zone_max_swap == 0);
2079     zone0.zone_max_swap_ctl = UINT64_MAX;
2080     zone0.zone_max_lofi = 0;
2081     zone0.zone_max_lofi_ctl = UINT64_MAX;
2082     zone0.zone_shmmax = 0;
2083     zone0.zone_ipc.ipcq_shmmni = 0;
2084     zone0.zone_ipc.ipcq_semni = 0;
2085     zone0.zone_ipc.ipcq_msgmni = 0;
2086     zone0.zone_name = GLOBAL_ZONENAME;
2087     zone0.zone_nodename = utsname.nodename;
2088     zone0.zone_domain = srpc_domain;
2089     zone0.zone_hostid = HW_INVALID_HOSTID;
2090     zone0.zone_fs_allowed = NULL;
2091     psecflags_default(&zone0.zone_secflags);
2092     zone0.zone_ref = 1;
2093     zone0.zone_id = GLOBAL_ZONEID;
2094     zone0.zone_status = ZONE_IS_RUNNING;
2095     zone0.zone_rootpath = "/";
2096     zone0.zone_rootpathlen = 2;
2097     zone0.zone_psetid = ZONE_PS_INVALID;
2098     zone0.zone_ncpus = 0;
2099     zone0.zone_ncpus_online = 0;
2100     zone0.zone_proc_initpid = 1;
2101     zone0.zone_initname = initname;
2102     zone0.zone_lockedmem_kstat = NULL;
2103     zone0.zone_swapresv_kstat = NULL;
2104     zone0.zone_nprocs_kstat = NULL;

2106     zone0.zone_stime = 0;
2107     zone0.zone_utime = 0;
2108     zone0.zone_wtime = 0;

2106     list_create(&zone0.zone_ref_list, sizeof (zone_ref_t),
2107         offsetof(zone_ref_t, zref_linkage));
2108     list_create(&zone0.zone_zsd, sizeof (struct zsd_entry),
2109         offsetof(struct zsd_entry, zsd_linkage));
2110     list_insert_head(&zone_active, &zone0);

2112 /*
2113 * The root filesystem is not mounted yet, so zone_rootvp cannot be set
2114 * to anything meaningful. It is assigned to be 'rootdir' in

```



```

2115     * vfs_mountroot().
2116     */
2117     zone0.zone_rootvp = NULL;
2118     zone0.zone_vfslist = NULL;
2119     zone0.zone_bootargs = initargs;
2120     zone0.zone_privset = kmem_alloc(sizeof (priv_set_t), KM_SLEEP);
2121     /*
2122     * The global zone has all privileges
2123     */
2124     priv_fillset(zone0.zone_privset);
2125     /*
2126     * Add p0 to the global zone
2127     */
2128     zone0.zone_zsched = &p0;
2129     p0.p_zone = &zone0;
2130 }
_____ unchanged portion omitted _____

2177 /*
2178  * Called by main() to initialize the zones framework.
2179  */
2180 void
2181 zone_init(void)
2182 {
2183     rctl_dict_entry_t *rde;
2184     rctl_val_t *dval;
2185     rctl_set_t *set;
2186     rctl_alloc_gp_t *gp;
2187     rctl_entity_p_t e;
2188     int res;

2190     ASSERT(curproc == &p0);

2192     /*
2193     * Create ID space for zone IDs. ID 0 is reserved for the
2194     * global zone.
2195     */
2196     zoneid_space = id_space_create("zoneid_space", 1, MAX_ZONEID);

2198     /*
2199     * Initialize generic zone resource controls, if any.
2200     */
2201     rc_zone_cpu_shares = rctl_register("zone.cpu-shares",
2202         RCENTITY_ZONE, RCTL_GLOBAL_SIGNAL_NEVER | RCTL_GLOBAL_DENY_NEVER |
2203         RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_COUNT | RCTL_GLOBAL_SYSLOG_NEVER,
2204         FSS_MAXSHARES, FSS_MAXSHARES, &zone_cpu_shares_ops);

2206     rc_zone_cpu_cap = rctl_register("zone.cpu-cap",
2207         RCENTITY_ZONE, RCTL_GLOBAL_SIGNAL_NEVER | RCTL_GLOBAL_DENY_ALWAYS |
2208         RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_COUNT | RCTL_GLOBAL_SYSLOG_NEVER |
2209         RCTL_GLOBAL_INFINITE,
2210         MAXCAP, MAXCAP, &zone_cpu_cap_ops);

2212     rc_zone_nlwps = rctl_register("zone.max-lwps", RCENTITY_ZONE,
2213         RCTL_GLOBAL_NOACTION | RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_COUNT,
2214         INT_MAX, INT_MAX, &zone_lwps_ops);

2216     rc_zone_nprocs = rctl_register("zone.max-processes", RCENTITY_ZONE,
2217         RCTL_GLOBAL_NOACTION | RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_COUNT,
2218         INT_MAX, INT_MAX, &zone_procs_ops);

2220     /*
2221     * System V IPC resource controls
2222     */
2223     rc_zone_msgmni = rctl_register("zone.max-msg-ids",
2224         RCENTITY_ZONE, RCTL_GLOBAL_DENY_ALWAYS | RCTL_GLOBAL_NOBASIC |

```

```

2225         RCTL_GLOBAL_COUNT, IPC_IDS_MAX, IPC_IDS_MAX, &zone_msgmni_ops);

2227     rc_zone_semjni = rctl_register("zone.max-sem-ids",
2228         RCENTITY_ZONE, RCTL_GLOBAL_DENY_ALWAYS | RCTL_GLOBAL_NOBASIC |
2229         RCTL_GLOBAL_COUNT, IPC_IDS_MAX, IPC_IDS_MAX, &zone_semjni_ops);

2231     rc_zone_shmjni = rctl_register("zone.max-shm-ids",
2232         RCENTITY_ZONE, RCTL_GLOBAL_DENY_ALWAYS | RCTL_GLOBAL_NOBASIC |
2233         RCTL_GLOBAL_COUNT, IPC_IDS_MAX, IPC_IDS_MAX, &zone_shmjni_ops);

2235     rc_zone_shmmax = rctl_register("zone.max-shm-memory",
2236         RCENTITY_ZONE, RCTL_GLOBAL_DENY_ALWAYS | RCTL_GLOBAL_NOBASIC |
2237         RCTL_GLOBAL_BYTES, UINT64_MAX, UINT64_MAX, &zone_shmmax_ops);

2239     /*
2240     * Create a rctl_val with PRIVILEGED, NOACTION, value = 1. Then attach
2241     * this at the head of the rctl_dict_entry for ``zone.cpu-shares``.
2242     */
2243     dval = kmem_cache_alloc(rctl_val_cache, KM_SLEEP);
2244     bzero(dval, sizeof (rctl_val_t));
2245     dval->rcv_value = 1;
2246     dval->rcv_privilege = RCPRIV_PRIVILEGED;
2247     dval->rcv_flagaction = RCTL_LOCAL_NOACTION;
2248     dval->rcv_action_recip_pid = -1;

2250     rde = rctl_dict_lookup("zone.cpu-shares");
2251     (void) rctl_val_list_insert(&rde->rcd_default_value, dval);

2253     rc_zone_locked_mem = rctl_register("zone.max-locked-memory",
2254         RCENTITY_ZONE, RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_BYTES |
2255         RCTL_GLOBAL_DENY_ALWAYS, UINT64_MAX, UINT64_MAX,
2256         &zone_locked_mem_ops);

2258     rc_zone_max_swap = rctl_register("zone.max-swap",
2259         RCENTITY_ZONE, RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_BYTES |
2260         RCTL_GLOBAL_DENY_ALWAYS, UINT64_MAX, UINT64_MAX,
2261         &zone_max_swap_ops);

2263     rc_zone_max_lofi = rctl_register("zone.max-lofi",
2264         RCENTITY_ZONE, RCTL_GLOBAL_NOBASIC | RCTL_GLOBAL_COUNT |
2265         RCTL_GLOBAL_DENY_ALWAYS, UINT64_MAX, UINT64_MAX,
2266         &zone_max_lofi_ops);

2268     /*
2269     * Initialize the ``global zone``.
2270     */
2271     set = rctl_set_create();
2272     gp = rctl_set_init_prealloc(RCENTITY_ZONE);
2273     mutex_enter(&p0.p_lock);
2274     e.rcep_p_zone = &zone0;
2275     e.rcep_t = RCENTITY_ZONE;
2276     zone0.zone_rctls = rctl_set_init(RCENTITY_ZONE, &p0, &e, set,
2277         gp);

2279     zone0.zone_nlwps = p0.p_lwpcnt;
2280     zone0.zone_nprocs = 1;
2281     zone0.zone_ntasks = 1;
2282     mutex_exit(&p0.p_lock);
2283     zone0.zone_restart_init = B_TRUE;
2284     zone0.zone_brand = &native_brand;
2285     rctl_prealloc_destroy(gp);
2286     /*
2287     * pool_default hasn't been initialized yet, so we let pool_init()
2288     * take care of making sure the global zone is in the default pool.
2289     */

```

```

2291  /*
2292  * Initialize global zone kstats
2293  */
2294  zone_kstat_create(&zone0);

2296  /*
2297  * Initialize zone label.
2298  * mlp are initialized when tnzonecfg is loaded.
2299  */
2300  zone0.zone_slabel = l_admin_low;
2301  rw_init(&zone0.zone_mlps.mpl_rwlock, NULL, RW_DEFAULT, NULL);
2302  label_hold(l_admin_low);

2304  /*
2305  * Initialise the lock for the database structure used by mntfs.
2306  */
2307  rw_init(&zone0.zone_mntfs_db_lock, NULL, RW_DEFAULT, NULL);

2309  zone0.zone_ustate = cpu_uarray_zalloc(ZONE_USTATE_MAX, KM_SLEEP);

2311  mutex_enter(&zonehash_lock);
2312  zone_uniqid(&zone0);
2313  ASSERT(zone0.zone_uniqid == GLOBAL_ZONEUNIQID);

2315  zonehashbyid = mod_hash_create_idhash("zone_by_id", zone_hash_size,
2316  mod_hash_null_valdtor);
2317  zonehashbyname = mod_hash_create_strhash("zone_by_name",
2318  zone_hash_size, mod_hash_null_valdtor);
2319  /*
2320  * maintain zonehashbylabel only for labeled systems
2321  */
2322  if (is_system_labeled())
2323  zonehashbylabel = mod_hash_create_extended("zone_by_label",
2324  zone_hash_size, mod_hash_null_keydtor,
2325  mod_hash_null_valdtor, hash_bylabel, NULL,
2326  hash_labelkey_cmp, KM_SLEEP);
2327  zonecount = 1;

2329  (void) mod_hash_insert(zonehashbyid, (mod_hash_key_t)GLOBAL_ZONEID,
2330  (mod_hash_val_t)&zone0);
2331  (void) mod_hash_insert(zonehashbyname, (mod_hash_key_t)zone0.zone_name,
2332  (mod_hash_val_t)&zone0);
2333  if (is_system_labeled()) {
2334  zone0.zone_flags |= ZF_HASHED_LABEL;
2335  (void) mod_hash_insert(zonehashbylabel,
2336  (mod_hash_key_t)zone0.zone_slabel, (mod_hash_val_t)&zone0);
2337  }
2338  mutex_exit(&zonehash_lock);

2340  /*
2341  * We avoid setting zone_kcred until now, since kcred is initialized
2342  * sometime after zone_zsd_init() and before zone_init().
2343  */
2344  zone0.zone_kcred = kcred;
2345  /*
2346  * The global zone is fully initialized (except for zone_rootvp which
2347  * will be set when the root filesystem is mounted).
2348  */
2349  global_zone = &zone0;

2351  /*
2352  * Setup an event channel to send zone status change notifications on
2353  */
2354  res = sysevent_evc_bind(ZONE_EVENT_CHANNEL, &zone_event_chan,
2355  EVCH_CREAT);

```

```

2357  if (res)
2358  panic("Sysevent_evc_bind failed during zone setup.\n");

2360 }

2362 static void
2363 zone_free(zone_t *zone)
2364 {
2365  ASSERT(zone != global_zone);
2366  ASSERT(zone->zone_ntasks == 0);
2367  ASSERT(zone->zone_nlwps == 0);
2368  ASSERT(zone->zone_nprocs == 0);
2369  ASSERT(zone->zone_cred_ref == 0);
2370  ASSERT(zone->zone_kcred == NULL);
2371  ASSERT(zone_status_get(zone) == ZONE_IS_DEAD ||
2372  zone_status_get(zone) == ZONE_IS_UNINITIALIZED);
2373  ASSERT(list_is_empty(&zone->zone_ref_list));

2375  /*
2376  * Remove any zone caps.
2377  */
2378  cpucaps_zone_remove(zone);

2380  ASSERT(zone->zone_cpucap == NULL);

2382  /* remove from deathrow list */
2383  if (zone_status_get(zone) == ZONE_IS_DEAD) {
2384  ASSERT(zone->zone_ref == 0);
2385  mutex_enter(&zone_deathrow_lock);
2386  list_remove(&zone_deathrow, zone);
2387  mutex_exit(&zone_deathrow_lock);
2388  }

2390  list_destroy(&zone->zone_ref_list);
2391  zone_free_zsd(zone);
2392  zone_free_datasets(zone);
2393  list_destroy(&zone->zone_dl_list);

2395  cpu_uarray_free(zone->zone_ustate);

2397  if (zone->zone_rootvp != NULL)
2398  VN_RELE(zone->zone_rootvp);
2399  if (zone->zone_rootpath)
2400  kmem_free(zone->zone_rootpath, zone->zone_rootpathlen);
2401  if (zone->zone_name != NULL)
2402  kmem_free(zone->zone_name, ZONENAME_MAX);
2403  if (zone->zone_slabel != NULL)
2404  label_rele(zone->zone_slabel);
2405  if (zone->zone_nodename != NULL)
2406  kmem_free(zone->zone_nodename, _SYS_NMLN);
2407  if (zone->zone_domain != NULL)
2408  kmem_free(zone->zone_domain, _SYS_NMLN);
2409  if (zone->zone_privset != NULL)
2410  kmem_free(zone->zone_privset, sizeof(priv_set_t));
2411  if (zone->zone_rctls != NULL)
2412  rctl_set_free(zone->zone_rctls);
2413  if (zone->zone_bootargs != NULL)
2414  strfree(zone->zone_bootargs);
2415  if (zone->zone_initname != NULL)
2416  strfree(zone->zone_initname);
2417  if (zone->zone_fs_allowed != NULL)
2418  strfree(zone->zone_fs_allowed);
2419  if (zone->zone_pfxecd != NULL)
2420  klpd_freelist(&zone->zone_pfxecd);
2421  id_free(zoneid_space, zone->zone_id);
2422  mutex_destroy(&zone->zone_lock);

```

```

2423     cv_destroy(&zone->zone_cv);
2424     rw_destroy(&zone->zone_mlps.mlpl_rwlock);
2425     rw_destroy(&zone->zone_mntfs_db_lock);
2426     kmem_free(zone, sizeof (zone_t));
2427 }
-----
unchanged_portion_omitted
-----
3205 /*
3206  * Public interface for updating per-zone load averages.  Called once per
3207  * second.
3208  *
3209  * Based on loadavg_update(), genloadavg() and calcloadavg() from clock.c.
3210  */
3211 void
3212 zone_loadavg_update(void)
3206 zone_loadavg_update()
3213 {
3214     zone_t *zp;
3215     zone_status_t status;
3216     struct loadavg_s *lavg;
3217     hrtime_t zone_total;
3218     uint64_t tmp;
3219     int i;
3220     hrtime_t hr_avg;
3221     int nrun;
3222     static int64_t f[3] = { 135, 27, 9 };
3223     int64_t q, r;

3225     mutex_enter(&zonehash_lock);
3226     for (zp = list_head(&zone_active); zp != NULL;
3227         zp = list_next(&zone_active, zp)) {
3228         mutex_enter(&zp->zone_lock);

3230         /* Skip zones that are on the way down or not yet up */
3231         status = zone_status_get(zp);
3232         if (status < ZONE_IS_READY || status >= ZONE_IS_DOWN) {
3233             /* For all practical purposes the zone doesn't exist. */
3234             mutex_exit(&zp->zone_lock);
3235             continue;
3236         }

3238         /*
3239          * Update the 10 second moving average data in zone_loadavg.
3240          */
3241         lavg = &zp->zone_loadavg;

3243         tmp = cpu_uarray_sum_all(zp->zone_ustate);
3244         zone_total = UINT64_OVERFLOW_TO_INT64(tmp);

3236         zone_total = zp->zone_utime + zp->zone_stime + zp->zone_wtime;
3246         scalehrtime(&zone_total);

3248         /* The zone_total should always be increasing. */
3249         lavg->lg_loads[lavg->lg_cur] = (zone_total > lavg->lg_total) ?
3250             zone_total - lavg->lg_total : 0;
3251         lavg->lg_cur = (lavg->lg_cur + 1) % S_LOADAVG_SZ;
3252         /* lg_total holds the prev. 1 sec. total */
3253         lavg->lg_total = zone_total;

3255         /*
3256          * To simplify the calculation, we don't calculate the load avg.
3257          * until the zone has been up for at least 10 seconds and our
3258          * moving average is thus full.
3259          */
3260         if ((lavg->lg_len + 1) < S_LOADAVG_SZ) {
3261             lavg->lg_len++;

```

```

3262             mutex_exit(&zp->zone_lock);
3263             continue;
3264         }

3266         /* Now calculate the lmin, 5min, 15 min load avg. */
3267         hr_avg = 0;
3268         for (i = 0; i < S_LOADAVG_SZ; i++)
3269             hr_avg += lavg->lg_loads[i];
3270         hr_avg = hr_avg / S_LOADAVG_SZ;
3271         nrun = hr_avg / (NANOSEC / LGRP_LOADAVG_IN_THREAD_MAX);

3273         /* Compute load avg. See comment in calcloadavg() */
3274         for (i = 0; i < 3; i++) {
3275             q = (zp->zone_hp_avenrun[i] >> 16) << 7;
3276             r = (zp->zone_hp_avenrun[i] & 0xffff) << 7;
3277             zp->zone_hp_avenrun[i] +=
3278                 ((nrun - q) * f[i] - ((r * f[i]) >> 16)) >> 4;

3280             /* avenrun[] can only hold 31 bits of load avg. */
3281             if (zp->zone_hp_avenrun[i] <
3282                 ((uint64_t)1<<(31+16-FSHIFT)))
3283                 zp->zone_avenrun[i] = (int32_t)
3284                     (zp->zone_hp_avenrun[i] >> (16 - FSHIFT));
3285             else
3286                 zp->zone_avenrun[i] = 0x7fffffff;
3287         }

3289         mutex_exit(&zp->zone_lock);
3290     }
3291     mutex_exit(&zonehash_lock);
3292 }
-----
unchanged_portion_omitted
-----
4390 /*
4391  * System call to create/initialize a new zone named 'zone_name', rooted
4392  * at 'zone_root', with a zone-wide privilege limit set of 'zone_privs',
4393  * and initialized with the zone-wide rctls described in 'rctlbuf', and
4394  * with labeling set by 'match', 'doi', and 'label'.
4395  *
4396  * If extended error is non-null, we may use it to return more detailed
4397  * error information.
4398  */
4399 static zoneid_t
4400 zone_create(const char *zone_name, const char *zone_root,
4401             const priv_set_t *zone_privs, size_t zone_privssz,
4402             caddr_t rctlbuf, size_t rctlbufsz,
4403             caddr_t zfsbuf, size_t zfsbufsz, int *extended_error,
4404             int match, uint32_t doi, const bslabel_t *label,
4405             int flags)
4406 {
4407     struct zsched_arg zarg;
4408     nvlist_t *rctls = NULL;
4409     proc_t *pp = curproc;
4410     zone_t *zone, *ztmp;
4411     zoneid_t zoneid, start = GLOBAL_ZONEID;
4412     int error;
4413     int error2 = 0;
4414     char *str;
4415     cred_t *zkr;
4416     boolean_t insert_label_hash;

4418     if (secpolicy_zone_config(CRED()) != 0)
4419         return (set_errno(EPERM));

4421     /* can't boot zone from within chroot environment */
4422     if (PTOU(pp)->u_rdir != NULL && PTOU(pp)->u_rdir != rootdir)

```

```

4423         return (zone_create_error(ENOTSUP, ZE_CHROOTED,
4424             extended_error));
4425     /*
4426     * As the first step of zone creation, we want to allocate a zoneid.
4427     * This allocation is complicated by the fact that netstacks use the
4428     * zoneid to determine their stackid, but netstacks themselves are
4429     * freed asynchronously with respect to zone destruction. This means
4430     * that a netstack reference leak (or in principle, an extraordinarily
4431     * long netstack reference hold) could result in a zoneid being
4432     * allocated that in fact corresponds to a stackid from an active
4433     * (referenced) netstack -- unleashing all sorts of havoc when that
4434     * netstack is actually (re)used. (In the abstract, we might wish a
4435     * zoneid to not be deallocated until its last referencing netstack
4436     * has been released, but netstacks lack a backpointer into their
4437     * referencing zone -- and changing them to have such a pointer would
4438     * be substantial, to put it euphemistically.) To avoid this, we
4439     * detect this condition on allocation: if we have allocated a zoneid
4440     * that corresponds to a netstack that's still in use, we warn about
4441     * it (as it is much more likely to be a reference leak than an actual
4442     * netstack reference), free it, and allocate another. That these
4443     * identifiers are allocated out of an ID space assures that we won't
4444     * see the identifier we just allocated.
4445     */
4446     for (;;) {
4447         zoneid = id_alloc(zoneid_space);
4448
4449         if (!netstack_inuse_by_stackid(zoneid_to_netstackid(zoneid)))
4450             break;
4451
4452         id_free(zoneid_space, zoneid);
4453
4454         if (start == GLOBAL_ZONEID) {
4455             start = zoneid;
4456         } else if (zoneid == start) {
4457             /*
4458             * We have managed to iterate over the entire available
4459             * zoneid space -- there are no identifiers available,
4460             * presumably due to some number of leaked netstack
4461             * references. While it's in principle possible for us
4462             * to continue to try, it seems wiser to give up at
4463             * this point to warn and fail explicitly with a
4464             * distinctive error.
4465             */
4466             cmn_err(CE_WARN, "zone_create() failed: all available "
4467                 "zone IDs have netstacks still in use");
4468             return (set_errno(ENFILE));
4469         }
4470
4471         cmn_err(CE_WARN, "unable to reuse zone ID %d: "
4472             "netstack still in use", zoneid);
4473     }
4474
4475     zone = kmem_zalloc(sizeof (zone_t), KM_SLEEP);
4476     zone->zone_id = zoneid;
4477     zone->zone_status = ZONE_IS_UNINITIALIZED;
4478     zone->zone_pool = pool_default;
4479     zone->zone_pool_mod = gethrtime();
4480     zone->zone_psetid = ZONE_PS_INVALID;
4481     zone->zone_ncpus = 0;
4482     zone->zone_ncpus_online = 0;
4483     zone->zone_restart_init = B_TRUE;
4484     zone->zone_brand = &native_brand;
4485     zone->zone_initname = NULL;
4486     mutex_init(&zone->zone_lock, NULL, MUTEX_DEFAULT, NULL);
4487     mutex_init(&zone->zone_nlwps_lock, NULL, MUTEX_DEFAULT, NULL);
4488     mutex_init(&zone->zone_mem_lock, NULL, MUTEX_DEFAULT, NULL);

```

```

4489     cv_init(&zone->zone_cv, NULL, CV_DEFAULT, NULL);
4490     list_create(&zone->zone_ref_list, sizeof (zone_ref_t),
4491         offsetof(zone_ref_t, zref_linkage));
4492     list_create(&zone->zone_zsd, sizeof (struct zsd_entry),
4493         offsetof(struct zsd_entry, zsd_linkage));
4494     list_create(&zone->zone_datasets, sizeof (zone_dataset_t),
4495         offsetof(zone_dataset_t, zd_linkage));
4496     list_create(&zone->zone_dl_list, sizeof (zone_dl_t),
4497         offsetof(zone_dl_t, zdl_linkage));
4498     rw_init(&zone->zone_mlps.mlpl_rwlock, NULL, RW_DEFAULT, NULL);
4499     rw_init(&zone->zone_mntfs_db_lock, NULL, RW_DEFAULT, NULL);
4500
4501     if (flags & ZCF_NET_EXCL) {
4502         zone->zone_flags |= ZF_NET_EXCL;
4503     }
4504
4505     if ((error = zone_set_name(zone, zone_name)) != 0) {
4506         zone_free(zone);
4507         return (zone_create_error(error, 0, extended_error));
4508     }
4509
4510     if ((error = zone_set_root(zone, zone_root)) != 0) {
4511         zone_free(zone);
4512         return (zone_create_error(error, 0, extended_error));
4513     }
4514     if ((error = zone_set_privset(zone, zone_privs, zone_privszz)) != 0) {
4515         zone_free(zone);
4516         return (zone_create_error(error, 0, extended_error));
4517     }
4518
4519     /* initialize node name to be the same as zone name */
4520     zone->zone_nodename = kmem_alloc(_SYS_NMLN, KM_SLEEP);
4521     (void) strncpy(zone->zone_nodename, zone_name, _SYS_NMLN);
4522     zone->zone_nodename[_SYS_NMLN - 1] = '\0';
4523
4524     zone->zone_domain = kmem_alloc(_SYS_NMLN, KM_SLEEP);
4525     zone->zone_domain[0] = '\0';
4526     zone->zone_hostid = HW_INVALID_HOSTID;
4527     zone->zone_shares = 1;
4528     zone->zone_shmmax = 0;
4529     zone->zone_ipc.ipcq_shmuni = 0;
4530     zone->zone_ipc.ipcq_semmni = 0;
4531     zone->zone_ipc.ipcq_msgmni = 0;
4532     zone->zone_bootargs = NULL;
4533     zone->zone_fs_allowed = NULL;
4534
4535     psecflags_default(&zone->zone_secflags);
4536     secflags_zero(&zone0.zone_secflags.psf_lower);
4537     secflags_zero(&zone0.zone_secflags.psf_effective);
4538     secflags_zero(&zone0.zone_secflags.psf_inherit);
4539     secflags_fullset(&zone0.zone_secflags.psf_upper);
4540
4541     zone->zone_initname =
4542         kmem_alloc(strlen(zone_default_initname) + 1, KM_SLEEP);
4543     (void) strcpy(zone->zone_initname, zone_default_initname);
4544     zone->zone_nlwps = 0;
4545     zone->zone_nlwps_ctl = INT_MAX;
4546     zone->zone_nprocs = 0;
4547     zone->zone_nprocs_ctl = INT_MAX;
4548     zone->zone_locked_mem = 0;
4549     zone->zone_locked_mem_ctl = UINT64_MAX;
4550     zone->zone_max_swap = 0;
4551     zone->zone_max_swap_ctl = UINT64_MAX;
4552     zone->zone_max_lofi = 0;
4553     zone->zone_max_lofi_ctl = UINT64_MAX;
4554     zone0.zone_lockedmem_kstat = NULL;

```

```

4551     zone0.zone_swapresv_kstat = NULL;
4553     zone->zone_ustate = cpu_uarray_zalloc(ZONE_USTATE_MAX, KM_SLEEP);
4555     /*
4556      * Zsched initializes the rctls.
4557      */
4558     zone->zone_rctls = NULL;

4560     if ((error = parse_rctls(rctlbuf, rctlbufsz, &rctls)) != 0) {
4561         zone_free(zone);
4562         return (zone_create_error(error, 0, extended_error));
4563     }

4565     if ((error = parse_zfs(zone, zfsbuf, zfsbufsz)) != 0) {
4566         zone_free(zone);
4567         return (set_errno(error));
4568     }

4570     /*
4571      * Read in the trusted system parameters:
4572      * match flag and sensitivity label.
4573      */
4574     zone->zone_match = match;
4575     if (is_system_labeled() && !(zone->zone_flags & ZF_IS_SCRATCH)) {
4576         /* Fail if requested to set doi to anything but system's doi */
4577         if (doi != 0 && doi != default_doi) {
4578             zone_free(zone);
4579             return (set_errno(EINVAL));
4580         }
4581         /* Always apply system's doi to the zone */
4582         error = zone_set_label(zone, label, default_doi);
4583         if (error != 0) {
4584             zone_free(zone);
4585             return (set_errno(error));
4586         }
4587         insert_label_hash = B_TRUE;
4588     } else {
4589         /* all zones get an admin_low label if system is not labeled */
4590         zone->zone_slabel = l_admin_low;
4591         label_hold(l_admin_low);
4592         insert_label_hash = B_FALSE;
4593     }

4595     /*
4596      * Stop all lwps since that's what normally happens as part of fork().
4597      * This needs to happen before we grab any locks to avoid deadlock
4598      * (another lwp in the process could be waiting for the held lock).
4599      */
4600     if (curthread != pp->p_agenttp && !holdlwps(SHOLDFORK)) {
4601         zone_free(zone);
4602         nvlist_free(rctls);
4603         return (zone_create_error(error, 0, extended_error));
4604     }

4606     if (block_mounts(zone) == 0) {
4607         mutex_enter(&pp->p_lock);
4608         if (curthread != pp->p_agenttp)
4609             continue_lwps(pp);
4610         mutex_exit(&pp->p_lock);
4611         zone_free(zone);
4612         nvlist_free(rctls);
4613         return (zone_create_error(error, 0, extended_error));
4614     }

4616     /*

```

```

4617     * Set up credential for kernel access. After this, any errors
4618     * should go through the dance in errout rather than calling
4619     * zone_free directly.
4620     */
4621     zone->zone_kcred = crdup(kcred);
4622     crsetzone(zone->zone_kcred, zone);
4623     priv_intersect(zone->zone_privset, &CR_PPRIV(zone->zone_kcred));
4624     priv_intersect(zone->zone_privset, &CR_EPRIV(zone->zone_kcred));
4625     priv_intersect(zone->zone_privset, &CR_IPRIV(zone->zone_kcred));
4626     priv_intersect(zone->zone_privset, &CR_LPRIV(zone->zone_kcred));

4628     mutex_enter(&zonehash_lock);
4629     /*
4630     * Make sure zone doesn't already exist.
4631     *
4632     * If the system and zone are labeled,
4633     * make sure no other zone exists that has the same label.
4634     */
4635     if ((ztmp = zone_find_all_by_name(zone->zone_name)) != NULL ||
4636         (insert_label_hash &&
4637          (ztmp = zone_find_all_by_label(zone->zone_slabel)) != NULL)) {
4638         zone_status_t status;

4640         status = zone_status_get(ztmp);
4641         if (status == ZONE_IS_READY || status == ZONE_IS_RUNNING)
4642             error = EXIST;
4643         else
4644             error = EBUSY;

4646         if (insert_label_hash)
4647             error2 = ZE_LABELINUSE;

4649         goto errout;
4650     }

4652     /*
4653     * Don't allow zone creations which would cause one zone's rootpath to
4654     * be accessible from that of another (non-global) zone.
4655     */
4656     if (zone_is_nested(zone->zone_rootpath)) {
4657         error = EBUSY;
4658         goto errout;
4659     }

4661     ASSERT(zonecount != 0); /* check for leaks */
4662     if (zonecount + 1 > maxzones) {
4663         error = ENOMEM;
4664         goto errout;
4665     }

4667     if (zone_mount_count(zone->zone_rootpath) != 0) {
4668         error = EBUSY;
4669         error2 = ZE_AREMOUNTS;
4670         goto errout;
4671     }

4673     /*
4674     * Zone is still incomplete, but we need to drop all locks while
4675     * zsched() initializes this zone's kernel process. We
4676     * optimistically add the zone to the hashtable and associated
4677     * lists so a parallel zone_create() doesn't try to create the
4678     * same zone.
4679     */
4680     zonecount++;
4681     (void) mod_hash_insert(zonehashbyid,
4682                          (mod_hash_key_t)(uintptr_t)zone->zone_id,

```

```

4683     (mod_hash_val_t)(uintptr_t)zone);
4684     str = kmem_alloc(strlen(zone->zone_name) + 1, KM_SLEEP);
4685     (void) strcpy(str, zone->zone_name);
4686     (void) mod_hash_insert(zonehashbyname, (mod_hash_key_t)str,
4687     (mod_hash_val_t)(uintptr_t)zone);
4688     if (insert_label_hash) {
4689         (void) mod_hash_insert(zonehashbylabel,
4690         (mod_hash_key_t)zone->zone_slabel, (mod_hash_val_t)zone);
4691         zone->zone_flags |= ZF_HASHED_LABEL;
4692     }

4694     /*
4695     * Insert into active list. At this point there are no 'hold's
4696     * on the zone, but everyone else knows not to use it, so we can
4697     * continue to use it. zsched() will do a zone_hold() if the
4698     * newproc() is successful.
4699     */
4700     list_insert_tail(&zone_active, zone);
4701     mutex_exit(&zonehash_lock);

4703     zarg.zone = zone;
4704     zarg.nvlist = rctls;
4705     /*
4706     * The process, task, and project rctls are probably wrong;
4707     * we need an interface to get the default values of all rctls,
4708     * and initialize zsched appropriately. I'm not sure that that
4709     * makes much of a difference, though.
4710     */
4711     error = newproc(zsched, (void *)&zarg, syscid, minclsyspri, NULL, 0);
4712     if (error != 0) {
4713         /*
4714         * We need to undo all globally visible state.
4715         */
4716         mutex_enter(&zonehash_lock);
4717         list_remove(&zone_active, zone);
4718         if (zone->zone_flags & ZF_HASHED_LABEL) {
4719             ASSERT(zone->zone_slabel != NULL);
4720             (void) mod_hash_destroy(zonehashbylabel,
4721             (mod_hash_key_t)zone->zone_slabel);
4722         }
4723         (void) mod_hash_destroy(zonehashbyname,
4724         (mod_hash_key_t)(uintptr_t)zone->zone_name);
4725         (void) mod_hash_destroy(zonehashbyid,
4726         (mod_hash_key_t)(uintptr_t)zone->zone_id);
4727         ASSERT(zonecount > 1);
4728         zonecount--;
4729         goto errout;
4730     }

4732     /*
4733     * Zone creation can't fail from now on.
4734     */

4736     /*
4737     * Create zone kstats
4738     */
4739     zone_kstat_create(zone);

4741     /*
4742     * Let the other lwps continue.
4743     */
4744     mutex_enter(&pp->p_lock);
4745     if (curthread != pp->p_agenttp)
4746         continuelwps(pp);
4747     mutex_exit(&pp->p_lock);

```

```

4749     /*
4750     * Wait for zsched to finish initializing the zone.
4751     */
4752     zone_status_wait(zone, ZONE_IS_READY);
4753     /*
4754     * The zone is fully visible, so we can let mounts progress.
4755     */
4756     resume_mounts(zone);
4757     nvlist_free(rctls);

4759     return (zoneid);

4761 errout:
4762     mutex_exit(&zonehash_lock);
4763     /*
4764     * Let the other lwps continue.
4765     */
4766     mutex_enter(&pp->p_lock);
4767     if (curthread != pp->p_agenttp)
4768         continuelwps(pp);
4769     mutex_exit(&pp->p_lock);

4771     resume_mounts(zone);
4772     nvlist_free(rctls);
4773     /*
4774     * There is currently one reference to the zone, a cred_ref from
4775     * zone_kcred. To free the zone, we call crfree, which will call
4776     * zone_cred_rele, which will call zone_free.
4777     */
4778     ASSERT(zone->zone_cred_ref == 1);
4779     ASSERT(zone->zone_kcred->cr_ref == 1);
4780     ASSERT(zone->zone_ref == 0);
4781     zkcr = zone->zone_kcred;
4782     zone->zone_kcred = NULL;
4783     crfree(zkcr); /* triggers call to zone_free */
4784     return (zone_create_error(error, error2, extended_error));
4785 }

```

_____unchanged_portion_omitted_____

```

*****
21845 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/sys/Makefile
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 #
2 # CDDL HEADER START
3 #
4 # The contents of this file are subject to the terms of the
5 # Common Development and Distribution License (the "License").
6 # You may not use this file except in compliance with the License.
7 #
8 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9 # or http://www.opensolaris.org/os/licensing.
10 # See the License for the specific language governing permissions
11 # and limitations under the License.
12 #
13 # When distributing Covered Code, include this CDDL HEADER in each
14 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 # If applicable, add the following below this CDDL HEADER, with the
16 # fields enclosed by brackets "[]" replaced with your own identifying
17 # information: Portions Copyright [yyyy] [name of copyright owner]
18 #
19 # CDDL HEADER END
20 #
21 #
22 #
23 # Copyright (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2018, Joyent, Inc.
25 # Copyright 2014, Joyent, Inc. All rights reserved.
26 # Copyright 2013 Garrett D'Amore <garrett@damore.org>
27 # Copyright 2013 Saso Kiselkov. All rights reserved.
28 # Copyright 2015 Igor Kozhukhov <ikozhukhov@gmail.com>
29 # Copyright 2017 Nexenta Systems, Inc.
30 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
31 #
32 include $(SRC)/uts/Makefile.uts
33 #
34 FILEMODE=644
35 #
36 #
37 # Note that the following headers are present in the kernel but
38 # neither installed or shipped as part of the product:
39 # cpuid_drv.h: Private interface for cpuid consumers
40 # unix_bb_info.h: Private interface to kcov
41 # nvme.h Private interface to nvme
42 #
43 #
44 i386_HDRS= \
45 agp/agpamd64gart_io.h \
46 agp/agpdefs.h \
47 agp/agpgart_impl.h \
48 agp/agpmaster_io.h \
49 agp/agptarget_io.h \
50 agpgart.h \
51 asy.h \
52 fd_debug.h \
53 fdc.h \
54 fdmedia.h \
55 firmload.h \
56 gfx_private.h \
57 mouse.h \
58 nvme.h \
59 ucode.h

```

```

61 sparc_HDRS= \
62 mouse.h \
63 scsi/targets/ssddef.h \
64 $(MDESCHDRS)
65 #
66 # Generated headers
67 GENHDRS= \
68 priv_const.h \
69 priv_names.h \
70 usb/usbdevs.h
71 #
72 CHKHDRS= \
73 acpi_drv.h \
74 acct.h \
75 acctctl.h \
76 acl.h \
77 acl_impl.h \
78 aggr.h \
79 aggr_impl.h \
80 aio.h \
81 aio_impl.h \
82 aio_req.h \
83 aiocb.h \
84 ascii.h \
85 asynch.h \
86 atomic.h \
87 attr.h \
88 audio.h \
89 audioio.h \
90 autoconf.h \
91 auxv.h \
92 auxv_386.h \
93 auxv_SPARC.h \
94 avl.h \
95 avl_impl.h \
96 bitmap.h \
97 bitset.h \
98 bl.h \
99 blkdev.h \
100 bofi.h \
101 bofi_impl.h \
102 bpp_io.h \
103 bootstat.h \
104 brand.h \
105 buf.h \
106 bufmod.h \
107 bustypes.h \
108 byteorder.h \
109 callb.h \
110 callo.h \
111 cap_util.h \
112 cpucaps.h \
113 cpucaps_impl.h \
114 ccompile.h \
115 cdio.h \
116 cladm.h \
117 class.h \
118 clconf.h \
119 clock_impl.h \
120 cmlb.h \
121 cmn_err.h \
122 compress.h \
123 condvar.h \
124 condvar_impl.h \
125 conf.h

```

```

126     consdev.h      \|
127     console.h     \|
128     consplat.h    \|
129     vt.h          \|
130     vtdaemon.h   \|
131     kd.h          \|
132     containerof.h \|
133     contract.h    \|
134     contract_impl.h \|
135     copyops.h     \|
136     core.h        \|
137     corectl.h     \|
138     cpc_impl.h    \|
139     cpc_pcbe.h    \|
140     cpr.h         \|
141     cpu_uarray.h \|
142     cpupart.h     \|
143     cpuvar.h      \|
144     crc32.h       \|
145     cred.h        \|
146     cred_impl.h  \|
147     crtctl.h     \|
148     cryptmod.h   \|
149     csioctl.h    \|
150     ctf.h         \|
151     ctfs.h        \|
152     ctfs_impl.h  \|
153     ctf_api.h    \|
154     ctype.h       \|
155     cyclic.h      \|
156     cyclic_impl.h \|
157     dacf.h        \|
158     dacf_impl.h  \|
159     damap.h       \|
160     damap_impl.h \|
161     dc_ki.h       \|
162     ddi.h         \|
163     ddifm.h       \|
164     ddifm_impl.h \|
165     ddi_hp.h      \|
166     ddi_hp_impl.h \|
167     ddi_intr.h    \|
168     ddi_intr_impl.h \|
169     ddi_impldefs.h \|
170     ddi_implfuncs.h \|
171     ddi_obsolete.h \|
172     ddi_periodic.h \|
173     ddidevmap.h  \|
174     ddi_dmareq.h \|
175     ddi_dmapreq.h \|
176     ddi_propdefs.h \|
177     ddi_types.h  \|
178     debug.h       \|
179     des.h         \|
180     devctl.h     \|
181     devcache.h   \|
182     devcache_impl.h \|
183     devfm.h      \|
184     devid_cache.h \|
185     devinfo_impl.h \|
186     devops.h     \|
187     devpolicy.h  \|
188     devpoll.h    \|
189     dirent.h     \|
190     disp.h       \|
191     dkbad.h      \|

```

```

192     dkio.h        \|
193     dkioc_free_util.h \|
194     dklabel.h    \|
195     dl.h         \|
196     dlpi.h       \|
197     dld.h        \|
198     dld_impl.h   \|
199     dld_ioc.h    \|
200     dls.h        \|
201     dls_mgmt.h   \|
202     dls_impl.h   \|
203     dma_i8237A.h \|
204     dnlc.h       \|
205     door.h       \|
206     door_data.h  \|
207     door_impl.h  \|
208     dtrace.h     \|
209     dtrace_impl.h \|
210     dumpadm.h    \|
211     dumphdr.h    \|
212     ecppsys.h    \|
213     ecppio.h     \|
214     ecppreg.h    \|
215     ecppvar.h    \|
216     edonr.h      \|
217     efi_partition.h \|
218     elf.h        \|
219     elf_386.h    \|
220     elf_SPARC.h  \|
221     elf_notes.h  \|
222     elf_amd64.h  \|
223     elftypes.h   \|
224     emul64.h     \|
225     emul64cmd.h \|
226     emul64var.h \|
227     epm.h        \|
228     epoll.h      \|
229     errno.h      \|
230     errorq.h     \|
231     errorq_impl.h \|
232     esunddi.h    \|
233     ethernet.h   \|
234     euc.h        \|
235     eucioctl.h   \|
236     eventfd.h    \|
237     exacct.h     \|
238     exacct_catalog.h \|
239     exacct_impl.h \|
240     exec.h       \|
241     exechdr.h    \|
242     extdirent.h  \|
243     fault.h      \|
244     fasttrap.h   \|
245     fasttrap_impl.h \|
246     fbio.h       \|
247     fbuf.h       \|
248     fcntl.h      \|
249     fct.h        \|
250     fct_defines.h \|
251     fctio.h      \|
252     fdbuffer.h   \|
253     fdio.h       \|
254     feature_tests.h \|
255     fem.h        \|
256     file.h       \|
257     filio.h      \|

```



```

258 flock.h \
259 flock_impl.h \
260 fork.h \
261 fss.h \
262 fsspriocntl.h \
263 fsid.h \
264 fssnap.h \
265 fssnap_if.h \
266 fstyp.h \
267 ftrace.h \
268 fx.h \
269 fxpriocntl.h \
270 gfs.h \
271 gld.h \
272 gldpriv.h \
273 group.h \
274 hdio.h \
275 hook.h \
276 hook_event.h \
277 hook_impl.h \
278 hwconf.h \
279 ia.h \
280 iapriocntl.h \
281 ibpart.h \
282 id32.h \
283 idmap.h \
284 ieeeefp.h \
285 id_space.h \
286 instance.h \
287 int_const.h \
288 int_fmto.h \
289 int_limits.h \
290 int_types.h \
291 inttypes.h \
292 ioccom.h \
293 ioctl.h \
294 ipc.h \
295 ipc_impl.h \
296 ipc_rctl.h \
297 ipd.h \
298 ipmi.h \
299 isa_defs.h \
300 iscsi_authclient.h \
301 iscsi_authclientglue.h \
302 iscsi_protocol.h \
303 jioctl.h \
304 kbd.h \
305 kbdreg.h \
306 kbio.h \
307 kcpc.h \
308 kdi.h \
309 kdi_impl.h \
310 kiconv.h \
311 kiconv_big5_utf8.h \
312 kiconv_ck_common.h \
313 kiconv_cp950hkscs_utf8.h \
314 kiconv_emeal.h \
315 kiconv_emea2.h \
316 kiconv_euckr_utf8.h \
317 kiconv_euctw_utf8.h \
318 kiconv_gb18030_utf8.h \
319 kiconv_gb2312_utf8.h \
320 kiconv_hkscs_utf8.h \
321 kiconv_ja.h \
322 kiconv_ja_jis_to_unicode.h \
323 kiconv_ja_unicode_to_jis.h \

```

```

324 kiconv_ko.h \
325 kiconv_latin1.h \
326 kiconv_sc.h \
327 kiconv_tc.h \
328 kiconv_uhc_utf8.h \
329 kiconv_utf8_big5.h \
330 kiconv_utf8_cp950hkscs.h \
331 kiconv_utf8_euckr.h \
332 kiconv_utf8_euctw.h \
333 kiconv_utf8_gb18030.h \
334 kiconv_utf8_gb2312.h \
335 kiconv_utf8_hkscs.h \
336 kiconv_utf8_uhc.h \
337 kidmap.h \
338 klpd.h \
339 klwp.h \
340 kmdb.h \
341 kmem.h \
342 kmem_impl.h \
343 kobj.h \
344 kobj_impl.h \
345 ksocket.h \
346 kstat.h \
347 kstr.h \
348 ksyms.h \
349 ksynch.h \
350 ldterm.h \
351 lgrp.h \
352 lgrp_user.h \
353 libc_kernel.h \
354 link.h \
355 list.h \
356 list_impl.h \
357 llcl.h \
358 loadavg.h \
359 lock.h \
360 lockfs.h \
361 lockstat.h \
362 lofi.h \
363 lofi_impl.h \
364 log.h \
365 logindmux.h \
366 logindmux_impl.h \
367 lwp.h \
368 lwp_timer_impl.h \
369 lwp_upimutex_impl.h \
370 lpif.h \
371 mac.h \
372 mac_client.h \
373 mac_client_impl.h \
374 mac_ether.h \
375 mac_flow.h \
376 mac_flow_impl.h \
377 mac_impl.h \
378 mac_provider.h \
379 mac_soft_ring.h \
380 mac_stat.h \
381 machelf.h \
382 map.h \
383 md4.h \
384 md5.h \
385 md5_consts.h \
386 mdi_impldefs.h \
387 mem.h \
388 mem_config.h \
389 memlist.h \

```

```

390     mkdev.h          \|
391     mhd.h            \|
392     mii.h            \|
393     miiregs.h        \|
394     mixer.h          \|
395     mman.h           \|
396     mmapobj.h        \|
397     mntent.h         \|
398     mntio.h          \|
399     mnttab.h         \|
400     modctl.h         \|
401     mode.h           \|
402     model.h          \|
403     modhash.h        \|
404     modhash_impl.h  \|
405     mount.h          \|
406     mouse.h          \|
407     msacct.h         \|
408     msg.h            \|
409     msg_impl.h       \|
410     msio.h           \|
411     msreg.h          \|
412     mtio.h           \|
413     multidata.h     \|
414     multidata_impl.h \|
415     mutex.h          \|
416     nbmlock.h       \|
417     ndifm.h          \|
418     ndi_impldefs.h  \|
419     net80211.h       \|
420     net80211_amrr.h \|
421     net80211_crypto.h \|
422     net80211_ht.h   \|
423     net80211_proto.h \|
424     netconfig.h     \|
425     neti.h           \|
426     netstack.h      \|
427     nexusdefs.h     \|
428     note.h           \|
429     null.h           \|
430     nvme.h           \|
431     nvpair.h         \|
432     nvpair_impl.h   \|
433     objfs.h          \|
434     objfs_impl.h    \|
435     ontrap.h         \|
436     open.h           \|
437     openpromio.h    \|
438     panic.h          \|
439     param.h          \|
440     pathconf.h       \|
441     pathname.h       \|
442     pattr.h          \|
443     queue.h          \|
444     serializer.h     \|
445     pbio.h           \|
446     pccard.h         \|
447     pci.h            \|
448     pcie.h           \|
449     pci_impl.h       \|
450     pci_tools.h      \|
451     pcmcia.h         \|
452     ptypes.h         \|
453     pfmod.h          \|
454     pg.h             \|
455     pghw.h           \|

```

```

456     physmem.h       \|
457     pkp_hash.h      \|
458     pm.h             \|
459     policy.h         \|
460     poll.h           \|
461     poll_impl.h     \|
462     pool.h           \|
463     pool_impl.h     \|
464     pool_pset.h     \|
465     port.h           \|
466     port_impl.h     \|
467     port_kernel.h   \|
468     portif.h        \|
469     ppmio.h          \|
470     pppt_ic_if.h    \|
471     pppt_ioctl.h    \|
472     priocntl.h      \|
473     priv.h           \|
474     priv_impl.h     \|
475     prnio.h          \|
476     proc.h           \|
477     processor.h     \|
478     procfs.h         \|
479     procset.h       \|
480     project.h       \|
481     protosw.h       \|
482     prsystem.h      \|
483     pset.h           \|
484     pshot.h          \|
485     ptem.h           \|
486     ptms.h           \|
487     ptyvar.h        \|
488     raidioctl.h     \|
489     ramdisk.h       \|
490     random.h         \|
491     rctl.h           \|
492     rctl_impl.h     \|
493     rds.h            \|
494     reboot.h        \|
495     refstr.h         \|
496     refstr_impl.h   \|
497     resource.h      \|
498     rliocntl.h      \|
499     rt.h             \|
500     rtpriocntl.h    \|
501     rwlock.h        \|
502     rwlock_impl.h   \|
503     rwstlock.h      \|
504     sad.h            \|
505     schedctl.h      \|
506     sdt.h            \|
507     secflags.h      \|
508     select.h        \|
509     sem.h            \|
510     sem_impl.h      \|
511     sema_impl.h     \|
512     semaphore.h     \|
513     sendfile.h      \|
514     ser_sync.h       \|
515     session.h       \|
516     sha1.h           \|
517     sha1_consts.h   \|
518     sha2.h           \|
519     sha2_consts.h   \|
520     share.h          \|
521     shm.h            \|

```

```

522     shm_impl.h      \|
523     sid.h           \|
524     siginfo.h       \|
525     signal.h        \|
526     signalfd.h     \|
527     skein.h         \|
528     sleepq.h        \|
529     smbios.h        \|
530     smbios_impl.h  \|
531     subject.h       \|
532     socket.h        \|
533     socket_impl.h  \|
534     socket_proto.h \|
535     socketvar.h     \|
536     sockfilter.h   \|
537     sockio.h        \|
538     soundcard.h    \|
539     queue.h         \|
540     queue_impl.h   \|
541     srn.h           \|
542     sservice.h     \|
543     stat.h          \|
544     statfs.h       \|
545     statvfs.h      \|
546     stdbool.h      \|
547     stddef.h       \|
548     stdint.h       \|
549     stermio.h      \|
550     stmf.h          \|
551     stmf_defines.h \|
552     stmf_ioctl.h   \|
553     stmf_sbd_ioctl.h \|
554     stream.h       \|
555     strft.h        \|
556     strlog.h       \|
557     strmddep.h     \|
558     stropts.h      \|
559     strredir.h     \|
560     strstat.h      \|
561     strsubr.h      \|
562     strsun.h       \|
563     strtty.h       \|
564     sunddi.h       \|
565     sunldi.h       \|
566     sunldi_impl.h \|
567     sunmdi.h       \|
568     sunndi.h       \|
569     sunos_dhcp_class.h \|
570     sunpm.h        \|
571     suntpl.h       \|
572     suntty.h       \|
573     swap.h         \|
574     synch.h        \|
575     sysdc.h        \|
576     sysdc_impl.h  \|
577     syscall.h     \|
578     sysconf.h     \|
579     sysconfig.h   \|
580     sysevent.h    \|
581     sysevent_impl.h \|
582     sysinfo.h     \|
583     syslog.h       \|
584     sysmacros.h   \|
585     sysmsg_impl.h \|
586     systeminfo.h \|
587     system.h      \|

```

```

588     task.h          \|
589     taskq.h         \|
590     taskq_impl.h   \|
591     t_kuser.h       \|
592     t_lock.h        \|
593     telioctl.h      \|
594     termio.h        \|
595     termios.h       \|
596     termiox.h       \|
597     thread.h        \|
598     ticlts.h        \|
599     ticots.h        \|
600     ticotsord.h    \|
601     tihdr.h         \|
602     time.h          \|
603     time_impl.h    \|
604     time_std_impl.h \|
605     timeb.h         \|
606     timer.h         \|
607     timerfd.h       \|
608     times.h         \|
609     timex.h         \|
610     timod.h         \|
611     tirdwr.h        \|
612     tiuser.h        \|
613     tl.h            \|
614     tn timer.h      \|
615     tn timer_com.h \|
616     tn timer_probe.h \|
617     tn timer_writer.h \|
618     todio.h         \|
619     tpicommon.h     \|
620     ts.h            \|
621     tspriocntl.h   \|
622     ttcompat.h     \|
623     ttold.h         \|
624     tty.h           \|
625     ttychars.h     \|
626     ttydev.h       \|
627     tuneable.h     \|
628     turnstile.h    \|
629     types.h         \|
630     types32.h       \|
631     tzfile.h        \|
632     u8_textprep.h  \|
633     u8_textprep_data.h \|
634     uadmin.h        \|
635     ucred.h         \|
636     uio.h           \|
637     ulimit.h        \|
638     un.h            \|
639     unistd.h        \|
640     user.h          \|
641     ustat.h         \|
642     utime.h         \|
643     utsname.h       \|
644     utssys.h        \|
645     uuid.h          \|
646     va_impl.h       \|
647     va_list.h       \|
648     var.h           \|
649     varargs.h       \|
650     vfs.h           \|
651     vfs_opreg.h     \|
652     vfstab.h        \|
653     vgareg.h        \|

```

```

654     videodev2.h      \
655     visual_io.h     \
656     vlan.h          \
657     vm.h            \
658     vm_usage.h      \
659     vmem.h          \
660     vmem_impl.h     \
661     vmsystem.h      \
662     vnic.h          \
663     vnic_impl.h     \
664     vnode.h         \
665     vscan.h        \
666     vtoc.h         \
667     vtrace.h       \
668     vuid_event.h   \
669     vuid_wheel.h   \
670     vuid_queue.h   \
671     vuid_state.h   \
672     vuid_store.h   \
673     wait.h         \
674     waitq.h        \
675     watchpoint.h   \
676     winlockio.h    \
677     zcons.h        \
678     zone.h         \
679     xti_inet.h     \
680     xti_osi.h      \
681     xti_xtiopt.h   \
682     zmod.h         \

684 HDRS= \
685     $(GENHDRS) \
686     $(CHKHDRS) \

688 AUDIOHDRS= \
689     ac97.h \
690     audio_common.h \
691     audio_driver.h \
692     audio_oss.h \
693     g711.h \

695 AVHDRS= \
696     iec61883.h \

698 BSCHDRS= \
699     bscbus.h \
700     bscv_impl.h \
701     lom_ebuscodes.h \
702     lom_io.h \
703     lom_priv.h \
704     lombus.h \

706 MDESCHDRS= \
707     mdesc.h \
708     mdesc_impl.h \

710 CPUDRVHDRS= \
711     cpudrv.h \

713 CRYPTOHDRS= \
714     elfsign.h \
715     ioctl.h \
716     ioctladm.h \
717     common.h \
718     impl.h \
719     spi.h \

```

```

720     api.h \
721     ops_impl.h \
722     sched_impl.h \

724 DCAMHDRS= \
725     dcam1394_io.h \

727 IBHDRS= \
728     ib_types.h \
729     ib_pkt_hdrs.h \

731 IBTLHDRS= \
732     ibtl_types.h \
733     ibtl_status.h \
734     ibti.h \
735     ibti_cm.h \
736     ibci.h \
737     ibti_common.h \
738     ibvti.h \
739     ibtl_ci_types.h \

741 IBTLMPLHDRS= \
742     ibtl_util.h \

744 IBNEXHDRS= \
745     ibnex_devctl.h \

747 IBMFHDRS= \
748     ibmf.h \
749     ibmf_msg.h \
750     ibmf_saa.h \
751     ibmf_utils.h \

753 IBMGTHDRS= \
754     ib_dm_attr.h \
755     ib_mad.h \
756     sm_attr.h \
757     sa_recs.h \

759 IBDHDRS= \
760     ibd.h \

762 OFHDRS= \
763     ofa_solaris.h \
764     ofed_kernel.h \

766 RDMAHDRS= \
767     ib_addr.h \
768     ib_user_mad.h \
769     ib_user_sa.h \
770     ib_user_verbs.h \
771     ib_verbs.h \
772     rdma_cm.h \
773     rdma_user_cm.h \

775 SOL_UVERBSHDRS= \
776     sol_uverbs.h \
777     sol_uverbs2ucma.h \
778     sol_uverbs_comp.h \
779     sol_uverbs_hca.h \
780     sol_uverbs_qp.h \
781     sol_uverbs_event.h \

783 SOL_UMADHDRS= \
784     sol_umad.h \

```

```

786 SOL_UCMAHDRS= \
787     sol_ucma.h \
788     sol_rdma_user_cm.h

790 SOL_OFSHDRS= \
791     sol_cma.h \
792     sol_ib_cma.h \
793     sol_ofs_common.h \
794     sol_kverb_impl.h

796 TAVORHDRS= \
797     tavor_ioctl.h

799 HERMONHDRS= \
800     hermon_ioctl.h

802 MLNXHDRS= \
803     mlnx_umap.h

805 IDMHDRS= \
806     idm.h \
807     idm_impl.h \
808     idm_so.h \
809     idm_text.h \
810     idm_transport.h \
811     idm_conn_sm.h

813 ISCSITHDRS= \
814     radius_packet.h \
815     radius_protocol.h \
816     chap.h \
817     isns_protocol.h \
818     iscsi_if.h \
819     iscsit_common.h

821 ISOHDRS= \
822     signal_iso.h

824 FMHDRS= \
825     protocol.h \
826     util.h

828 FMFSHDRS= \
829     zfs.h

831 FMIOHDRS= \
832     ddi.h \
833     disk.h \
834     pci.h \
835     scsi.h \
836     sun4upci.h \
837     opl_mc_fm.h

839 FSHDRS= \
840     autofs.h \
841     decomp.h \
842     dv_node.h \
843     sdev_impl.h \
844     fifonode.h \
845     hsfs_isospec.h \
846     hsfs_node.h \
847     hsfs_rrip.h \
848     hsfs_spec.h \
849     hsfs_susp.h \
850     lofs_info.h \
851     lofs_node.h

```

```

852     mntdata.h \
853     namenode.h \
854     pc_dir.h \
855     pc_fs.h \
856     pc_label.h \
857     pc_node.h \
858     pxfs_ki.h \
859     snode.h \
860     swapnode.h \
861     tmp.h \
862     tmpnode.h \
863     udf_inode.h \
864     udf_volume.h \
865     ufs_acl.h \
866     ufs_bio.h \
867     ufs_filio.h \
868     ufs_fs.h \
869     ufs_fsdire.h \
870     ufs_inode.h \
871     ufs_lockfs.h \
872     ufs_log.h \
873     ufs_mount.h \
874     ufs_panic.h \
875     ufs_prot.h \
876     ufs_quota.h \
877     ufs_snap.h \
878     ufs_trans.h \
879     zfs.h \
880     zut.h

882 SCSIHDRS= \
883     scsi.h \
884     scsi_address.h \
885     scsi_ctl.h \
886     scsi_fm.h \
887     scsi_names.h \
888     scsi_params.h \
889     scsi_pkt.h \
890     scsi_resource.h \
891     scsi_types.h \
892     scsi_watch.h

894 SCSSICONFHDRS= \
895     autoconf.h \
896     device.h

898 SCSSIGENHDRS= \
899     commands.h \
900     dad_mode.h \
901     inquiry.h \
902     message.h \
903     mode.h \
904     persist.h \
905     sense.h \
906     sff_frames.h \
907     smp_frames.h \
908     status.h

910 SCSSIMPLHDRS= \
911     commands.h \
912     inquiry.h \
913     mode.h \
914     scsi_reset_notify.h \
915     scsi_sas.h \
916     sense.h \
917     services.h

```

```

918     smp_transport.h \
919     spc3_types.h   \
920     status.h       \
921     transport.h    \
922     types.h        \
923     uscsi.h        \
924     usmp.h         \

926 SCSTITARGETSHDRS= \
927     ses.h           \
928     sesio.h        \
929     sgendef.h      \
930     stdef.h        \
931     sddef.h        \
932     smp.h          \

934 SCSIADHDRS=

936 SCASICADHDRS=

938 SCIIISCSIHDRS= \
939     iscsi_door.h  \
940     iscsi_if.h    \

942 SCIVHCIHDRS= \
943     scsi_vhci.h  \
944     mpapi_impl.h \
945     mpapi_scsi_vhci.h \

947 SDCARDHDRS= \
948     sda.h        \
949     sda_impl.h   \
950     sda_ioctl.h  \

952 FC4HDRS= \
953     fc_transport.h \
954     linkapp.h     \
955     fc.h          \
956     fcp.h        \
957     fcal_transport.h \
958     fcal.h       \
959     fcal_linkapp.h \
960     fcio.h       \

962 FCHDRS= \
963     fc.h      \
964     fcio.h   \
965     fc_types.h \
966     fc_appif.h \

968 FCIMPLHDRS= \
969     fc_error.h \
970     fcph.h     \

972 FCULPHDRS= \
973     fcp_util.h \
974     fcsm.h     \

976 SATAGENHDRS= \
977     sata_hba.h \
978     sata_defs.h \
979     sata_cfgadm.h \

981 SYSEVENTHDRS= \
982     ap_driver.h \
983     dev.h       \

```

```

984     domain.h \
985     dr.h     \
986     env.h   \
987     eventdefs.h \
988     imp.h   \
989     pwrctl.h \
990     vrrp.h  \

992 CONTRACTHDRS= \
993     process.h   \
994     process_impl.h \
995     device.h    \
996     device_impl.h \

998 USBHDRS= \
999     usba.h \
1000     usbai.h \

1002 USBAUDHDRS= \
1003     usb_audio.h \

1005 USBHUBHDRS= \
1006     hub.h \
1007     hubd_impl.h \

1009 USBHIDHDRS= \
1010     hid.h \

1012 USBMSHDRS= \
1013     usb_bulkonly.h \
1014     usb_cbi.h \

1016 USBPRNHDRS= \
1017     usb_printer.h \

1019 USBDCCHDRS= \
1020     usb_cdc.h \

1022 USBVIDHDRS= \
1023     usbvc.h \

1025 USBWCMHDRS= \
1026     usbwcm.h \

1028 UGENHDRS= \
1029     usb_ugen.h \

1031 HOTPLUGHDRS= \
1032     hpcsvc.h \
1033     hpctrl.h \

1035 HOTPLUGPCIHDRS= \
1036     pcicfg.h \
1037     pcihp.h \

1039 RSMHDRS= \
1040     rsm.h \
1041     rsm_common.h \
1042     rsmapi_common.h \
1043     rsmpl.h \
1044     rsmpl_driver.h \
1045     rsmka_path_int.h \

1047 TSOLHDRS= \
1048     label.h \
1049     label_macro.h \

```

```

1050     priv.h           \
1051     tndb.h           \
1052     tsyscall.h      \

1054 I1394HDRS=        \
1055     cmd1394.h        \
1056     id1394.h        \
1057     ieeel212.h       \
1058     ieeel394.h       \
1059     ixl1394.h        \
1060     sl394_impl.h    \
1061     t1394.h         \

1063 # "cmdk" headers used on sparc
1064 SDKTPHDRS=         \
1065     dadkio.h         \
1066     fdisk.h          \

1068 # "cmdk" headers used on i386
1069 DKTPHDRS=          \
1070     altsctr.h        \
1071     bbh.h            \
1072     cm.h             \
1073     cmddev.h         \
1074     cmdk.h           \
1075     cmpkt.h          \
1076     controller.h    \
1077     dadev.h          \
1078     dadk.h           \
1079     dadkio.h         \
1080     fctypes.h        \
1081     fdisk.h          \
1082     flowctrl.h      \
1083     gda.h            \
1084     quetypes.h       \
1085     queue.h          \
1086     tgcom.h          \
1087     tgdk.h           \

1089 # "pc" header files used on i386
1090 PCHDRS=             \
1091     avintr.h         \
1092     dma_engine.h     \
1093     i8272A.h         \
1094     pcic_reg.h       \
1095     pcic_var.h       \
1096     pic.h            \
1097     pit.h            \
1098     rtc.h            \

1100 NXGEHDRS=           \
1101     nxge.h           \
1102     nxge_common.h   \
1103     nxge_common_impl.h \
1104     nxge_defs.h     \
1105     nxge_hw.h        \
1106     nxge_impl.h     \
1107     nxge_ipp.h       \
1108     nxge_ipp_hw.h   \
1109     nxge_mac.h       \
1110     nxge_mac_hw.h   \
1111     nxge_fflp.h     \
1112     nxge_fflp_hw.h  \
1113     nxge_mii.h       \
1114     nxge_rxdma.h    \
1115     nxge_rxdma_hw.h \

```

```

1116     nxge_txc.h      \
1117     nxge_txc_hw.h   \
1118     nxge_txdma.h    \
1119     nxge_txdma_hw.h \
1120     nxge_virtual.h  \
1121     nxge_espc.h     \

1123 include Makefile.sysshdrs

1125 dcam/%.check: dcam/%.h
1126     $(DOT_H_CHECK)

1128 CHECKHDRS=          \
1129     $(MACH)_HDRS:%.h=%%.check) \
1130     $(AUDIOHDRS:%.h=audio/%.check) \
1131     $(AVHDRS:%.h=av/%.check) \
1132     $(BSCHDRS:%.h=%%.check) \
1133     $(CHKHDRS:%.h=%%.check) \
1134     $(CPUDRVHDRS:%.h=%%.check) \
1135     $(CRYPTOHDRS:%.h=crypto/%.check) \
1136     $(DCAMHDRS:%.h=dcam/%.check) \
1137     $(FC4HDRS:%.h=fc4/%.check) \
1138     $(FCHDRS:%.h=fibre-channel/%.check) \
1139     $(FCIMPLHDRS:%.h=fibre-channel/impl/%.check) \
1140     $(FCULPHDRS:%.h=fibre-channel/ulp/%.check) \
1141     $(IBHDRS:%.h=ib/%.check) \
1142     $(IBDHDRS:%.h=ib/clients/ibd/%.check) \
1143     $(IBTLHDRS:%.h=ib/ibtl/%.check) \
1144     $(IBTLIMPLHDRS:%.h=ib/ibtl/impl/%.check) \
1145     $(IBNEXHDRS:%.h=ib/ibnex/%.check) \
1146     $(IBMGTHDRS:%.h=ib/mgt/%.check) \
1147     $(IBMFHDRS:%.h=ib/mgt/ibmf/%.check) \
1148     $(OFHDRS:%.h=ib/clients/of/%.check) \
1149     $(RDMAHDRS:%.h=ib/clients/of/rdma/%.check) \
1150     $(SOL_UVERBSHDRS:%.h=ib/clients/of/sol_uverbs/%.check) \
1151     $(SOL_UCMAHDRS:%.h=ib/clients/of/sol_ucma/%.check) \
1152     $(SOL_OFSHDRS:%.h=ib/clients/of/sol_ofs/%.check) \
1153     $(TAVORHDRS:%.h=ib/adapters/tavor/%.check) \
1154     $(HERMONHDRS:%.h=ib/adapters/hermon/%.check) \
1155     $(MLNXHDRS:%.h=ib/adapters/%.check) \
1156     $(IDMHDRS:%.h=idm/%.check) \
1157     $(ISCSIHDRS:%.h=iscsi/%.check) \
1158     $(ISCSITHDRS:%.h=iscsit/%.check) \
1159     $(ISOHDRS:%.h=iso/%.check) \
1160     $(FMHDRS:%.h=fm/%.check) \
1161     $(FMFSHDRS:%.h=fm/fs/%.check) \
1162     $(FMIOHDRS:%.h=fm/io/%.check) \
1163     $(FSHDRS:%.h=fs/%.check) \
1164     $(SCSIHDRS:%.h=scsi/%.check) \
1165     $(SCSIADHDRS:%.h=scsi/adapters/%.check) \
1166     $(SCSICONFHDRS:%.h=scsi/conf/%.check) \
1167     $(SCSIIMPLHDRS:%.h=scsi/impl/%.check) \
1168     $(SCSIISCSIHDRS:%.h=scsi/adapters/%.check) \
1169     $(SCSIGHDRS:%.h=scsi/generic/%.check) \
1170     $(SCSITARGETSHDRS:%.h=scsi/targets/%.check) \
1171     $(SCSIVHCIHDRS:%.h=scsi/adapters/%.check) \
1172     $(SATAGENHDRS:%.h=sata/%.check) \
1173     $(SDCARDHDRS:%.h=sdcard/%.check) \
1174     $(SYSEVENTHDRS:%.h=sysevent/%.check) \
1175     $(CONTRACTHDRS:%.h=contract/%.check) \
1176     $(USBAUDHDRS:%.h=usb/clients/audio/%.check) \
1177     $(USBHUBHDRS:%.h=usb/hubd/%.check) \
1178     $(USBHIDHDRS:%.h=usb/clients/hid/%.check) \
1179     $(USBMSHDRS:%.h=usb/clients/mass_storage/%.check) \
1180     $(USBPRNHDRS:%.h=usb/clients/printer/%.check) \
1181     $(USBCDCHDRS:%.h=usb/clients/usbcdc/%.check) \

```

```

1182 $(USBVIDHDRS:%.h=usb/clients/video/usbvc/%.check) \
1183 $(USBWCMHDRS:%.h=usb/clients/usbinput/usbwcm/%.check) \
1184 $(UGENHDRS:%.h=usb/clients/ugen/%.check) \
1185 $(USBHDRS:%.h=usb/%.check) \
1186 $(I1394HDRS:%.h=1394/%.check) \
1187 $(RSMHDRS:%.h=rsm/%.check) \
1188 $(TSOLHDRS:%.h=tsol/%.check) \
1189 $(NXGEHDRS:%.h=nxge/%.check)

```

```
1192 .KEEP_STATE:
```

```

1194 .PARALLEL: \
1195 $(CHECKHDRS) \
1196 $(ROOTHDRS) \
1197 $(ROOTAUDHDRS) \
1198 $(ROOTAVHDRS) \
1199 $(ROOTCRYPTOHDRS) \
1200 $(ROOTDCAMHDRS) \
1201 $(ROOTISOHDRS) \
1202 $(ROOTIDMHDRS) \
1203 $(ROOTISCSIHDRS) \
1204 $(ROOTISCSITHDRS) \
1205 $(ROOTFC4HDRS) \
1206 $(ROOTFCHDRS) \
1207 $(ROOTFCIMPLHDRS) \
1208 $(ROOTFCULPHDRS) \
1209 $(ROOTFMHDRS) \
1210 $(ROOTFMIOHDRS) \
1211 $(ROOTFMFSDHDRS) \
1212 $(ROOTFSDHDRS) \
1213 $(ROOTIBDHDRS) \
1214 $(ROOTIBHDRS) \
1215 $(ROOTIBTLHDRS) \
1216 $(ROOTIBTLIMPLHDRS) \
1217 $(ROOTIBNEXHDRS) \
1218 $(ROOTIBMGTHDRS) \
1219 $(ROOTIBMFHDRS) \
1220 $(ROOTOFHDRS) \
1221 $(ROOTRDMAHDRS) \
1222 $(ROOTSOL_OFSDHDRS) \
1223 $(ROOTSOL_UMADHDRS) \
1224 $(ROOTSOL_UVERBSHDRS) \
1225 $(ROOTSOL_UCMAHDRS) \
1226 $(ROOTTAVORHDRS) \
1227 $(ROOTTHERMONHDRS) \
1228 $(ROOTMLNXHDRS) \
1229 $(ROOTSCSIHDRS) \
1230 $(ROOTSCSIADHDRS) \
1231 $(ROOTSCSICONFHDRS) \
1232 $(ROOTSCSIIISCSIIHDRS) \
1233 $(ROOTSCSIIIGENHDRS) \
1234 $(ROOTSCSIIIMPLHDRS) \
1235 $(ROOTSCSIIIVHCIHDRS) \
1236 $(ROOTSDCARDHDRS) \
1237 $(ROOTSYSEVENTHDRS) \
1238 $(ROOTCONTRACTHDRS) \
1239 $(ROOTUSBHDRS) \
1240 $(ROOTUWBHDRS) \
1241 $(ROOTUWBAHDRS) \
1242 $(ROOTUSBAUDHDRS) \
1243 $(ROOTUSBHUBDHDRS) \
1244 $(ROOTUSBHIDHDRS) \
1245 $(ROOTUSBHRCHDRS) \
1246 $(ROOTUSBMSHDRS) \
1247 $(ROOTUSBPRNHDRS) \

```

```

1248 $(ROOTUSBCDCHDRS) \
1249 $(ROOTUSBVIDHDRS) \
1250 $(ROOTUSBWCMHDRS) \
1251 $(ROOTUGENHDRS) \
1252 $(ROOTI1394HDRS) \
1253 $(ROOTHOTPLUGHDRS) \
1254 $(ROOTHOTPLUGPCIHDRS) \
1255 $(ROOTRSMHDRS) \
1256 $(ROOTTSOLHDRS) \
1257 $($ (MACH)_ROOTHDRS)

```

```

1260 install_h: \
1261 $(ROOTDIRS) \
1262 .WAIT \
1263 $(ROOTHDRS) \
1264 $(ROOTAUDHDRS) \
1265 $(ROOTAVHDRS) \
1266 $(ROOTCRYPTOHDRS) \
1267 $(ROOTDCAMHDRS) \
1268 $(ROOTISOHDRS) \
1269 $(ROOTIDMHDRS) \
1270 $(ROOTISCSIHDRS) \
1271 $(ROOTISCSITHDRS) \
1272 $(ROOTFC4HDRS) \
1273 $(ROOTFCHDRS) \
1274 $(ROOTFCIMPLHDRS) \
1275 $(ROOTFCULPHDRS) \
1276 $(ROOTFMHDRS) \
1277 $(ROOTFMFSDHDRS) \
1278 $(ROOTFMIOHDRS) \
1279 $(ROOTFSDHDRS) \
1280 $(ROOTIBDHDRS) \
1281 $(ROOTIBHDRS) \
1282 $(ROOTIBTLHDRS) \
1283 $(ROOTIBTLIMPLHDRS) \
1284 $(ROOTIBNEXHDRS) \
1285 $(ROOTIBMGTHDRS) \
1286 $(ROOTIBMFHDRS) \
1287 $(ROOTOFHDRS) \
1288 $(ROOTRDMAHDRS) \
1289 $(ROOTSOL_OFSDHDRS) \
1290 $(ROOTSOL_UMADHDRS) \
1291 $(ROOTSOL_UVERBSHDRS) \
1292 $(ROOTSOL_UCMAHDRS) \
1293 $(ROOTTAVORHDRS) \
1294 $(ROOTTHERMONHDRS) \
1295 $(ROOTMLNXHDRS) \
1296 $(ROOTSCSIHDRS) \
1297 $(ROOTSCSIADHDRS) \
1298 $(ROOTSCSIIISCSIIHDRS) \
1299 $(ROOTSCSIIIGENHDRS) \
1300 $(ROOTSCSIIIMPLHDRS) \
1301 $(ROOTSCSIIIVHCIHDRS) \
1302 $(ROOTSDCARDHDRS) \
1303 $(ROOTSYSEVENTHDRS) \
1304 $(ROOTCONTRACTHDRS) \
1305 $(ROOTUWBHDRS) \
1306 $(ROOTUWBAHDRS) \
1307 $(ROOTUSBAUDHDRS) \
1308 $(ROOTUSBHUBDHDRS) \
1309 $(ROOTUSBHIDHDRS) \
1310 $(ROOTUSBHRCHDRS) \
1311 $(ROOTUSBMSHDRS) \
1312 $(ROOTUSBPRNHDRS) \
1313

```



```
1314      $(ROOTUSBPRNHDRS)      \  
1315      $(ROOTUSBCDCHDRS)      \  
1316      $(ROOTUSBVIDHDRS)      \  
1317      $(ROOTUSBWCMHDRS)      \  
1318      $(ROOTUGENHDRS)        \  
1319      $(ROOTI1394HDRS)       \  
1320      $(ROOTHOTPLUGHDRS)     \  
1321      $(ROOTHOTPLUGPCIHDRS)  \  
1322      $(ROOTRSMHDRS)         \  
1323      $(ROOTTSOLHDRS)        \  
1324      $($MACH)_ROOTHDRS      \  
  
1326 all_h: $(GENHDRS)  
  
1328 priv_const.h: $(PRIVS_AWK) $(PRIVS_DEF)  
1329      $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v privhfile=$@  
  
1331 priv_names.h: $(PRIVS_AWK) $(PRIVS_DEF)  
1332      $(AWK) -f $(PRIVS_AWK) < $(PRIVS_DEF) -v pubhfile=$@  
  
1334 usb/usbdevs.h: $(USBDEVS_AWK) $(USBDEVS_DATA)  
1335      $(AWK) -f $(USBDEVS_AWK) $(USBDEVS_DATA) -H > $@  
  
1337 clean:  
1338      $(RM) $(GENHDRS)  
  
1340 clobber: clean  
  
1342 check: $(CHECKHDRS)  
  
1344 FRC:
```

```

*****
2270 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/sys/cpu_uarray.h
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 /*
2  * This file and its contents are supplied under the terms of the
3  * Common Development and Distribution License ("CDDL"), version 1.0.
4  * You may only use this file in accordance with the terms of version
5  * 1.0 of the CDDL.
6  *
7  * A full copy of the text of the CDDL should have accompanied this
8  * source. A copy of the CDDL is also available via the Internet at
9  * http://www.illumos.org/license/CDDL.
10 */
12 /*
13  * Copyright (c) 2018, Joyent, Inc.
14 */
16 /*
17  * Use a cpu_uarray_t for an array of uint64_t values that are written on a
18  * per-CPU basis. We align each CPU on a 128-byte boundary (so two cachelines).
19  * It's not clear why, but this can have a significant effect in multi-socket
20  * systems running certain benchmarks on a relatively current Intel system.
21  *
22  * So the layout is like this, for example:
23  *
24  * 0:   STAT1 for CPU 0
25  * 8:   STAT2 for CPU 0
26  * 16:  STAT3 for CPU 0
27  * 24:  padding
28  * 128: STAT1 for CPU 1
29  * 136: STAT2 for CPU 1
30  * ...
31  *
32  * At collection time, cpu_uarray_sum() can be used to sum the given value index
33  * across all CPUs, or cpu_uarray_sum_all() sums all stats across all CPUs.
34  * The summation is done such that it saturates at UIN64_MAX.
35  */
37 #ifndef _SYS_CPU_UARRAY_H
38 #define _SYS_CPU_UARRAY_H
40 #include <sys/types.h>
42 #ifdef __cplusplus
43 extern "C" {
44 #endif
46 #ifdef _KERNEL
48 /*
49  * Trying to include sysmacros.h for P2ROUNDUP() here is just too painful.
50  */
51 #define CUA_ROUNDUP(x, align) (-(x) & -(align))
52 #define CUA_ALIGN (128)
53 #define CUA_CPU_STRIDE(nr_items) \
54     CUA_ROUNDUP((nr_items), CUA_ALIGN / sizeof (uint64_t))
55 #define CUA_INDEX(nr_items, c, i) (((c) * CUA_CPU_STRIDE(nr_items)) + (i))
57 #define CPU_UARRAY_VAL(cua, cpu_index, stat_index) \
58     ((cua)->cu_vals[CUA_INDEX((cua)->cu_nr_items, cpu_index, stat_index)])
60 typedef struct {

```

```

61     uint64_t cu_nr_items;
62     char cu_pad[CUA_ALIGN - sizeof (uint64_t)];
63 #ifdef __lint
64     volatile uint64_t cu_vals[1];
65 #else
66     volatile uint64_t cu_vals[];
67 #endif
68 } cpu_uarray_t __aligned(CUA_ALIGN);
70 extern cpu_uarray_t *cpu_uarray_zalloc(size_t, int);
71 extern void cpu_uarray_free(cpu_uarray_t *);
72 extern uint64_t cpu_uarray_sum(cpu_uarray_t *, size_t);
73 extern uint64_t cpu_uarray_sum_all(cpu_uarray_t *);
75 #endif /* _KERNEL */
77 #ifdef __cplusplus
78 }
79 #endif
81 #endif /* _SYS_CPU_UARRAY_H */

```

```

*****
12367 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/sys/sysmacros.h
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*      Copyright (c) 1984, 1986, 1987, 1988, 1989 AT&T */
22 /*      All Rights Reserved      */

25 /*
26 * Copyright 2008 Sun Microsystems, Inc.  All rights reserved.
27 * Use is subject to license terms.
28 *
29 * Copyright 2013 Nexenta Systems, Inc.  All rights reserved.
30 *
31 * Copyright 2018 Joyent Inc.
32 */

34 #ifndef _SYS_SYSMACROS_H
35 #define _SYS_SYSMACROS_H

37 #include <sys/param.h>
38 #include <sys/stddef.h>

40 #ifdef __cplusplus
41 extern "C" {
42 #endif

44 /*
45 * Some macros for units conversion
46 */
47 /*
48 * Disk blocks (sectors) and bytes.
49 */
50 #define dtob(DD)      ((DD) << DEV_BSHIFT)
51 #define btod(BB)     (((BB) + DEV_BSIZE - 1) >> DEV_BSHIFT)
52 #define btodt(BB)    ((BB) >> DEV_BSHIFT)
53 #define lbtod(BB)    (((offset_t)(BB) + DEV_BSIZE - 1) >> DEV_BSHIFT)

55 /* common macros */
56 #ifndef MIN
57 #define MIN(a, b)     ((a) < (b) ? (a) : (b))
58 #endif
59 #ifndef MAX
60 #define MAX(a, b)     ((a) < (b) ? (b) : (a))

```

```

61 #endif
62 #ifndef ABS
63 #define ABS(a)        ((a) < 0 ? -(a) : (a))
64 #endif
65 #ifndef SIGNOF
66 #define SIGNOF(a)     ((a) < 0 ? -1 : (a) > 0)
67 #endif

69 #ifdef _KERNEL

71 /*
72 * Convert a single byte to/from binary-coded decimal (BCD).
73 */
74 extern unsigned char byte_to_bcd[256];
75 extern unsigned char bcd_to_byte[256];

77 #define BYTE_TO_BCD(x)  byte_to_bcd[(x) & 0xff]
78 #define BCD_TO_BYTE(x) bcd_to_byte[(x) & 0xff]

80 #endif /* _KERNEL */

82 /*
83 * WARNING: The device number macros defined here should not be used by device
84 * drivers or user software. Device drivers should use the device functions
85 * defined in the DDI/DKI interface (see also ddi.h). Application software
86 * should make use of the library routines available in makedev(3). A set of
87 * new device macros are provided to operate on the expanded device number
88 * format supported in SVR4. Macro versions of the DDI device functions are
89 * provided for use by kernel proper routines only. Macro routines bmajor(),
90 * major(), minor(), emajor(), emminor(), and makedev() will be removed or
91 * their definitions changed at the next major release following SVR4.
92 */

94 #define O_BITSMAJOR    7      /* # of SVR3 major device bits */
95 #define O_BITSMINOR   8      /* # of SVR3 minor device bits */
96 #define O_MAXMAJ      0x7f   /* SVR3 max major value */
97 #define O_MAXMIN      0xff   /* SVR3 max minor value */

100 #define L_BITSMAJOR32 14     /* # of SVR4 major device bits */
101 #define L_BITSMINOR32 18    /* # of SVR4 minor device bits */
102 #define L_MAXMAJ32    0x3fff /* SVR4 max major value */
103 #define L_MAXMIN32    0x3ffff /* MAX minor for 3b2 software drivers. */
104 /* For 3b2 hardware devices the minor is */
105 /* restricted to 256 (0-255) */

107 #ifdef _LP64
108 #define L_BITSMAJOR    32     /* # of major device bits in 64-bit Solaris */
109 #define L_BITSMINOR   32     /* # of minor device bits in 64-bit Solaris */
110 #define L_MAXMAJ      0xffffffff /* max major value */
111 #define L_MAXMIN      0xffffffff /* max minor value */
112 #else
113 #define L_BITSMAJOR    L_BITSMAJOR32
114 #define L_BITSMINOR   L_BITSMINOR32
115 #define L_MAXMAJ      L_MAXMAJ32
116 #define L_MAXMIN      L_MAXMIN32
117 #endif

119 #ifdef _KERNEL

121 /* major part of a device internal to the kernel */

123 #define major(x)      (major_t)((((unsigned)(x)) >> O_BITSMINOR) & O_MAXMAJ)
124 #define bmajor(x)    (major_t)((((unsigned)(x)) >> O_BITSMINOR) & O_MAXMAJ)

126 /* get internal major part of expanded device number */

```

```

128 #define getmajor(x)      (major_t)((((dev_t)(x)) >> L_BITSMINOR) & L_MAXMAJ)
130 /* minor part of a device internal to the kernel */
132 #define minor(x)        (minor_t)((x) & O_MAXMIN)
134 /* get internal minor part of expanded device number */
136 #define getminor(x)      (minor_t)((x) & L_MAXMIN)
138 #else /* _KERNEL */
140 /* major part of a device external from the kernel (same as emajor below) */
142 #define major(x)         (major_t)((((unsigned)(x)) >> O_BITSMINOR) & O_MAXMAJ)
144 /* minor part of a device external from the kernel (same as eminor below) */
146 #define minor(x)        (minor_t)((x) & O_MAXMIN)
148 #endif /* _KERNEL */
150 /* create old device number */
152 #define makedev(x, y) (unsigned short)((((x) << O_BITSMINOR) | ((y) & O_MAXMIN)))
154 /* make an new device number */
156 #define makedevice(x, y) (dev_t)((((dev_t)(x) << L_BITSMINOR) | ((y) & L_MAXMIN)))
159 /*
160 * emajor() allows kernel/driver code to print external major numbers
161 * eminor() allows kernel/driver code to print external minor numbers
162 */
164 #define emajor(x) \
165     (major_t)((((unsigned int)(x) >> O_BITSMINOR) > O_MAXMAJ) ? \
166     NODEV : (((unsigned int)(x) >> O_BITSMINOR) & O_MAXMAJ)
168 #define eminor(x) \
169     (minor_t)((x) & O_MAXMIN)
171 /*
172 * get external major and minor device
173 * components from expanded device number
174 */
175 #define getemajor(x)      (major_t)((((dev_t)(x) >> L_BITSMINOR) > L_MAXMAJ) ? \
176     NODEV : (((dev_t)(x) >> L_BITSMINOR) & L_MAXMAJ))
177 #define geteminor(x)      (minor_t)((x) & L_MAXMIN)
179 /*
180 * These are versions of the kernel routines for compressing and
181 * expanding long device numbers that don't return errors.
182 */
183 #if (L_BITSMAJOR32 == L_BITSMAJOR) && (L_BITSMINOR32 == L_BITSMINOR)
185 #define DEVCML(x)        (x)
186 #define DEVEXPL(x)      (x)
188 #else
190 #define DEVCML(x)        \
191     (dev32_t)((((x) >> L_BITSMINOR) > L_MAXMAJ32 || \
192     ((x) & L_MAXMIN) > L_MAXMIN32) ? NODEV32 : \

```

```

193     (((x) >> L_BITSMINOR) << L_BITSMINOR32) | ((x) & L_MAXMIN32)))
195 #define DEVEXPL(x)      \
196     (((x) == NODEV32) ? NODEV : \
197     makedevice(((x) >> L_BITSMINOR32) & L_MAXMAJ32, (x) & L_MAXMIN32))
199 #endif /* L_BITSMAJOR32 ... */
201 /* convert to old (SVR3.2) dev format */
203 #define cmpdev(x) \
204     (o_dev_t)((((x) >> L_BITSMINOR) > O_MAXMAJ || \
205     ((x) & L_MAXMIN) > O_MAXMIN) ? NODEV : \
206     (((x) >> L_BITSMINOR) << O_BITSMINOR) | ((x) & O_MAXMIN))
208 /* convert to new (SVR4) dev format */
210 #define expdev(x) \
211     (dev_t)((((dev_t)((x) >> O_BITSMINOR) & O_MAXMAJ) << L_BITSMINOR) | \
212     ((x) & O_MAXMIN))
214 /*
215 * Macro for checking power of 2 address alignment.
216 */
217 #define IS_P2ALIGNED(v, a) (((uintptr_t)(v)) & ((uintptr_t)(a) - 1)) == 0)
219 /*
220 * Macros for counting and rounding.
221 */
222 #define howmany(x, y)      (((x)+(y)-1)/(y))
223 #define roundup(x, y)     (((x)+(y)-1)/(y))*y)
225 /*
226 * Macro to determine if value is a power of 2
227 */
228 #define ISP2(x)           (((x) & ((x) - 1)) == 0)
230 /*
231 * Macros for various sorts of alignment and rounding. The "align" must
232 * be a power of 2. Often times it is a block, sector, or page.
233 */
235 /*
236 * return x rounded down to an align boundary
237 * eg, P2ALIGN(1200, 1024) == 1024 (1*align)
238 * eg, P2ALIGN(1024, 1024) == 1024 (1*align)
239 * eg, P2ALIGN(0x1234, 0x100) == 0x1200 (0x12*align)
240 * eg, P2ALIGN(0x5600, 0x100) == 0x5600 (0x56*align)
241 */
242 #define P2ALIGN(x, align)      ((x) & ~(align))
244 /*
245 * return x % (mod) align
246 * eg, P2PHASE(0x1234, 0x100) == 0x34 (x-0x12*align)
247 * eg, P2PHASE(0x5600, 0x100) == 0x00 (x-0x56*align)
248 */
249 #define P2PHASE(x, align)      ((x) & ((align) - 1))
251 /*
252 * return how much space is left in this block (but if it's perfectly
253 * aligned, return 0).
254 * eg, P2NPHASE(0x1234, 0x100) == 0xcc (0x13*align-x)
255 * eg, P2NPHASE(0x5600, 0x100) == 0x00 (0x56*align-x)
256 */
257 #define P2NPHASE(x, align)      (-(x) & ((align) - 1))

```

```

259 /*
260 * return x rounded up to an align boundary
261 * eg, P2ROUNDUP(0x1234, 0x100) == 0x1300 (0x13*align)
262 * eg, P2ROUNDUP(0x5600, 0x100) == 0x5600 (0x56*align)
263 */
264 #define P2ROUNDUP(x, align)      (-(~(x) & ~(align)))

266 /*
267 * return the ending address of the block that x is in
268 * eg, P2END(0x1234, 0x100) == 0x12ff (0x13*align - 1)
269 * eg, P2END(0x5600, 0x100) == 0x56ff (0x57*align - 1)
270 */
271 #define P2END(x, align)         (-(~(x) & ~(align)))

273 /*
274 * return x rounded up to the next phase (offset) within align.
275 * phase should be < align.
276 * eg, P2PHASEUP(0x1234, 0x100, 0x10) == 0x1310 (0x13*align + phase)
277 * eg, P2PHASEUP(0x5600, 0x100, 0x10) == 0x5610 (0x56*align + phase)
278 */
279 #define P2PHASEUP(x, align, phase)  ((phase) - (((phase) - (x)) & ~(align)))

281 /*
282 * return TRUE if adding len to off would cause it to cross an align
283 * boundary.
284 * eg, P2BOUNDARY(0x1234, 0xe0, 0x100) == TRUE (0x1234 + 0xe0 == 0x1314)
285 * eg, P2BOUNDARY(0x1234, 0x50, 0x100) == FALSE (0x1234 + 0x50 == 0x1284)
286 */
287 #define P2BOUNDARY(off, len, align) \
288     (((off) ^ ((off) + (len) - 1)) > (align) - 1)

290 /*
291 * Return TRUE if they have the same highest bit set.
292 * eg, P2SAMEHIGHBIT(0x1234, 0x1001) == TRUE (the high bit is 0x1000)
293 * eg, P2SAMEHIGHBIT(0x1234, 0x3010) == FALSE (high bit of 0x3010 is 0x2000)
294 */
295 #define P2SAMEHIGHBIT(x, y)      (((x) ^ (y)) < ((x) & (y)))

297 /*
298 * Typed version of the P2* macros. These macros should be used to ensure
299 * that the result is correctly calculated based on the data type of (x),
300 * which is passed in as the last argument, regardless of the data
301 * type of the alignment. For example, if (x) is of type uint64_t,
302 * and we want to round it up to a page boundary using "PAGESIZE" as
303 * the alignment, we can do either
304 *     P2ROUNDUP(x, (uint64_t)PAGESIZE)
305 * or
306 *     P2ROUNDUP_TYPED(x, PAGESIZE, uint64_t)
307 */
308 #define P2ALIGN_TYPED(x, align, type) \
309     ((type)(x) & ~(type)(align))
310 #define P2PHASE_TYPED(x, align, type) \
311     ((type)(x) & ((type)(align) - 1))
312 #define P2NPHASE_TYPED(x, align, type) \
313     (~(type)(x) & ((type)(align) - 1))
314 #define P2ROUNDUP_TYPED(x, align, type) \
315     (-(~(type)(x) & ~(type)(align)))
316 #define P2END_TYPED(x, align, type) \
317     (~(type)(x) & ~(type)(align))
318 #define P2PHASEUP_TYPED(x, align, phase, type) \
319     ((type)(phase) - (((type)(phase) - (type)(x)) & ~(type)(align)))
320 #define P2CROSS_TYPED(x, y, align, type) \
321     (((type)(x) ^ (type)(y)) > (type)(align) - 1)
322 #define P2SAMEHIGHBIT_TYPED(x, y, type) \
323     (((type)(x) ^ (type)(y)) < ((type)(x) & (type)(y)))

```

```

325 /*
326 * Macros to atomically increment/decrement a variable. mutex and var
327 * must be pointers.
328 */
329 #define INCR_COUNT(var, mutex) mutex_enter(mutex), (*(var))++, mutex_exit(mutex)
330 #define DECR_COUNT(var, mutex) mutex_enter(mutex), (*(var))--, mutex_exit(mutex)

332 /*
333 * Macros to declare bitfields - the order in the parameter list is
334 * Low to High - that is, declare bit 0 first. We only support 8-bit bitfields
335 * because if a field crosses a byte boundary it's not likely to be meaningful
336 * without reassembly in its nonnative endianness.
337 */
338 #if defined(_BIT_FIELDS_LTOH)
339 #define DECL_BITFIELD2(_a, _b) \
340     uint8_t _a, _b \
341 #define DECL_BITFIELD3(_a, _b, _c) \
342     uint8_t _a, _b, _c \
343 #define DECL_BITFIELD4(_a, _b, _c, _d) \
344     uint8_t _a, _b, _c, _d \
345 #define DECL_BITFIELD5(_a, _b, _c, _d, _e) \
346     uint8_t _a, _b, _c, _d, _e \
347 #define DECL_BITFIELD6(_a, _b, _c, _d, _e, _f) \
348     uint8_t _a, _b, _c, _d, _e, _f \
349 #define DECL_BITFIELD7(_a, _b, _c, _d, _e, _f, _g) \
350     uint8_t _a, _b, _c, _d, _e, _f, _g \
351 #define DECL_BITFIELD8(_a, _b, _c, _d, _e, _f, _g, _h) \
352     uint8_t _a, _b, _c, _d, _e, _f, _g, _h
353 #elif defined(_BIT_FIELDS_HTOH)
354 #define DECL_BITFIELD2(_a, _b) \
355     uint8_t _b, _a \
356 #define DECL_BITFIELD3(_a, _b, _c) \
357     uint8_t _c, _b, _a \
358 #define DECL_BITFIELD4(_a, _b, _c, _d) \
359     uint8_t _d, _c, _b, _a \
360 #define DECL_BITFIELD5(_a, _b, _c, _d, _e) \
361     uint8_t _e, _d, _c, _b, _a \
362 #define DECL_BITFIELD6(_a, _b, _c, _d, _e, _f) \
363     uint8_t _f, _e, _d, _c, _b, _a \
364 #define DECL_BITFIELD7(_a, _b, _c, _d, _e, _f, _g) \
365     uint8_t _g, _f, _e, _d, _c, _b, _a \
366 #define DECL_BITFIELD8(_a, _b, _c, _d, _e, _f, _g, _h) \
367     uint8_t _h, _g, _f, _e, _d, _c, _b, _a
368 #else
369 #error One of _BIT_FIELDS_LTOH or _BIT_FIELDS_HTOH must be defined
370 #endif /* _BIT_FIELDS_LTOH */

372 #if !defined(ARRAY_SIZE)
373 #define ARRAY_SIZE(x) (sizeof (x) / sizeof (x[0]))
374 #endif

376 /*
377 * Add a value to a uint64_t that saturates at UINT64_MAX instead of wrapping
378 * around.
379 */
380 #define UINT64_OVERFLOW_ADD(val, add) \
381     ((val) > ((val) + (add)) ? (UINT64_MAX) : ((val) + (add)))

383 /*
384 * Convert to an int64, saturating at INT64_MAX.
385 */
386 #define UINT64_OVERFLOW_TO_INT64(uval) \
387     (((uval) > INT64_MAX) ? INT64_MAX : (int64_t)(uval))

389 #ifdef __cplusplus
390 }
    unchanged_portion_omitted

```

```

*****
26899 Tue Nov 20 12:08:35 2018
new/usr/src/uts/common/sys/zone.h
9936 atomic ops in syscall_mstate() induce significant overhead
9942 zone secflags are not initialized correctly
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22 * Copyright (c) 2003, 2010, Oracle and/or its affiliates. All rights reserved.
23 * Copyright 2018 Joyent, Inc.
24 * Copyright 2015 Joyent, Inc. All rights reserved.
25 * Copyright 2014 Nexenta Systems, Inc. All rights reserved.
26 * Copyright 2014 Igor Kozhukhov <ikozhukhov@gmail.com>.
27 */

28 #ifndef _SYS_ZONE_H
29 #define _SYS_ZONE_H

30
31 #include <sys/types.h>
32 #include <sys/mutex.h>
33 #include <sys/param.h>
34 #include <sys/rctl.h>
35 #include <sys/ipc_rctl.h>
36 #include <sys/pset.h>
37 #include <sys/tsol/label.h>
38 #include <sys/cred.h>
39 #include <sys/netstack.h>
40 #include <sys/uadmin.h>
41 #include <sys/ksynch.h>
42 #include <sys/socket_impl.h>
43 #include <sys/secflags.h>
44 #include <netinet/in.h>
45 #include <sys/cpu_uarray.h>

46
47 #ifdef __cplusplus
48 extern "C" {
49 #endif

50
51 /*
52  * NOTE
53  *
54  * The contents of this file are private to the implementation of
55  * Solaris and are subject to change at any time without notice.
56  * Applications and drivers using these interfaces may fail to
57  * run on future releases.
58  */

```

```

60 /* Available both in kernel and for user space */

62 /* zone id restrictions and special ids */
63 #define MAX_ZONEID 9999
64 #define MIN_USERZONEID 1 /* lowest user-creatable zone ID */
65 #define MIN_ZONEID 0 /* minimum zone ID on system */
66 #define GLOBAL_ZONEID 0
67 #define ZONEID_WIDTH 4 /* for printf */

69 /*
70  * Special zoneid_t token to refer to all zones.
71  */
72 #define ALL_ZONES (-1)

74 /* system call subcodes */
75 #define ZONE_CREATE 0
76 #define ZONE_DESTROY 1
77 #define ZONE_GETATTR 2
78 #define ZONE_ENTER 3
79 #define ZONE_LIST 4
80 #define ZONE_SHUTDOWN 5
81 #define ZONE_LOOKUP 6
82 #define ZONE_BOOT 7
83 #define ZONE_VERSION 8
84 #define ZONE_SETATTR 9
85 #define ZONE_ADD_DATALINK 10
86 #define ZONE_DEL_DATALINK 11
87 #define ZONE_CHECK_DATALINK 12
88 #define ZONE_LIST_DATALINK 13

90 /* zone attributes */
91 #define ZONE_ATTR_ROOT 1
92 #define ZONE_ATTR_NAME 2
93 #define ZONE_ATTR_STATUS 3
94 #define ZONE_ATTR_PRIVSET 4
95 #define ZONE_ATTR_UNIQID 5
96 #define ZONE_ATTR_POOLID 6
97 #define ZONE_ATTR_INITPID 7
98 #define ZONE_ATTR_SLBL 8
99 #define ZONE_ATTR_INITNAME 9
100 #define ZONE_ATTR_BOOTARGS 10
101 #define ZONE_ATTR_BRAND 11
102 #define ZONE_ATTR_PHYS_MCAP 12
103 #define ZONE_ATTR_SCHED_CLASS 13
104 #define ZONE_ATTR_FLAGS 14
105 #define ZONE_ATTR_HOSTID 15
106 #define ZONE_ATTR_FS_ALLOWED 16
107 #define ZONE_ATTR_NETWORK 17
108 #define ZONE_ATTR_INITNORESTART 20
109 #define ZONE_ATTR_SECFLAGS 21

111 /* Start of the brand-specific attribute namespace */
112 #define ZONE_ATTR_BRAND_ATTRS 32768

114 #define ZONE_FS_ALLOWED_MAX 1024

116 #define ZONE_EVENT_CHANNEL "com.sun:zones:status"
117 #define ZONE_EVENT_STATUS_CLASS "status"
118 #define ZONE_EVENT_STATUS_SUBCLASS "change"

120 #define ZONE_EVENT_UNINITIALIZED "uninitialized"
121 #define ZONE_EVENT_INITIALIZED "initialized"
122 #define ZONE_EVENT_READY "ready"
123 #define ZONE_EVENT_RUNNING "running"
124 #define ZONE_EVENT_SHUTTING_DOWN "shutting_down"

```

```

126 #define ZONE_CB_NAME          "zonename"
127 #define ZONE_CB_NEWSTATE     "newstate"
128 #define ZONE_CB_OLDSTATE     "oldstate"
129 #define ZONE_CB_TIMESTAMP    "when"
130 #define ZONE_CB_ZONEID       "zoneid"

132 /*
133  * Exit values that may be returned by scripts or programs invoked by various
134  * zone commands.
135  *
136  * These are defined as:
137  *
138  *     ZONE_SUBPROC_OK
139  *     =====
140  *     The subprocess completed successfully.
141  *
142  *     ZONE_SUBPROC_USAGE
143  *     =====
144  *     The subprocess failed with a usage message, or a usage message should
145  *     be output in its behalf.
146  *
147  *     ZONE_SUBPROC_NOTCOMPLETE
148  *     =====
149  *     The subprocess did not complete, but the actions performed by the
150  *     subprocess require no recovery actions by the user.
151  *
152  *     For example, if the subprocess were called by "zoneadm install," the
153  *     installation of the zone did not succeed but the user need not perform
154  *     a "zoneadm uninstall" before attempting another install.
155  *
156  *     ZONE_SUBPROC_FATAL
157  *     =====
158  *     The subprocess failed in a fatal manner, usually one that will require
159  *     some type of recovery action by the user.
160  *
161  *     For example, if the subprocess were called by "zoneadm install," the
162  *     installation of the zone did not succeed and the user will need to
163  *     perform a "zoneadm uninstall" before another install attempt is
164  *     possible.
165  *
166  *     The non-success exit values are large to avoid accidental collision
167  *     with values used internally by some commands (e.g. "Z_ERR" and
168  *     "Z_USAGE" as used by zoneadm.)
169  */
170 #define ZONE_SUBPROC_OK          0
171 #define ZONE_SUBPROC_USAGE      253
172 #define ZONE_SUBPROC_NOTCOMPLETE 254
173 #define ZONE_SUBPROC_FATAL      255

175 #ifdef _SYSCALL32
176 typedef struct {
177     caddr32_t zone_name;
178     caddr32_t zone_root;
179     caddr32_t zone_privs;
180     size32_t zone_privssz;
181     caddr32_t rctlbuf;
182     size32_t rctlbufsz;
183     caddr32_t extended_error;
184     caddr32_t zfsbuf;
185     size32_t zfsbufsz;
186     int match; /* match level */
187     uint32_t doi; /* DOI for label */
188     caddr32_t label; /* label associated with zone */
189     int flags;
190 } zone_def32;
    unchanged portion omitted

```

```

323 #ifdef _KERNEL

325 /*
326  * We need to protect the definition of 'list_t' from userland applications and
327  * libraries which may be defining their own versions.
328  */
329 #include <sys/list.h>
330 #include <sys/loadavg.h>

332 #define GLOBAL_ZONEUNIQID      0 /* uniqid of the global zone */

334 /*
335  * Indexes into ->zone_ustate array, summing the micro state of all threads in a
336  * particular zone.
337  */
338 #define ZONE_USTATE_STIME (0)
339 #define ZONE_USTATE_UTIME (1)
340 #define ZONE_USTATE_WTIME (2)
341 #define ZONE_USTATE_MAX (3)

343 struct pool;
344 struct brand;

346 /*
347  * Each of these constants identifies a kernel subsystem that acquires and
348  * releases zone references. Each subsystem that invokes
349  * zone_hold_ref() and zone_rele_ref() should specify the
350  * zone_ref_subsys_t constant associated with the subsystem. Tracked holds
351  * help users and developers quickly identify subsystems that stall zone
352  * shutdowns indefinitely.
353  *
354  * NOTE: You must modify zone_ref_subsys_names in usr/src/uts/common/os/zone.c
355  * when you modify this enumeration.
356  */
357 typedef enum zone_ref_subsys {
358     ZONE_REF_NFS, /* NFS */
359     ZONE_REF_NFSV4, /* NFSv4 */
360     ZONE_REF_SMBFS, /* SMBFS */
361     ZONE_REF_MNTFS, /* MNTFS */
362     ZONE_REF_LOFI, /* LOFI devices */
363     ZONE_REF_VFS, /* VFS infrastructure */
364     ZONE_REF_IPC, /* IPC infrastructure */
365     ZONE_REF_NUM_SUBSYS /* This must be the last entry. */
366 } zone_ref_subsys_t;
    unchanged portion omitted

423 typedef struct zone {
424     /*
425      * zone_name is never modified once set.
426      */
427     char *zone_name; /* zone's configuration name */
428     /*
429      * zone_nodename and zone_domain are never freed once allocated.
430      */
431     char *zone_nodename; /* utsname.nodename equivalent */
432     char *zone_domain; /* srpc_domain equivalent */
433     /*
434      * zone_hostid is used for per-zone hostid emulation.
435      * Currently it isn't modified after it's set (so no locks protect
436      * accesses), but that might have to change when we allow
437      * administrators to change running zones' properties.
438      */
439     /* The global zone's zone_hostid must always be HW_INVALID_HOSTID so
440      * that zone_get_hostid() will function correctly.

```

```

441      */
442      uint32_t      zone_hostid; /* zone's hostid, HW_INVALID_HOSTID */
443                  /* if not emulated */
444      /*
445      * zone_lock protects the following fields of a zone_t:
446      *   zone_ref
447      *   zone_cred_ref
448      *   zone_subsys_ref
449      *   zone_ref_list
450      *   zone_ntasks
451      *   zone_flags
452      *   zone_zsd
453      *   zone_pfexecd
454      */
455      kmutex_t      zone_lock;
456      /*
457      * zone_linkage is the zone's linkage into the active or
458      * death-row list. The field is protected by zonehash_lock.
459      */
460      list_node_t   zone_linkage;
461      zoneid_t      zone_id; /* ID of zone */
462      uint_t        zone_ref; /* count of zone_hold()s on zone */
463      uint_t        zone_cred_ref; /* count of zone_hold_cred()s on zone */
464      /*
465      * Fixed-sized array of subsystem-specific reference counts
466      * The sum of all of the counts must be less than or equal to zone_ref.
467      * The array is indexed by the counts' subsystems' zone_ref_subsys_t
468      * constants.
469      */
470      uint_t        zone_subsys_ref[ZONE_REF_NUM_SUBSYS];
471      list_t        zone_ref_list; /* list of zone_ref_t structs */
472      /*
473      * zone_rootvp and zone_rootpath can never be modified once set.
474      */
475      struct vnode  *zone_rootvp; /* zone's root vnode */
476      char          *zone_rootpath; /* Path to zone's root + '/' */
477      ushort_t      zone_flags; /* misc flags */
478      zone_status_t zone_status; /* protected by zone_status_lock */
479      uint_t        zone_ntasks; /* number of tasks executing in zone */
480      kmutex_t      zone_nlwps_lock; /* protects zone_nlwps, and *nlwps */
481                  /* counters in projects and tasks */
482                  /* that are within the zone */
483      rctl_qty_t    zone_nlwps; /* number of lwps in zone */
484      rctl_qty_t    zone_nlwps_ctl; /* protected by zone_rctl->rctl_lock */
485      rctl_qty_t    zone_shmmax; /* System V shared memory usage */
486      ipc_rqty_t    zone_ipc; /* System V IPC id resource usage */

488      uint_t        zone_rootpathlen; /* strlen(zone_rootpath) + 1 */
489      uint32_t      zone_shares; /* FSS shares allocated to zone */
490      rctl_set_t    *zone_rctl; /* zone-wide (zone.*) rctls */
491      kmutex_t      zone_mem_lock; /* protects zone_locked_mem and */
492                  /* kpd_locked_mem for all */
493                  /* projects in zone. */
494                  /* Also protects zone_max_swap */
495                  /* grab after p_lock, before rcs_lock */
496      rctl_qty_t    zone_locked_mem; /* bytes of locked memory in */
497                  /* zone */
498      rctl_qty_t    zone_locked_mem_ctl; /* Current locked memory */
499                  /* limit. Protected by */
500                  /* zone_rctl->rctl_lock */
501      rctl_qty_t    zone_max_swap; /* bytes of swap reserved by zone */
502      rctl_qty_t    zone_max_swap_ctl; /* current swap limit. */
503                  /* Protected by */
504                  /* zone_rctl->rctl_lock */
505      kmutex_t      zone_rctl_lock; /* protects zone_max_lofi */
506      rctl_qty_t    zone_max_lofi; /* lofi devs for zone */

```

```

507      rctl_qty_t    zone_max_lofi_ctl; /* current lofi limit. */
508                  /* Protected by */
509                  /* zone_rctl->rctl_lock */
510      list_t        zone_zsd; /* list of Zone-Specific Data values */
511      kcondvar_t    zone_cv; /* used to signal state changes */
512      struct proc   *zone_zsched; /* Dummy kernel "zsched" process */
513      pid_t         zone_proc_initpid; /* pid of "init" for this zone */
514      char          *zone_initname; /* fs path to 'init' */
515      int           zone_boot_err; /* for zone_boot() if boot fails */
516      char          *zone_bootargs; /* arguments passed via zone_boot() */
517      uint64_t      zone_phys_mcap; /* physical memory cap */
518      /*
519      * zone_kthreads is protected by zone_status_lock.
520      */
521      kthread_t     *zone_kthreads; /* kernel threads in zone */
522      struct priv_set *zone_privset; /* limit set for zone */
523      /*
524      * zone_vfslist is protected by vfs_list_lock().
525      */
526      struct vfs     *zone_vfslist; /* list of FS's mounted in zone */
527      uint64_t       zone_uniqid; /* unique zone generation number */
528      struct cred     *zone_kcred; /* kcred-like, zone-limited cred */
529      /*
530      * zone_pool is protected by pool_lock().
531      */
532      struct pool     *zone_pool; /* pool the zone is bound to */
533      hrtime_t       zone_pool_mod; /* last pool bind modification time */
534      /* zone_psetid is protected by cpu_lock */
535      psetid_t       zone_psetid; /* pset the zone is bound to */

537      time_t         zone_boot_time; /* Similar to boot_time */

539      /*
540      * The following two can be read without holding any locks. They are
541      * updated under cpu_lock.
542      */
543      int            zone_ncpus; /* zone's idea of ncpus */
544      int            zone_ncpus_online; /* zone's idea of ncpus_online */
545      /*
546      * List of ZFS datasets exported to this zone.
547      */
548      list_t         zone_datasets; /* list of datasets */

550      ts_label_t     *zone_slabel; /* zone sensitivity label */
551      int            zone_match; /* require label match for packets */
552      tsol_mlp_list_t zone_mlps; /* MLPs on zone-private addresses */

554      boolean_t     zone_restart_init; /* Restart init if it dies? */
555      struct brand   *zone_brand; /* zone's brand */
556      void           *zone_brand_data; /* store brand specific data */
557      id_t           zone_defaultcid; /* dflt scheduling class id */
558      kstat_t        *zone_swapresv_kstat;
559      kstat_t        *zone_lockedmem_kstat;
560      /*
561      * zone_dl_list is protected by zone_lock
562      */
563      list_t         zone_dl_list;
564      netstack_t     *zone_netstack;
565      struct cpucap   *zone_cpucap; /* CPU caps data */
566      /*
567      * Solaris Auditing per-zone audit context
568      */
569      struct au_kcontext *zone_audit_kctx;
570      /*
571      * For private use by mntfs.
572      */

```



```

573 struct mntelem *zone_mntfs_db;
574 krwlock_t zone_mntfs_db_lock;

576 struct klpd_reg *zone_pfexecd;

578 char *zone_fs_allowed;
579 rctl_qty_t zone_nprocs; /* number of processes in the zone */
580 rctl_qty_t zone_nprocs_ctl; /* current limit protected by */
581 /* zone_rctls->rcls_lock */
582 kstat_t *zone_nprocs_kstat;

584 kmutex_t zone_mcap_lock; /* protects mcap statistics */
585 kstat_t *zone_mcap_ksp;
586 zone_mcap_kstat_t *zone_mcap_stats;
587 uint64_t zone_pgpgin; /* pages paged in */
588 uint64_t zone_anonpgin; /* anon pages paged in */
589 uint64_t zone_execpgin; /* exec pages paged in */
590 uint64_t zone_fspgin; /* fs pages paged in */
591 uint64_t zone_anon_alloc_fail; /* cnt of anon alloc fails */

593 psecflags_t zone_secflags; /* default zone security-flags */

595 /*
596 * Misc. kstats and counters for zone cpu-usage aggregation.
597 * The zone_Xtime values are the sum of the micro-state accounting
598 * values for all threads that are running or have run in the zone.
599 * This is tracked in msacct.c as threads change state.
600 * The zone_stime is the sum of the LMS_SYSTEM times.
601 * The zone_utime is the sum of the LMS_USER times.
602 * The zone_wtime is the sum of the LMS_WAIT_CPU times.
603 * As with per-thread micro-state accounting values, these values are
604 * not scaled to nanosecs. The scaling is done by the
605 * zone_misc_kstat_update function when kstats are requested.
606 */
598 kmutex_t zone_misc_lock; /* protects misc statistics */
599 kstat_t *zone_misc_ksp;
600 zone_misc_kstat_t *zone_misc_stats;
601 /* Accumulated microstate for all threads in this zone. */
602 cpu_uarray_t *zone_ustate;
600 uint64_t zone_stime; /* total system time */
601 uint64_t zone_utime; /* total user time */
602 uint64_t zone_wtime; /* total time waiting in runq */
603 /* fork-fail kstat tracking */
604 uint32_t zone_ffcap; /* hit an rctl cap */
605 uint32_t zone_ffnopro; /* get proc/lwp error */
606 uint32_t zone_ffnomem; /* as_dup/memory error */
607 uint32_t zone_ffmisc; /* misc. other error */

609 uint32_t zone_nested_intp; /* nested interp. kstat */

611 struct loadavg_s zone_loadavg; /* loadavg for this zone */
612 uint64_t zone_hp_avenrun[3]; /* high-precision avenrun */
613 int zone_avenrun[3]; /* FSCALEd avg. run queue len */

615 /*
616 * FSS stats updated once per second by fss_decay_usage.
617 */
618 uint32_t zone_fss_gen; /* FSS generation cntr */
619 uint64_t zone_run_ticks; /* tot # of ticks running */

621 /*
622 * DTrace-private per-zone state
623 */
624 int zone_dtrace_getf; /* # of unprivileged getf(s) */

626 /*

```

```

627 * Synchronization primitives used to synchronize between mounts and
628 * zone creation/destruction.
629 */
630 int zone_mounts_in_progress;
631 kcondvar_t zone_mount_cv;
632 kmutex_t zone_mount_lock;
633 } zone_t;
unchanged_portion_omitted

```