

new/usr/src/pkg/manifests/driver-network-vioif.mf

1

1788 Thu Mar 28 23:16:18 2013

new/usr/src/pkg/manifests/driver-network-vioif.mf

3644 Add virtio-net support into the Illumos

Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com

Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com

Reviewed by: David Hoppner, 0xfee@gmail.com

```
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) 2013, Nexenta Systems. All rights reserved.
24 #
25 #
26 #
27 # The default for payload-bearing actions in this package is to appear in the
28 # global zone only. See the include file for greater detail, as well as
29 # information about overriding the defaults.
30 #
31 <include global_zone_only_component>
32 set name=pkg.fmri value=pkg:/driver/network/vioif@$(PKGVERS)
33 set name=pkg.description value="VirtIO network driver"
34 set name=pkg.summary value="VirtIO network driver"
35 set name=info.classification \
36     value=org.opensolaris.category.2008:Drivers/Networking
37 set name=variant.arch value=$(ARCH)
38 dir path=kernel group=sys
39 dir path=kernel/drv group=sys
40 dir path=kernel/drv/$(ARCH64) group=sys
41 driver name=vioif alias=pcilaf4,1 perms="* 0666 root sys"
42 file path=kernel/drv/$(ARCH64)/vioif group=sys
43 $(i386_ONLY)file path=kernel/drv/vioif group=sys
44 legacy pkg=SUNWvioif desc="VirtIO network driver" name="VirtIO network driver"
45 license cr_Sun license=cr_Sun
46 license lic_CDDL license=lic_CDDL
```

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

1

43166 Thu Mar 28 23:16:18 2013

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

3644 Add virtio-net support into the Illumos

Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com

Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com

Reviewed by: David Hoppner, 0xfee@gmail.com

```
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) 2012 Nexenta Systems, Inc. All rights reserved.
25 # Copyright (c) 2012 by Delphix. All rights reserved.
26 # Copyright (c) 2013 by Saso Kiselkov. All rights reserved.
27 #
28 #
29 #
30 # This Makefile defines all file modules for the directory uts/common
31 # and its children. These are the source files which may be considered
32 # common to all SunOS systems.
33 #
34 i386_CORE_OBJS += \
35     atomic.o          \
36     avintr.o         \
37     pic.o
38 #
39 sparc_CORE_OBJS +=
40 #
41 COMMON_CORE_OBJS += \
42     beep.o           \
43     bitset.o        \
44     bp_map.o        \
45     brand.o         \
46     cpucaps.o       \
47     cmt.o           \
48     cmt_policy.o    \
49     cpu.o           \
50     cpu_event.o     \
51     cpu_intr.o      \
52     cpu_pm.o        \
53     cpupart.o       \
54     cap_util.o      \
55     disp.o          \
56     group.o         \
57     kstat_fr.o      \
58     iscsiboot_prop.o \
```

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

2

```
59     lgrp.o           \
60     lgrp_topo.o     \
61     mmapobj.o       \
62     mutex.o         \
63     page_lock.o     \
64     page_retire.o   \
65     panic.o         \
66     param.o         \
67     pg.o            \
68     pghw.o          \
69     putnext.o       \
70     rctl_proc.o     \
71     rwlock.o        \
72     seg_kmem.o      \
73     softint.o       \
74     string.o        \
75     strtol.o        \
76     strtoul.o       \
77     strtoll.o       \
78     strtoull.o      \
79     thread_intr.o   \
80     vm_page.o       \
81     vm_pagelist.o   \
82     zlib_obj.o      \
83     clock_tick.o
84 #
85 CORE_OBJS += $(COMMON_CORE_OBJS) $(MACH)_CORE_OBJS
86 #
87 ZLIB_OBJS = zutil.o zmod.o zmod_subr.o \
88     adler32.o crc32.o deflate.o inffast.o \
89     inflate.o inftrees.o trees.o
90 #
91 GENUNIX_OBJS += \
92     access.o         \
93     acl.o            \
94     acl_common.o    \
95     adjtime.o        \
96     alarm.o          \
97     aio_subr.o       \
98     auditsys.o       \
99     audit_core.o     \
100    audit_zone.o     \
101    audit_memory.o   \
102    autoconf.o       \
103    avl.o             \
104    bdev_dsort.o     \
105    bio.o             \
106    bitmap.o         \
107    blabel.o         \
108    brandsys.o       \
109    bz2blocksort.o   \
110    bz2compress.o    \
111    bz2decompress.o  \
112    bz2randtable.o   \
113    bz2zlib.o        \
114    bz2crctable.o    \
115    bz2huffman.o     \
116    callb.o          \
117    callout.o        \
118    chdir.o          \
119    chmod.o          \
120    chown.o          \
121    cladm.o          \
122    class.o          \
123    clock.o          \
124    clock_highres.o \
```

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

```

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

```

3

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

```

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

```

4

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

```

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

```

5

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

```

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

```

6

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

7

```

389         vnode.o          \
390         vuid_queue.o     \
391         vuid_store.o    \
392         waitq.o         \
393         watchpoint.o   \
394         yield.o        \
395         scsi_confdata.o \
396         xattr.o        \
397         xattr_common.o \
398         xdr_mblk.o     \
399         xdr_mem.o      \
400         xdr.o          \
401         xdr_array.o   \
402         xdr_refer.o   \
403         xhat.o        \
404         zone.o

406 #
407 #     Stubs for the stand-alone linker/loader
408 #
409 sparc_GENSTUBS_OBJS = \
410     kobj_stubs.o

412 i386_GENSTUBS_OBJS =

414 COMMON_GENSTUBS_OBJS =

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

418 #
419 #     DTrace and DTrace Providers
420 #
421 DTRACE_OBJS += dtrace.o dtrace_isa.o dtrace_asm.o

423 SDT_OBJS += sdt_subr.o

425 PROFILE_OBJS += profile.o

427 SYSTRACE_OBJS += systrace.o

429 LOCKSTAT_OBJS += lockstat.o

431 FASTTRAP_OBJS += fasttrap.o fasttrap_isa.o

433 DCPC_OBJS += dcpc.o

435 #
436 #     Driver (pseudo-driver) Modules
437 #
438 IPP_OBJS += ippctl.o

440 AUDIO_OBJS += audio_client.o audio_ddi.o audio_engine.o \
441     audio_fltdata.o audio_format.o audio_ctrl.o \
442     audio_grc3.o audio_output.o audio_input.o \
443     audio_oss.o audio_sun.o

445 AUDIOEMU10K_OBJS += audioemu10k.o

447 AUDIOENS_OBJS += audioens.o

449 AUDIOVIA823X_OBJS += audiovia823x.o

451 AUDIOVIA97_OBJS += audiovia97.o

453 AUDIO1575_OBJS += audio1575.o

```

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

8

```

455 AUDIO810_OBJS += audio810.o

457 AUDIOCMI_OBJS += audiocmi.o

459 AUDIOCMIHD_OBJS += audiocmihd.o

461 AUDIOHD_OBJS += audiohd.o

463 AUDIOIXP_OBJS += audioixp.o

465 AUDIOLS_OBJS += audiols.o

467 AUDIOP16X_OBJS += audiop16x.o

469 AUDIOPCI_OBJS += audiopci.o

471 AUDIOSOLO_OBJS += audiosolo.o

473 AUDIOTS_OBJS += audiot.s.o

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

477 BLKDEV_OBJS += blkdev.o

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

481 CONSKBD_OBJS += conskbd.o

483 CONSMS_OBJS += consms.o

485 OLDPTY_OBJS += tty_ptyconf.o

487 PTC_OBJS += tty_pty.o

489 PTSL_OBJS += tty_pts.o

491 PTM_OBJS += ptm.o

493 MII_OBJS += mii.o mii_cicada.o mii_natsemi.o mii_intel.o mii_qualsemi.o \
494     mii_marvell.o mii_realtek.o mii_other.o

496 PTS_OBJS += pts.o

498 PTY_OBJS += ptms_conf.o

500 SAD_OBJS += sad.o

502 MD4_OBJS += md4.o md4_mod.o

504 MD5_OBJS += md5.o md5_mod.o

506 SHA1_OBJS += sha1.o sha1_mod.o

508 SHA2_OBJS += sha2.o sha2_mod.o

510 IPGPC_OBJS += classifierddi.o classifier.o filters.o trie.o table.o \
511     ba_table.o

513 DSCPMK_OBJS += dscpmk.o dscpmkddi.o

515 DLCOSMK_OBJS += dlcosmk.o dlcosmkddi.o

517 FLOWACCT_OBJS += flowacctddi.o flowacct.o

519 TOKENMT_OBJS += tokenmt.o tokenmtddi.o

```

```

521 TSWTCL_OBJS += tswtcl.o tswtclddi.o
523 ARP_OBJS += arpddi.o
525 ICMP_OBJS += icmpddi.o
527 ICMP6_OBJS += icmp6ddi.o
529 RTS_OBJS += rtsddi.o

531 IP_ICMP_OBJS = icmp.o icmp_opt_data.o
532 IP_RTS_OBJS = rts.o rts_opt_data.o
533 IP_TCP_OBJS = tcp.o tcp_fusion.o tcp_opt_data.o tcp_sack.o tcp_stats.o \
534 tcp_misc.o tcp_timers.o tcp_time_wait.o tcp_tpi.o tcp_output.o \
535 tcp_input.o tcp_socket.o tcp_bind.o tcp_cluster.o tcp_tunables.o
536 IP_UDP_OBJS = udp.o udp_opt_data.o udp_tunables.o udp_stats.o
537 IP_SCTP_OBJS = sctp.o sctp_opt_data.o sctp_output.o \
538 sctp_init.o sctp_input.o sctp_cookie.o \
539 sctp_conn.o sctp_error.o sctp_snmp.o \
540 sctp_tunables.o sctp_shutdown.o sctp_common.o \
541 sctp_timer.o sctp_heartbeat.o sctp_hash.o \
542 sctp_bind.o sctp_notify.o sctp_asconf.o \
543 sctp_addr.o tn_ipopt.o tnet.o ip_netinfo.o \
544 sctp_misc.o
545 IP_ILB_OBJS = ilb.o ilb_nat.o ilb_conn.o ilb_alg_hash.o ilb_alg_rr.o

547 IP_OBJS += igmp.o ipmp.o ip.o ip6.o ip6_asp.o ip6_if.o ip6_ire.o \
548 ip6_rts.o ip_if.o ip_ire.o ip_listutils.o ip_mrout.o \
549 ip_multi.o ip2mac.o ip_ndp.o ip_rts.o ip_srcid.o \
550 ipddi.o ipdrop.o mi.o nd.o tunables.o optcom.o snmpcom.o \
551 ipsec_loader.o spd.o ipclassifier.o inet_common.o ip_queue.o \
552 squeue.o ip_sadb.o ip_ftable.o proto_set.o radix.o ip_dummy.o \
553 ip_helper_stream.o ip_tunables.o \
554 ip_output.o ip_input.o ip6_input.o ip6_output.o ip_arp.o \
555 conn_opt.o ip_attr.o ip_dce.o \
556 $(IP_ICMP_OBJS) \
557 $(IP_RTS_OBJS) \
558 $(IP_TCP_OBJS) \
559 $(IP_UDP_OBJS) \
560 $(IP_SCTP_OBJS) \
561 $(IP_ILB_OBJS)

563 IP6_OBJS += ip6ddi.o
565 HOOK_OBJS += hook.o
567 NETI_OBJS += neti_impl.o neti_mod.o neti_stack.o
569 KEYSOCK_OBJS += keysockddi.o keysock.o keysock_opt_data.o
571 IPNET_OBJS += ipnet.o ipnet_bpf.o
573 SPDSOCK_OBJS += spdsockddi.o spdsock.o spdsock_opt_data.o
575 IPSECESP_OBJS += ipsecespddi.o ipsecesp.o
577 IPSECAH_OBJS += ipsecahddi.o ipsecah.o sadb.o
579 SPPP_OBJS += sPPP.o sPPP_dlpi.o sPPP_mod.o s_common.o
581 SPPPTUN_OBJS += sPPPtun.o sPPPtun_mod.o
583 SPPPASYN_OBJS += sPPPpasyn.o sPPPpasyn_mod.o
585 SPPPCOMP_OBJS += sPPPcomp.o sPPPcomp_mod.o deflate.o BSD-comp.o vjcompress.o \
586 zlib.o

```

```

588 TCP_OBJS += tcpddi.o
590 TCP6_OBJS += tcp6ddi.o
592 NCA_OBJS += ncaddi.o
594 SDP SOCK_MOD_OBJS += sockmod_sdp.o socksdp.o socksdpsubr.o
596 SCTP SOCK_MOD_OBJS += sockmod_sctp.o sockscctp.o socksctpsubr.o
598 PFP SOCK_MOD_OBJS += sockmod_pfp.o
600 RDS SOCK_MOD_OBJS += sockmod_rds.o
602 RDS_OBJS += rdsddi.o rdssubr.o rds_opt.o rds_ioctl.o
604 RDSIB_OBJS += rdsib.o rdsib_ib.o rdsib_cm.o rdsib_ep.o rdsib_buf.o \
605 rdsib_debug.o rdsib_sc.o
607 RDSV3_OBJS += af_rds.o rdsv3_ddi.o bind.o loop.o threads.o connection.o \
608 transport.o cong.o sysctl.o message.o rds_rcv.o send.o \
609 stats.o info.o page.o rdma_transport.o ib_ring.o ib_rdma.o \
610 ib_rcv.o ib.o ib_send.o ib_sysctl.o ib_stats.o ib_cm.o \
611 rdsv3_sc.o rdsv3_debug.o rdsv3_impl.o rdma.o rdsv3_af_thr.o

613 ISER_OBJS += iser.o iser_cm.o iser_cg.o iser_ib.o iser_idm.o \
614 iser_resource.o iser_xfer.o
616 UDP_OBJS += udpddi.o
618 UDP6_OBJS += udp6ddi.o
620 SY_OBJS += gentyty.o
622 TCO_OBJS += ticots.o
624 TCOO_OBJS += ticotsord.o
626 TCL_OBJS += ticlts.o
628 TL_OBJS += tl.o
630 DUMP_OBJS += dump.o
632 BPF_OBJS += bpf.o bpf_filter.o bpf_mod.o bpf_dlt.o bpf_mac.o
634 CLONE_OBJS += clone.o
636 CN_OBJS += cons.o
638 DLD_OBJS += dld_drv.o dld_proto.o dld_str.o dld_flow.o
640 DLS_OBJS += dls.o dls_link.o dls_mod.o dls_stat.o dls_mgmt.o
642 GLD_OBJS += gld.o gldutil.o
644 MAC_OBJS += mac.o mac_bcast.o mac_client.o mac_datapath_setup.o mac_flow.o
645 mac_hio.o mac_mod.o mac_ndd.o mac_provider.o mac_sched.o \
646 mac_protect.o mac_soft_ring.o mac_stat.o mac_util.o

648 MAC_6TO4_OBJS += mac_6to4.o
650 MAC_ETHER_OBJS += mac_ether.o
652 MAC_IPV4_OBJS += mac_ipv4.o

```

```

654 MAC_IPV6_OBJS +=      mac_ipv6.o
656 MAC_WIFI_OBJS +=      mac_wifi.o
658 MAC_IB_OBJS +=        mac_ib.o
660 IPTUN_OBJS +=         iptun_dev.o iptun_ctl.o iptun.o
662 AGGR_OBJS +=          aggr_dev.o aggr_ctl.o aggr_grp.o aggr_port.o \
663                        aggr_send.o aggr_recv.o aggr_lacp.o
665 SOFTMAC_OBJS +=       softmac_main.o softmac_ctl.o softmac_capab.o \
666                        softmac_dev.o softmac_stat.o softmac_pkt.o softmac_fp.o
668 NET80211_OBJS +=       net80211.o net80211_proto.o net80211_input.o \
669                        net80211_output.o net80211_node.o net80211_crypto.o \
670                        net80211_crypto_none.o net80211_crypto_wep.o net80211_ioctl.o \
671                        net80211_crypto_tkip.o net80211_crypto_ccmp.o \
672                        net80211_ht.o
674 VNIC_OBJS +=          vnic_ctl.o vnic_dev.o
676 SIMNET_OBJS +=        simnet.o
678 IB_OBJS +=            ibnex.o ibnex_ioctl.o ibnex_hca.o
680 IBCM_OBJS +=          ibcm_impl.o ibcm_sm.o ibcm_ti.o ibcm_utils.o ibcm_path.o \
681                        ibcm_arp.o ibcm_arp_link.o
683 IBDM_OBJS +=          ibdm.o
685 IBDMA_OBJS +=         ibdma.o
687 IBMF_OBJS +=          ibmf.o ibmf_impl.o ibmf_dr.o ibmf_wqe.o ibmf_ud_dest.o ibmf_mod.
688                        ibmf_send.o ibmf_recv.o ibmf_handlers.o ibmf_trans.o \
689                        ibmf_timers.o ibmf_msg.o ibmf_utils.o ibmf_rmpp.o \
690                        ibmf_saa.o ibmf_saa_impl.o ibmf_saa_utils.o ibmf_saa_events.o
692 IBTL_OBJS +=          ibtl_impl.o ibtl_util.o ibtl_mem.o ibtl_handlers.o ibtl_qp.o \
693                        ibtl_cq.o ibtl_wr.o ibtl_hca.o ibtl_chan.o ibtl_cm.o \
694                        ibtl_mcg.o ibtl_ibnex.o ibtl_srq.o ibtl_part.o
696 TAVOR_OBJS +=         tavor.o tavor_agents.o tavor_cfg.o tavor_ci.o tavor_cmd.o \
697                        tavor_cq.o tavor_event.o tavor_ioctl.o tavor_misc.o \
698                        tavor_mr.o tavor_qp.o tavor_qpmod.o tavor_rsrc.o \
699                        tavor_srq.o tavor_stats.o tavor_umap.o tavor_wr.o
701 HERMON_OBJS +=        hermon.o hermon_agents.o hermon_cfg.o hermon_ci.o hermon_cmd.o \
702                        hermon_cq.o hermon_event.o hermon_ioctl.o hermon_misc.o \
703                        hermon_mr.o hermon_qp.o hermon_qpmod.o hermon_rsrc.o \
704                        hermon_srq.o hermon_stats.o hermon_umap.o hermon_wr.o \
705                        hermon_fcoib.o hermon_fm.o
707 DAPLT_OBJS +=         daplt.o
709 SOL_OFS_OBJS +=       sol_cma.o sol_ib_cma.o sol_uobj.o \
710                        sol_ofs_debug_util.o sol_ofs_gen_util.o \
711                        sol_kverbs.o
713 SOL_UCMA_OBJS +=      sol_ucma.o
715 SOL_UVERBS_OBJS +=    sol_uverbs.o sol_uverbs_comp.o sol_uverbs_event.o \
716                        sol_uverbs_hca.o sol_uverbs_qp.o
718 SOL_UMAD_OBJS +=      sol_umad.o

```

```

720 KSTAT_OBJS +=         kstat.o
722 KSYMS_OBJS +=         ksyms.o
724 INSTANCE_OBJS +=      inst_sync.o
726 IWSCN_OBJS +=         iwscns.o
728 LOFI_OBJS +=          lofi.o LzmaDec.o
730 FSSNAP_OBJS +=        fssnap.o
732 FSSNAPIF_OBJS +=      fssnap_if.o
734 MM_OBJS +=            mem.o
736 PHYSMEM_OBJS +=       physmem.o
738 OPTIONS_OBJS +=       options.o
740 WINLOCK_OBJS +=        winlockio.o
742 PM_OBJS +=            pm.o
743 SRN_OBJS +=           srn.o
745 PSEUDO_OBJS +=        pseudonex.o
747 RAMDISK_OBJS +=       ramdisk.o
749 LLC1_OBJS +=          llc1.o
751 USBKBM_OBJS +=        usbkbm.o
753 USBWCM_OBJS +=        usbwcm.o
755 BOFI_OBJS +=          bofi.o
757 HID_OBJS +=           hid.o
759 HWA_RC_OBJS +=        hwarc.o
761 USBSKEL_OBJS +=        usbskel.o
763 USBVC_OBJS +=          usbvc.o usbvc_v412.o
765 HIDPARSER_OBJS +=     hidparser.o
767 USB_AC_OBJS +=         usb_ac.o
769 USB_AS_OBJS +=         usb_as.o
771 USB_AH_OBJS +=         usb_ah.o
773 USBMS_OBJS +=          usbms.o
775 USBPRN_OBJS +=         usbprn.o
777 UGEN_OBJS +=          ugen.o
779 USBSER_OBJS +=         usbser.o usbser_rseq.o
781 USBSACM_OBJS +=        usbsacm.o
783 USBSER_KEYSPAN_OBJS += usbser_keyspan.o keyspan_dsd.o keyspan_pipe.o

```

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

13

```

785 USBS49_FW_OBJS += keyspan_49fw.o
787 USBSPRL_OBJS += usbser_pl2303.o pl2303_dsd.o
789 WUSB_CA_OBJS += wusb_ca.o
791 USBFTDI_OBJS += usbser_uftdi.o uftdi_dsd.o
793 USBECM_OBJS += usbecm.o
795 WC_OBJS += wscons.o vcons.o
797 VCONS_CONF_OBJS += vcons_conf.o
799 SCSI_OBJS +=      scsi_capabilities.o scsi_confsubr.o scsi_control.o \
800                  scsi_data.o scsi_fm.o scsi_hba.o scsi_reset_notify.o \
801                  scsi_resource.o scsi_subr.o scsi_transport.o scsi_watch.o \
802                  smp_transport.o
804 SCSI_VHCI_OBJS +=      scsi_vhci.o mpapi_impl.o scsi_vhci_tpgs.o
806 SCSI_VHCI_F_SYM_OBJS +=      sym.o
808 SCSI_VHCI_F_TPGS_OBJS +=      tpgs.o
810 SCSI_VHCI_F_ASYM_SUN_OBJS +=  asym_sun.o
812 SCSI_VHCI_F_SYM_HDS_OBJS +=  sym_hds.o
814 SCSI_VHCI_F_TAPE_OBJS +=     tape.o
816 SCSI_VHCI_F_TPGS_TAPE_OBJS += tpgs_tape.o
818 SGEN_OBJS +=      sgen.o
820 SMP_OBJS +=      smp.o
822 SATA_OBJS +=      sata.o
824 USBA_OBJS +=      hcidi.o usba.o usbai.o hubdi.o parser.o genconsole.o \
825                  usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
826                  usba_devdb.o usba10_calls.o usba_uugen.o whcdi.o wa.o
827 USBA_WITHOUT_WUSB_OBJS +=    hcidi.o usba.o usbai.o hubdi.o parser.o gencons
828                  usbai_pipe_mgmt.o usbai_req.o usbai_util.o usbai_register.o \
829                  usba_devdb.o usba10_calls.o usba_uugen.o
831 USBA10_OBJS +=     usba10.o
833 RSM_OBJS +=      rsm.o rsmka_pathmanager.o rsmka_util.o
835 RSMOPS_OBJS +=   rsmops.o
837 S1394_OBJS +=    t1394.o t1394_errmsg.o s1394.o s1394_addr.o s1394_async.o \
838                  s1394_bus_reset.o s1394_cmp.o s1394_csr.o s1394_dev_disc.o \
839                  s1394_fa.o s1394_fcp.o \
840                  s1394_hotplug.o s1394_isoch.o s1394_misc.o h1394.o nx1394.o
842 HCI1394_OBJS +=  hcil1394.o hcil1394_async.o hcil1394_attach.o hcil1394_buf.o \
843                  hcil1394_csr.o hcil1394_detach.o hcil1394_extern.o \
844                  hcil1394_ioctl.o hcil1394_isoch.o hcil1394_isr.o \
845                  hcil1394_ixl_comp.o hcil1394_ixl_isr.o hcil1394_ixl_misc.o \
846                  hcil1394_ixl_update.o hcil1394_misc.o hcil1394_ohci.o \
847                  hcil1394_q.o hcil1394_s1394if.o hcil1394_tlabel.o \
848                  hcil1394_tlist.o hcil1394_vendor.o
850 AV1394_OBJS +=   av1394.o av1394_as.o av1394_async.o av1394_cfgrom.o \

```

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

14

```

851                  av1394_cmp.o av1394_fcp.o av1394_isoch.o av1394_isoch_chan.o \
852                  av1394_isoch_recv.o av1394_isoch_xmit.o av1394_list.o \
853                  av1394_queue.o
855 DCAM1394_OBJS +=  dcam.o dcam_frame.o dcam_param.o dcam_reg.o \
856                  dcam_ring_buff.o
858 SCSA1394_OBJS +=  hba.o sbp2_driver.o sbp2_bus.o
860 SBP2_OBJS +=      cfgrom.o sbp2.o
862 PMODEM_OBJS +=   pmodem.o pmodem_cis.o cis.o cis_callout.o cis_handlers.o cis_para
864 DSW_OBJS +=      dsw.o dsw_dev.o ii_tree.o
866 NCALL_OBJS +=    ncall.o \
867                  ncall_stub.o
869 RDC_OBJS +=      rdc.o \
870                  rdc_dev.o \
871                  rdc_io.o \
872                  rdc_clnt.o \
873                  rdc_prot_xdr.o \
874                  rdc_svc.o \
875                  rdc_bitmap.o \
876                  rdc_health.o \
877                  rdc_subr.o \
878                  rdc_diskq.o
880 RDCSRV_OBJS +=   rdcsrv.o
882 RDCSTUB_OBJS +=  rdc_stub.o
884 SDBC_OBJS +=     sd_bcache.o \
885                  sd_bio.o \
886                  sd_conf.o \
887                  sd_ft.o \
888                  sd_hash.o \
889                  sd_io.o \
890                  sd_misc.o \
891                  sd_pcu.o \
892                  sd_tdaemon.o \
893                  sd_trace.o \
894                  sd_iob_impl0.o \
895                  sd_iob_impl1.o \
896                  sd_iob_impl2.o \
897                  sd_iob_impl3.o \
898                  sd_iob_impl4.o \
899                  sd_iob_impl5.o \
900                  sd_iob_impl6.o \
901                  sd_iob_impl7.o \
902                  safestore.o \
903                  safestore_ram.o
905 NSCTL_OBJS +=    nsctl.o \
906                  nsc_cache.o \
907                  nsc_disk.o \
908                  nsc_dev.o \
909                  nsc_freeze.o \
910                  nsc_gen.o \
911                  nsc_mem.o \
912                  nsc_ncallio.o \
913                  nsc_power.o \
914                  nsc_resv.o \
915                  nsc_rmspin.o \
916                  nsc_solaris.o \

```



```

917         nsc_trap.o \
918         nsc_list.o \
919 UNISTAT_OBJS += spuni.o \
920         spcs_s_k.o

922 NSKERN_OBJS += nsc_ddi.o \
923         nsc_proc.o \
924         nsc_raw.o \
925         nsc_thread.o \
926         nskernd.o

928 SV_OBJS += sv.o

930 PMCS_OBJS += pmcs_attach.o pmcs_ds.o pmcs_intr.o pmcs_nvram.o pmcs_sata.o \
931         pmcs_scsa.o pmcs_smhba.o pmcs_subr.o pmcs_fwlog.o

933 PMCS8001FW_C_OBJS += pmcs_fw_hdr.o
934 PMCS8001FW_OBJS += $(PMCS8001FW_C_OBJS) SPCBoot.o ila.o firmware.o

936 #
937 #       Build up defines and paths.

939 ST_OBJS += st.o st_conf.o

941 EMLXS_OBJS += emlxs_clock.o emlxs_dfc.o emlxs_dhchap.o emlxs_diag.o \
942         emlxs_download.o emlxs_dump.o emlxs_els.o emlxs_event.o \
943         emlxs_fcf.o emlxs_fcp.o emlxs_fct.o emlxs_hba.o emlxs_ip.o \
944         emlxs_mbox.o emlxs_mem.o emlxs_msg.o emlxs_node.o \
945         emlxs_pkt.o emlxs_sli3.o emlxs_sli4.o emlxs_solaris.o \
946         emlxs_thread.o

948 EMLXS_FW_OBJS += emlxs_fw.o

950 OCE_OBJS += oce_buf.o oce_fm.o oce_gld.o oce_hw.o oce_intr.o oce_main.o \
951         oce_mbx.o oce_mq.o oce_queue.o oce_rx.o oce_stat.o oce_tx.o \
952         oce_utils.o

954 FCT_OBJS += discovery.o fct.o

956 QLT_OBJS += 2400.o 2500.o 8100.o qlt.o qlt_dma.o

958 SRPT_OBJS += srpt_mod.o srpt_ch.o srpt_cm.o srpt_ioc.o srpt_stp.o

960 FCOE_OBJS += fcoe.o fcoe_eth.o fcoe_fc.o

962 FCOET_OBJS += fcoet.o fcoet_eth.o fcoet_fc.o

964 FCOEI_OBJS += fcoei.o fcoei_eth.o fcoei_lv.o

966 ISCSIT_SHARED_OBJS += \
967         iscsit_common.o

969 ISCSIT_OBJS += $(ISCSIT_SHARED_OBJS) \
970         iscsit.o iscsit_tgt.o iscsit_sess.o iscsit_login.o \
971         iscsit_text.o iscsit_isns.o iscsit_radiusauth.o \
972         iscsit_radiuspacket.o iscsit_auth.o iscsit_authclient.o

974 PPPT_OBJS += alua_ic_if.o pppt.o pppt_msg.o pppt_tgt.o

976 STMF_OBJS += lun_map.o stmf.o

978 STMF_SBD_OBJS += sbd.o sbd_scsi.o sbd_pgr.o sbd_zvol.o

980 SYMSG_OBJS += sysmsg.o

982 SES_OBJS += ses.o ses_sen.o ses_safte.o ses_ses.o

```

```

984 TNF_OBJS += tnf_buf.o tnf_trace.o tnf_writer.o trace_init.o \
985         trace_funcs.o tnf_probe.o tnf.o

987 LOGINDMUX_OBJS += logindmux.o

989 DEVINFO_OBJS += devinfo.o

991 DEVPOLL_OBJS += devpoll.o

993 DEVPOOL_OBJS += devpool.o

995 I8042_OBJS += i8042.o

997 KB8042_OBJS += \
998         at_keyprocess.o \
999         kb8042.o \
1000        kb8042_keytables.o

1002 MOUSE8042_OBJS += mouse8042.o

1004 FDC_OBJS += fdc.o

1006 ASY_OBJS += asy.o

1008 ECPP_OBJS += ecpp.o

1010 VUIDM3P_OBJS += vuidmice.o vuidm3p.o

1012 VUIDM4P_OBJS += vuidmice.o vuidm4p.o

1014 VUIDM5P_OBJS += vuidmice.o vuidm5p.o

1016 VUIDPS2_OBJS += vuidmice.o vuidps2.o

1018 HPCSV_C_OBJS += hpcsvc.o

1020 PCIE_MISC_OBJS += pcie.o pcie_fault.o pcie_hp.o pciehpc.o pcishpc.o pcie_pwr.o p

1022 PCIHPNEXUS_OBJS += pcihp.o

1024 OPENEPR_OBJS += openprom.o

1026 RANDOM_OBJS += random.o

1028 PSHOT_OBJS += pshot.o

1030 GEN_DRV_OBJS += gen_drv.o

1032 TCLIENT_OBJS += tclient.o

1034 TPHCI_OBJS += tphci.o

1036 TVHCI_OBJS += tvhci.o

1038 EMUL64_OBJS += emul64.o emul64_bsd.o

1040 FCP_OBJS += fcp.o

1042 FCIP_OBJS += fcip.o

1044 FCSM_OBJS += fcsm.o

1046 FCTL_OBJS += fctl.o

1048 FP_OBJS += fp.o

```

```

1050 QLC_OBJS += ql_api.o ql_debug.o ql_hba_fru.o ql_init.o ql_iocb.o ql_ioctl.o \
1051     ql_isr.o ql_mbx.o ql_nx.o ql_xioctl.o ql_fw_table.o

1053 QLC_FW_2200_OBJS += ql_fw_2200.o

1055 QLC_FW_2300_OBJS += ql_fw_2300.o

1057 QLC_FW_2400_OBJS += ql_fw_2400.o

1059 QLC_FW_2500_OBJS += ql_fw_2500.o

1061 QLC_FW_6322_OBJS += ql_fw_6322.o

1063 QLC_FW_8100_OBJS += ql_fw_8100.o

1065 QLGE_OBJS += qlge.o qlge_dbg.o qlge_flash.o qlge_fm.o qlge_gld.o qlge_mpi.o

1067 ZCONS_OBJS += zcons.o

1069 NV_SATA_OBJS += nv_sata.o

1071 SI3124_OBJS += si3124.o

1073 AHCI_OBJS += ahci.o

1075 PCIIDE_OBJS += pci-ide.o

1077 PCEPP_OBJS += pcepp.o

1079 CPC_OBJS += cpc.o

1081 CPUID_OBJS += cpuid_drv.o

1083 SYSEVENT_OBJS += sysevent.o

1085 BL_OBJS += bl.o

1087 DRM_OBJS += drm_sunmod.o drm_kstat.o drm_agpsupport.o \
1088     drm_auth.o drm_bufs.o drm_context.o drm_dma.o \
1089     drm_drawable.o drm_drv.o drm_fops.o drm_ioctl.o drm_irq.o \
1090     drm_lock.o drm_memory.o drm_msg.o drm_pci.o drm_scatter.o \
1091     drm_cache.o drm_gem.o drm_mm.o ati_pciart.o

1093 FM_OBJS += devfm.o devfm_machdep.o

1095 RTLS_OBJS += rtls.o

1097 #
1098 #         exec modules
1099 #
1100 AOUTEXEC_OBJS += aout.o

1102 ELFEXEC_OBJS += elf.o elf_notes.o old_notes.o

1104 INTPEXEC_OBJS += intp.o

1106 SHBINEXEC_OBJS += shbin.o

1108 JAVAEXEC_OBJS += java.o

1110 #
1111 #         file system modules
1112 #
1113 AUTOFS_OBJS += auto_vfsops.o auto_vnops.o auto_subr.o auto_xdr.o auto_sys.o

```

```

1115 CACHEFS_OBJS += cachefs_cnode.o         cachefs_cod.o \
1116     cachefs_dir.o         cachefs_dlog.o  cachefs_filegrp.o \
1117     cachefs_fscache.o     cachefs_ioctl.o cachefs_log.o \
1118     cachefs_module.o \
1119     cachefs_noopc.o       cachefs_resource.o \
1120     cachefs_strict.o \
1121     cachefs_subr.o        cachefs_vfsops.o \
1122     cachefs_vnops.o

1124 DCFS_OBJS += dc_vnops.o

1126 DEVFS_OBJS += devfs_subr.o  devfs_vfsops.o  devfs_vnops.o

1128 DEV_OBJS  += sdev_subr.o     sdev_vfsops.o  sdev_vnops.o  \
1129     sdev_ptsops.o  sdev_zvolops.o  sdev_comm.o   \
1130     sdev_profile.o sdev_ncache.o  sdev_netops.o \
1131     sdev_ipnetops.o \
1132     sdev_vtops.o

1134 CTFS_OBJS += ctfs_all.o ctfs_cdir.o ctfs_ctl.o ctfs_event.o \
1135     ctfs_latest.o ctfs_root.o ctfs_sym.o ctfs_tdir.o ctfs_tmpl.o

1137 OBJFS_OBJS += objfs_vfs.o  objfs_root.o  objfs_common.o \
1138     objfs_odir.o  objfs_data.o

1140 FDFS_OBJS += fdops.o

1142 FIFO_OBJS += fifosubr.o  fifovnops.o

1144 PIPE_OBJS += pipe.o

1146 HSFS_OBJS += hsfs_node.o  hsfs_subr.o  hsfs_vfsops.o  hsfs_vnops.o \
1147     hsfs_susp.o  hsfs_rrip.o  hsfs_susp_subr.o

1149 LOFS_OBJS += lofs_subr.o  lofs_vfsops.o  lofs_vnops.o

1151 NAMEFS_OBJS += namevfs.o  namevno.o

1153 NFS_OBJS += nfs_client.o  nfs_common.o  nfs_dump.o \
1154     nfs_subr.o  nfs_vfsops.o  nfs_vnops.o \
1155     nfs_xdr.o  nfs_sys.o  nfs_strerror.o \
1156     nfs3_vfsops.o  nfs3_vnops.o  nfs3_xdr.o \
1157     nfs_acl_vnops.o  nfs_acl_xdr.o  nfs4_vfsops.o \
1158     nfs4_vnops.o  nfs4_xdr.o  nfs4_idmap.o \
1159     nfs4_shadow.o  nfs4_subr.o \
1160     nfs4_attr.o  nfs4_rnode.o  nfs4_client.o \
1161     nfs4_acache.o  nfs4_common.o  nfs4_client_state.o \
1162     nfs4_callback.o  nfs4_recovery.o  nfs4_client_secinfo.o \
1163     nfs4_client_debug.o  nfs_stats.o \
1164     nfs4_acl.o  nfs4_stub_vnops.o  nfs_cmd.o

1166 NFSSRV_OBJS += nfs_server.o  nfs_srv.o  nfs3_srv.o \
1167     nfs_acl_srv.o  nfs_auth.o  nfs_auth_xdr.o \
1168     nfs_export.o  nfs_log.o  nfs_log_xdr.o \
1169     nfs4_srv.o  nfs4_state.o  nfs4_srv_attr.o \
1170     nfs4_srv_ns.o  nfs4_db.o  nfs4_srv_deleg.o \
1171     nfs4_deleg_ops.o  nfs4_srv_readdir.o  nfs4_dispatch.o

1173 SMBSRV_SHARED_OBJS += \
1174     smb_inet.o \
1175     smb_match.o \
1176     smb_msgbuf.o \
1177     smb_oem.o \
1178     smb_string.o \
1179     smb_utf8.o \
1180     smb_door_legacy.o \

```

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

```

1181         smb_xdr.o \
1182         smb_token.o \
1183         smb_token_xdr.o \
1184         smb_sid.o \
1185         smb_native.o \
1186         smb_netbios_util.o

1188 SMBSRV_OBJS += $(SMBSRV_SHARED_OBJS) \
1189         smb_acl.o \
1190         smb_alloc.o \
1191         smb_close.o \
1192         smb_common_open.o \
1193         smb_common_transact.o \
1194         smb_create.o \
1195         smb_delete.o \
1196         smb_directory.o \
1197         smb_dispatch.o \
1198         smb_echo.o \
1199         smb_fem.o \
1200         smb_find.o \
1201         smb_flush.o \
1202         smb_fsinfo.o \
1203         smb_fsops.o \
1204         smb_init.o \
1205         smb_kdoor.o \
1206         smb_kshare.o \
1207         smb_kutil.o \
1208         smb_lock.o \
1209         smb_lock_byte_range.o \
1210         smb_locking_andx.o \
1211         smb_logoff_andx.o \
1212         smb_mangle_name.o \
1213         smb_mbuf_marshallng.o \
1214         smb_mbuf_util.o \
1215         smb_negotiate.o \
1216         smb_net.o \
1217         smb_node.o \
1218         smb_nt_cancel.o \
1219         smb_nt_create_andx.o \
1220         smb_nt_transact_create.o \
1221         smb_nt_transact_ioctl.o \
1222         smb_nt_transact_notify_change.o \
1223         smb_nt_transact_quota.o \
1224         smb_nt_transact_security.o \
1225         smb_odir.o \
1226         smb_ofile.o \
1227         smb_open_andx.o \
1228         smb_opipe.o \
1229         smb_oplock.o \
1230         smb_pathname.o \
1231         smb_print.o \
1232         smb_process_exit.o \
1233         smb_query_fileinfo.o \
1234         smb_read.o \
1235         smb_rename.o \
1236         smb_sd.o \
1237         smb_seek.o \
1238         smb_server.o \
1239         smb_session.o \
1240         smb_session_setup_andx.o \
1241         smb_set_fileinfo.o \
1242         smb_signing.o \
1243         smb_tree.o \
1244         smb_trans2_create_directory.o \
1245         smb_trans2_dfs.o \
1246         smb_trans2_find.o

```

19

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

```

1247         smb_tree_connect.o \
1248         smb_unlock_byte_range.o \
1249         smb_user.o \
1250         smb_vfs.o \
1251         smb_vops.o \
1252         smb_vss.o \
1253         smb_write.o \
1254         smb_write_raw.o

1256 PCFS_OBJS += pc_alloc.o pc_dir.o pc_node.o pc_subr.o \
1257         pc_vfsops.o pc_vnops.o

1259 PROC_OBJS += prcontrol.o prioctl.o prsubr.o prusrrio.o \
1260         prvnops.o

1262 MNTFS_OBJS += mntvfsops.o mntvnops.o

1264 SHAREFS_OBJS += sharetab.o sharefs_vfsops.o sharefs_vnops.o

1266 SPEC_OBJS += specsubr.o specvfsops.o specvnops.o

1268 SOCK_OBJS += socksubr.o sockvfsops.o sockparams.o \
1269         socksyscalls.o socktpi.o sockstr.o \
1270         sockcommon_vnops.o sockcommon_subr.o \
1271         sockcommon_sops.o sockcommon.o \
1272         sock_notsupp.o socknotify.o \
1273         nl7c.o nl7curi.o nl7chttp.o nl7clogd.o \
1274         nl7cnca.o sodirect.o sockfilter.o

1276 TMPFS_OBJS += tmp_dir.o tmp_subr.o tmp_tnode.o tmp_vfsops.o \
1277         tmp_vnops.o

1279 UDFS_OBJS += udf_alloc.o udf_bmap.o udf_dir.o \
1280         udf_inode.o udf_subr.o udf_vfsops.o \
1281         udf_vnops.o

1283 UFS_OBJS += ufs_alloc.o ufs_bmap.o ufs_dir.o ufs_xattr.o \
1284         ufs_inode.o ufs_subr.o ufs_tables.o ufs_vfsops.o \
1285         ufs_vnops.o quota.o quotacalls.o quota_ufs.o \
1286         ufs_filio.o ufs_lockfs.o ufs_thread.o ufs_trans.o \
1287         ufs_acl.o ufs_panic.o ufs_directio.o ufs_log.o \
1288         ufs_extvnops.o ufs_snap.o lufs.o lufs_thread.o \
1289         lufs_log.o lufs_map.o lufs_top.o lufs_debug.o \
1290         vscan_drv.o vscan_svc.o vscan_door.o

1292 NSMB_OBJS += smb_conn.o smb_dev.o smb_iod.o smb_pass.o \
1293         smb_rq.o smb_sign.o smb_smb.o smb_subrs.o \
1294         smb_time.o smb_tran.o smb_trantcp.o smb_usr.o \
1295         subr_mchain.o

1297 SMBFS_COMMON_OBJS += smbfs_ntacl.o \
1298         smbfs_vfsops.o smbfs_vnops.o smbfs_node.o \
1299         smbfs_acl.o smbfs_client.o smbfs_smb.o \
1300         smbfs_subr.o smbfs_subr2.o \
1301         smbfs_rwlock.o smbfs_xattr.o \
1302         $(SMBFS_COMMON_OBJS)

1305 #
1306 #           LVM modules
1307 #
1308 MD_OBJS += md.o md_error.o md_ioctl.o md_mddb.o md_names.o \
1309         md_med.o md_rename.o md_subr.o

1311 MD_COMMON_OBJS = md_convert.o md_crc.o md_revchk.o

```

20

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

21

```

1313 MD_DERIVED_OBJS = metamed_xdr.o meta_basic_xdr.o
1315 SOFTPART_OBJS += sp.o sp_ioctl.o
1317 STRIPE_OBJS += stripe.o stripe_ioctl.o
1319 HOTSPARES_OBJS += hotspares.o

1321 RAID_OBJS += raid.o raid_ioctl.o raid_replay.o raid_resync.o raid_hotspare.o

1323 MIRROR_OBJS += mirror.o mirror_ioctl.o mirror_resync.o

1325 NOTIFY_OBJS += md_notify.o

1327 TRANS_OBJS += mdtrans.o trans_ioctl.o trans_log.o

1329 ZFS_COMMON_OBJS += \
1330     arc.o \
1331     bplist.o \
1332     bpobj.o \
1333     bptree.o \
1334     dbuf.o \
1335     ddt.o \
1336     ddt_zap.o \
1337     dmuf.o \
1338     dmuf_diff.o \
1339     dmuf_send.o \
1340     dmuf_object.o \
1341     dmuf_objset.o \
1342     dmuf_traverse.o \
1343     dmuf_tx.o \
1344     dnode.o \
1345     dnode_sync.o \
1346     dsl_dir.o \
1347     dsl_dataset.o \
1348     dsl_deadlist.o \
1349     dsl_destroy.o \
1350     dsl_pool.o \
1351     dsl_synctask.o \
1352     dsl_userhold.o \
1353     dmuf_zfetch.o \
1354     dsl_deleg.o \
1355     dsl_prop.o \
1356     dsl_scan.o \
1357     zfeature.o \
1358     gzip.o \
1359     lz4.o \
1360     lzjb.o \
1361     metaslab.o \
1362     refcount.o \
1363     rrwlock.o \
1364     sa.o \
1365     sha256.o \
1366     spa.o \
1367     spa_config.o \
1368     spa_errlog.o \
1369     spa_history.o \
1370     spa_misc.o \
1371     space_map.o \
1372     txg.o \
1373     uberblock.o \
1374     unique.o \
1375     vdev.o \
1376     vdev_cache.o \
1377     vdev_file.o \
1378     vdev_label.o \

```

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

22

```

1379     vdev_mirror.o \
1380     vdev_missing.o \
1381     vdev_queue.o \
1382     vdev_raidz.o \
1383     vdev_root.o \
1384     zap.o \
1385     zap_leaf.o \
1386     zap_micro.o \
1387     zfs_byteswap.o \
1388     zfs_debug.o \
1389     zfs_fm.o \
1390     zfs_fuid.o \
1391     zfs_sa.o \
1392     zfs_znode.o \
1393     zil.o \
1394     zio.o \
1395     zio_checksum.o \
1396     zio_compress.o \
1397     zio_inject.o \
1398     zle.o \
1399     zlock.o

1401 ZFS_SHARED_OBJS += \
1402     zfeature_common.o \
1403     zfs_comutil.o \
1404     zfs_deleg.o \
1405     zfs_fletcher.o \
1406     zfs_namecheck.o \
1407     zfs_prop.o \
1408     zpool_prop.o \
1409     zprop_common.o

1411 ZFS_OBJS += \
1412     $(ZFS_COMMON_OBJS) \
1413     $(ZFS_SHARED_OBJS) \
1414     vdev_disk.o \
1415     zfs_acl.o \
1416     zfs_ctldir.o \
1417     zfs_dir.o \
1418     zfs_ioctl.o \
1419     zfs_log.o \
1420     zfs_onexit.o \
1421     zfs_replay.o \
1422     zfs_rlock.o \
1423     zfs_vfsops.o \
1424     zfs_vnops.o \
1425     zvol.o

1427 ZUT_OBJS += \
1428     zut.o

1430 # \
1431 #     streams modules
1432 #
1433 BUFMOD_OBJS += bufmod.o

1435 CONNLD_OBJS += connld.o

1437 DEDUMP_OBJS += dedump.o

1439 DRCOMPAT_OBJS += drcompat.o

1441 LDLINUX_OBJS += ldlinux.o

1443 LDTERM_OBJS += ldterm.o uwidth.o

```

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

23

```

1445 PKCT_OBJS +=      pckt.o
1447 PFMOD_OBJS +=      pfmod.o
1449 PTEM_OBJS +=      ptem.o
1451 REDIRMOD_OBJS += strredirm.o
1453 TIMOD_OBJS +=      timod.o
1455 TIRDWR_OBJS +=      tirdwr.o
1457 TTCOMPAT_OBJS += ttcompat.o
1459 LOG_OBJS +=        log.o
1461 PIPEMOD_OBJS +=      pipemod.o
1463 RPCMOD_OBJS +=      rpcmod.o      clnt_cots.o      clnt_clts.o \
1464      clnt_gen.o      clnt_perr.o      mt_rpcinit.o      rpc_calmsg.o \
1465      rpc_prot.o      rpc_sztypes.o      rpc_subr.o      rpcb_prot.o \
1466      svc.o           svc_clts.o      svc_gen.o      svc_cots.o \
1467      rpcsys.o      xdr_sizeof.o      clnt_rdma.o      svc_rdma.o \
1468      xdr_rdma.o      rdma_subr.o      xdrdma_sizeof.o
1470 TLIMOD_OBJS +=      tlimod.o      t_kalloc.o      t_kbind.o      t_kclose.o \
1471      t_kconnect.o    t_kfree.o      t_kgtstate.o    t_kopen.o \
1472      t_krcvdat.o     t_ksndudat.o  t_kspoll.o     t_kunbind.o \
1473      t_kutil.o
1475 RLMOD_OBJS +=      rlmod.o
1477 TELMOD_OBJS +=      telmod.o
1479 CRYPTMOD_OBJS +=      cryptmod.o
1481 KB_OBJS +=          kbd.o          keytables.o
1483 #
1484 #                      ID mapping module
1485 #
1486 IDMAP_OBJS +=      idmap_mod.o      idmap_kapi.o      idmap_xdr.o      idmap_cache.o
1488 #
1489 #                      scheduling class modules
1490 #
1491 SDC_OBJS +=        sysdc.o
1493 RT_OBJS +=         rt.o
1494 RT_DPTBL_OBJS +=  rt_dptbl.o
1496 TS_OBJS +=         ts.o
1497 TS_DPTBL_OBJS +=  ts_dptbl.o
1499 IA_OBJS +=         ia.o
1501 FSS_OBJS +=        fss.o
1503 FX_OBJS +=         fx.o
1504 FX_DPTBL_OBJS +=  fx_dptbl.o
1506 #
1507 #                      Inter-Process Communication (IPC) modules
1508 #
1509 IPC_OBJS +=         ipc.o

```

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

24

```

1511 IPCMSG_OBJS +=      msg.o
1513 IPCSEM_OBJS +=      sem.o
1515 IPCSHM_OBJS +=      shm.o
1517 #
1518 #                      bignum module
1519 #
1520 COMMON_BIGNUM_OBJS += bignum_mod.o bignumimpl.o
1522 BIGNUM_OBJS +=      $(COMMON_BIGNUM_OBJS) $(BIGNUM_PSR_OBJS)
1524 #
1525 #                      kernel cryptographic framework
1526 #
1527 KCF_OBJS +=          kcf.o kcf_callprov.o kcf_cbufcall.o kcf_cipher.o kcf_crypto.o \
1528      kcf_cryptoadm.o kcf_ctxops.o kcf_digest.o kcf_dual.o \
1529      kcf_keys.o kcf_mac.o kcf_mech_tabs.o kcf_miscapi.o \
1530      kcf_object.o kcf_policy.o kcf_prov_lib.o kcf_prov_tabs.o \
1531      kcf_sched.o kcf_session.o kcf_sign.o kcf_spi.o kcf_verify.o \
1532      kcf_random.o modes.o ecb.o cbc.o ctr.o ccm.o gcm.o \
1533      fips_random.o
1535 CRYPTOADM_OBJS +=      cryptoadm.o
1537 CRYPTO_OBJS +=        crypto.o
1539 DPROV_OBJS +=         dprov.o
1541 DCA_OBJS +=           dca.o dca_3des.o dca_debug.o dca_dsa.o dca_kstat.o dca_rng.o \
1542      dca_rsa.o
1544 AESPROV_OBJS +=      aes.o aes_impl.o aes_modes.o
1546 ARCFOURPROV_OBJS +=  arcfour.o arcfour_crypt.o
1548 BLOWFISHPROV_OBJS += blowfish.o blowfish_impl.o
1550 ECCPROV_OBJS +=      ecc.o ec.o ec2_163.o ec2_mont.o ecdecode.o ecl_mult.o \
1551      ecp_384.o ecp_jac.o ec2_193.o ecl.o ecp_192.o ecp_521.o \
1552      ecp_jm.o ec2_233.o ecl_curve.o ecp_224.o ecp_aff.o \
1553      ecp_mont.o ec2_aff.o ec_naf.o ecl_gf.o ecp_256.o mp_gf2m.o \
1554      mpi.o mplogic.o mpmontg.o mpprime.o oid.o \
1555      secitem.o ec2_test.o ecp_test.o
1557 RSAPROV_OBJS +=      rsa.o rsa_impl.o pkcs1.o
1559 SWRANDPROV_OBJS +=  swrand.o
1561 #
1562 #                      kernel SSL
1563 #
1564 KSSL_OBJS +=         kssl.o ksslioc1.o
1566 KSSL_SOCKETFIL_MOD_OBJS += ksslfilter.o ksslapi.o ksslrec.o
1568 #
1569 #                      misc. modules
1570 #
1572 C2AUDIT_OBJS +=      adr.o audit.o audit_event.o audit_io.o \
1573      audit_path.o audit_start.o audit_syscalls.o audit_token.o \
1574      audit_mem.o
1576 PCIC_OBJS +=         pcic.o

```

```

1578 RPCSEC_OBJS += secmod.o      sec_clnt.o      sec_svc.o      sec_gen.o \
1579                auth_des.o     auth_kern.o     auth_none.o    auth_loopb.o\
1580                authdesprt.o   authdesubr.o   authu_prot.o \
1581                key_call.o     key_prot.o     svc_authu.o    svcauthdes.o

1583 RPCSEC_GSS_OBJS +=          rpcsec_gssmod.o rpcsec_gss.o rpcsec_gss_misc.o \
1584                rpcsec_gss_utils.o svc_rpcsec_gss.o

1586 CONSCONFIG_OBJS += consconfig.o

1588 CONSCONFIG_DACF_OBJS += consconfig_dacf.o consplat.o

1590 TEM_OBJS += tem.o tem_safe.o 6x10.o 7x14.o 12x22.o

1592 KBTRANS_OBJS +=                \
1593                kbtrans.o         \
1594                kbtrans_keytables.o \
1595                kbtrans_polled.o   \
1596                kbtrans_streams.o  \
1597                usb_keytables.o

1599 KGSSD_OBJS += gssd_clnt_stubs.o gssd_handle.o gssd_prot.o \
1600                gss_display_name.o gss_release_name.o gss_import_name.o \
1601                gss_release_buffer.o gss_release_oid_set.o gen_oids.o gssdmod.o

1603 KGSSD_DERIVED_OBJS = gssd_xdr.o

1605 KGSS_DUMMY_OBJS += dmech.o

1607 KSOCKET_OBJS += ksocket.o ksocket_mod.o

1609 CRYPTO= cksumtypes.o decrypt.o encrypt.o encrypt_length.o etypes.o \
1610          nfold.o verify_checksum.o prng.o block_size.o make_checksum.o\
1611          checksum_length.o hmac.o default_state.o mandatory_sumtype.o

1613 # crypto/des
1614 CRYPTO_DES= f_cbc.o f_cksum.o f_parity.o weak_key.o d3_cbc.o ef_crypto.o

1616 CRYPTO_DK= checksum.o derive.o dk_decrypt.o dk_encrypt.o

1618 CRYPTO_ARCFOUR= k5_arcfour.o

1620 # crypto/enc_provider
1621 CRYPTO_ENC= des.o des3.o arcfour_provider.o aes_provider.o

1623 # crypto/hash_provider
1624 CRYPTO_HASH= hash_kef_generic.o hash_kmd5.o hash_crc32.o hash_kshal.o

1626 # crypto/keyhash_provider
1627 CRYPTO_KEYHASH= descbc.o k5_kmd5des.o k_hmac_md5.o

1629 # crypto/crc32
1630 CRYPTO_CRC32= crc32.o

1632 # crypto/old
1633 CRYPTO_OLD= old_decrypt.o old_encrypt.o

1635 # crypto/raw
1636 CRYPTO_RAW= raw_decrypt.o raw_encrypt.o

1638 K5_KRB= kfree.o copy_key.o \
1639         parse.o init_ctx.o \
1640         ser_adata.o ser_addr.o \
1641         ser_auth.o ser_cksum.o \
1642         ser_key.o ser_princ.o \

```

```

1643                serialize.o unparse.o \
1644                ser_actx.o

1646 K5_OS= timeofday.o toffset.o \
1647         init_os_ctx.o c_ustime.o

1649 SEAL=
1650 # EXPORT DELETE START
1651 SEAL= seal.o unseal.o
1652 # EXPORT DELETE END

1654 MECH= delete_sec_context.o \
1655        import_sec_context.o \
1656        gssapi_krb5.o \
1657        k5seal.o k5unseal.o k5sealv3.o \
1658        ser_sctx.o \
1659        sign.o \
1660        util_crypt.o \
1661        util_validate.o util_ordering.o \
1662        util_seqnum.o util_set.o util_seed.o \
1663        wrap_size_limit.o verify.o

1667 MECH_GEN= util_token.o

1670 KGSS_KRB5_OBJS += krb5mech.o \
1671                 $(MECH) $(SEAL) $(MECH_GEN) \
1672                 $(CRYPTO) $(CRYPTO_DES) $(CRYPTO_DK) $(CRYPTO_ARCFOUR) \
1673                 $(CRYPTO_ENC) $(CRYPTO_HASH) \
1674                 $(CRYPTO_KEYHASH) $(CRYPTO_CRC32) \
1675                 $(CRYPTO_OLD) \
1676                 $(CRYPTO_RAW) $(K5_KRB) $(K5_OS)

1678 DES_OBJS += des_crypt.o des_impl.o des_ks.o des_soft.o

1680 DLBOOT_OBJS += bootparam_xdr.o nfs_dlinet.o scan.o

1682 KRTLD_OBJS += kobj_bootflags.o getoptstr.o \
1683              kobj.o kobj_kdi.o kobj_lm.o kobj_subr.o

1685 MOD_OBJS += modctl.o modsubr.o modsysfile.o modconf.o modhash.o

1687 STRPLUMB_OBJS += strplumb.o

1689 CPR_OBJS += cpr_driver.o cpr_dump.o \
1690            cpr_main.o cpr_misc.o cpr_mod.o cpr_stat.o \
1691            cpr_uthread.o

1693 PROF_OBJS += prf.o

1695 SE_OBJS += se_driver.o

1697 SYSACCT_OBJS += acct.o

1699 ACCTCTL_OBJS += acctctl.o

1701 EXACCTSYS_OBJS += exacctsys.o

1703 KAIQ_OBJS += aio.o

1705 PCMCIA_OBJS += pcmcia.o cs.o cis.o cis_callout.o cis_handlers.o cis_params.o

1707 BUSRA_OBJS += busra.o

```

```

1709 PCS_OBJS += pcs.o
1711 PCAN_OBJS += pcan.o
1713 PCATA_OBJS += pcide.o pcdisk.o pclabel.o pcata.o
1715 PCSER_OBJS += pcser.o pcser_cis.o
1717 PCWL_OBJS += pcwl.o
1719 PSET_OBJS += pset.o
1721 OHCI_OBJS += ohci.o ohci_hub.o ohci_polled.o
1723 UHCI_OBJS += uhci.o uhciutil.o uhcitgt.o uhcihub.o uhcipolled.o
1725 EHCI_OBJS += ehci.o ehci_hub.o ehci_xfer.o ehci_intr.o ehci_util.o ehci_polled.o
1727 HUBD_OBJS += hubd.o
1729 USB_MID_OBJS += usb_mid.o
1731 USB_IA_OBJS += usb_ia.o
1733 UWBA_OBJS += uwba.o uwbai.o
1735 SCSA2USB_OBJS += scsa2usb.o usb_ms_bulkonly.o usb_ms_cbi.o
1737 HWAHC_OBJS += hwahc.o hwahc_util.o
1739 WUSB_DF_OBJS += wusb_df.o
1740 WUSB_FWMOD_OBJS += wusb_fwmod.o
1742 IPF_OBJS += ip_fil_solaris.o fil.o solaris.o ip_state.o ip_frag.o ip_nat.o \
1743 ip_proxy.o ip_auth.o ip_pool.o ip_hstable.o ip_lookup.o \
1744 ip_log.o misc.o ip_compat.o ip_nat6.o drand48.o
1746 IBD_OBJS += ibd.o ibd_cm.o
1748 EIBNX_OBJS += enx_main.o enx_hdlrs.o enx_ibt.o enx_log.o enx_fip.o \
1749 enx_misc.o enx_q.o enx_ctl.o
1751 EOIB_OBJS += eib_adm.o eib_chan.o eib_cmn.o eib_ctl.o eib_data.o \
1752 eib_fip.o eib_ibt.o eib_log.o eib_mac.o eib_main.o \
1753 eib_rsrc.o eib_svc.o eib_vnic.o
1755 DLPSTUB_OBJS += dlpistub.o
1757 SDP_OBJS += sdpddi.o
1759 TRILL_OBJS += trill.o
1761 CTF_OBJS += ctf_create.o ctf_decl.o ctf_error.o ctf_hash.o ctf_labels.o \
1762 ctf_lookup.o ctf_open.o ctf_types.o ctf_util.o ctf_subr.o ctf_mod.o
1764 SMBIOS_OBJS += smb_error.o smb_info.o smb_open.o smb_subr.o smb_dev.o
1766 RPCIB_OBJS += rpcib.o
1768 KMDB_OBJS += kdrv.o
1770 AFE_OBJS += afe.o
1772 BGE_OBJS += bge_main2.o bge_chip2.o bge_kstats.o bge_log.o bge_ndd.o \
1773 bge_atomic.o bge_mii.o bge_send.o bge_rcv2.o bge_mii_5906.o

```

```

1775 DMFE_OBJS += dmfe_log.o dmfe_main.o dmfe_mii.o
1777 EFE_OBJS += efe.o
1779 ELXL_OBJS += elxl.o
1781 HME_OBJS += hme.o
1783 IXGB_OBJS += ixgb.o ixgb_atomic.o ixgb_chip.o ixgb_gld.o ixgb_kstats.o \
1784 ixgb_log.o ixgb_ndd.o ixgb_rx.o ixgb_tx.o ixgb_xmii.o
1786 NGE_OBJS += nge_main.o nge_atomic.o nge_chip.o nge_ndd.o nge_kstats.o \
1787 nge_log.o nge_rx.o nge_tx.o nge_xmii.o
1789 PCN_OBJS += pcn.o
1791 RGE_OBJS += rge_main.o rge_chip.o rge_ndd.o rge_kstats.o rge_log.o rge_rxtx.o
1793 URTW_OBJS += urtw.o
1795 ARN_OBJS += arn_hw.o arn_eeprom.o arn_mac.o arn_calib.o arn_ani.o arn_phy.o arn_
1796 arn_main.o arn_rcv.o arn_xmit.o arn_rc.o
1798 ATH_OBJS += ath_aux.o ath_main.o ath_osdep.o ath_rate.o
1800 ATU_OBJS += atu.o
1802 IPW_OBJS += ipw2100_hw.o ipw2100.o
1804 IWI_OBJS += ipw2200_hw.o ipw2200.o
1806 IWH_OBJS += iwh.o
1808 IWK_OBJS += iwk2.o
1810 IWP_OBJS += iwp.o
1812 MWL_OBJS += mwl.o
1814 MWLFW_OBJS += mwlfw_mode.o
1816 WPI_OBJS += wpi.o
1818 RAL_OBJS += rt2560.o ral_rate.o
1820 RUM_OBJS += rum.o
1822 RWD_OBJS += rt2661.o
1824 RWN_OBJS += rt2860.o
1826 UATH_OBJS += uath.o
1828 UATHFW_OBJS += uathfw_mod.o
1830 URAL_OBJS += ural.o
1832 RTW_OBJS += rtw.o smc93cx6.o rtwphy.o rtwphyio.o
1834 ZYD_OBJS += zyd.o zyd_usb.o zyd_hw.o zyd_fw.o
1836 MXFE_OBJS += mxfe.o
1838 MPTSAS_OBJS += mptsas.o mptsas_impl.o mptsas_init.o mptsas_raid.o mptsas_smhba.o
1840 SFE_OBJS += sfe.o sfe_util.o

```

```

1842 BFE_OBJS += bfe.o
1844 BRIDGE_OBJS += bridge.o
1846 IDM_SHARED_OBJS += base64.o
1848 IDM_OBJS += $(IDM_SHARED_OBJS) \
1849 idm.o idm_impl.o idm_text.o idm_conn_sm.o idm_so.o
1851 VR_OBJS += vr.o
1853 ATGE_OBJS += atge_main.o atge_lle.o atge_mii.o atge_ll.o atge_llc.o
1855 YGE_OBJS = yge.o
1857 #
1858 # Build up defines and paths.
1859 #
1860 LINT_DEFS += -Dunix
1862 #
1863 # This duality can be removed when the native and target compilers
1864 # are the same (or at least recognize the same command line syntax!)
1865 # It is a bug in the current compilation system that the assembler
1866 # can't process the -Y I, flag.
1867 #
1868 NATIVE_INC_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1869 AS_INC_PATH += $(INC_PATH) -I$(UTSBASE)/common
1870 INCLUDE_PATH += $(INC_PATH) $(CCYFLAG)$(UTSBASE)/common
1872 PCIEB_OBJS += pcieb.o
1874 # Chelsio N110 10G NIC driver module
1875 #
1876 CH_OBJS = ch.o glue.o pe.o sge.o
1878 CH_COM_OBJS = ch_mac.o ch_subr.o csapi.o espi.o ixfl1010.o mc3.o mc4.o mc5.o \
1879 mv88elxxx.o mv88x20lx.o my3126.o pm3393.o tp.o ulp.o \
1880 vsc7321.o vsc7326.o xpak.o
1882 #
1883 # PCI strings file
1884 #
1885 PCI_STRING_OBJS = pci_strings.o
1887 NET_DACF_OBJS += net_dacf.o
1889 #
1890 # Xframe 10G NIC driver module
1891 #
1892 XGE_OBJS = xge.o xgell.o
1894 XGE_HAL_OBJS = xgehal-channel.o xgehal-fifo.o xgehal-ring.o xgehal-config.o \
1895 xgehal-driver.o xgehal-mm.o xgehal-stats.o xgehal-device.o \
1896 xge-queue.o xgehal-mgmt.o xgehal-mgmtaux.o
1898 #
1899 # e1000g module
1900 #
1901 E1000G_OBJS += e1000_80003es2lan.o e1000_82540.o e1000_82541.o e1000_82542.o \
1902 e1000_82543.o e1000_82571.o e1000_api.o e1000_ich8lan.o \
1903 e1000_mac.o e1000_manage.o e1000_nvmm.o e1000_osdep.o \
1904 e1000_phy.o e1000g_debug.o e1000g_main.o e1000g_alloc.o \
1905 e1000g_tx.o e1000g_rx.o e1000g_stat.o

```

```

1907 #
1908 # Intel 82575 1G NIC driver module
1909 #
1910 IGB_OBJS = igb_82575.o igb_api.o igb_mac.o igb_manage.o \
1911 igb_nvmm.o igb_osdep.o igb_phy.o igb_buf.o \
1912 igb_debug.o igb_gld.o igb_log.o igb_main.o \
1913 igb_rx.o igb_stat.o igb_tx.o
1915 #
1916 # Intel Pro/100 NIC driver module
1917 #
1918 IPRB_OBJS = iprb.o
1920 #
1921 # Intel 10GbE PCIE NIC driver module
1922 #
1923 IXGBE_OBJS = ixgbe_82598.o ixgbe_82599.o ixgbe_api.o \
1924 ixgbe_common.o ixgbe_phy.o \
1925 ixgbe_buf.o ixgbe_debug.o ixgbe_gld.o \
1926 ixgbe_log.o ixgbe_main.o \
1927 ixgbe_osdep.o ixgbe_rx.o ixgbe_stat.o \
1928 ixgbe_tx.o ixgbe_x540.o ixgbe_mbx.o
1930 #
1931 # NIU 10G/1G driver module
1932 #
1933 NXGE_OBJS = nxge_mac.o nxge_ipp.o nxge_rxdma.o \
1934 nxge_txdma.o nxge_txc.o nxge_main.o \
1935 nxge_hw.o nxge_fzc.o nxge_virtual.o \
1936 nxge_send.o nxge_classify.o nxge_fflp.o \
1937 nxge_fflp_hash.o nxge_ndd.o nxge_kstats.o \
1938 nxge_zcp.o nxge_fm.o nxge_espc.o nxge_hv.o \
1939 nxge_hio.o nxge_hio_guest.o nxge_intr.o
1941 NXGE_NPI_OBJS = \
1942 npi.o npi_mac.o npi_ipp.o \
1943 npi_txdma.o npi_rxdma.o npi_txc.o \
1944 npi_zcp.o npi_espc.o npi_fflp.o \
1945 npi_vir.o
1947 NXGE_HCALL_OBJS = \
1948 nxge_hcall.o
1950 #
1951 # Virtio modules
1952 #
1954 # Virtio core
1955 VIRTIO_OBJS = virtio.o
1957 # Virtio block driver
1958 VIOBLK_OBJS = vioblk.o
1960 #Virtio network driver
1961 VIOIF_OBJS = vioif.o
1963 #
1964 # kiconv modules
1965 #
1966 KICONV_EMEA_OBJS += kiconv_emea.o
1968 KICONV_JA_OBJS += kiconv_ja.o
1970 KICONV_KO_OBJS += kiconv_cck_common.o kiconv_ko.o
1972 KICONV_SC_OBJS += kiconv_cck_common.o kiconv_sc.o

```



```
1974 KICONV_TC_OBJS += kiconv_cck_common.o kiconv_tc.o

1976 #
1977 #     AAC module
1978 #
1979 AAC_OBJS = aac.o aac_ioctl.o

1981 #
1982 #     sdcard modules
1983 #
1984 SDA_OBJS =     sda_cmd.o sda_host.o sda_init.o sda_mem.o sda_mod.o sda_slot.o
1985 SDHOST_OBJS = sdhost.o

1987 #
1988 #     hxge 10G driver module
1989 #
1990 HXGE_OBJS =     hxge_main.o hxge_vmac.o hxge_send.o           \
1991                hxge_txdma.o hxge_rxdma.o hxge_virtual.o     \
1992                hxge_fm.o hxge_fzc.o hxge_hw.o hxge_kstats.o \
1993                hxge_ndd.o hxge_pfc.o                       \
1994                hpi.o hpi_vmac.o hpi_rxdma.o hpi_txdma.o     \
1995                hpi_vir.o hpi_pfc.o

1997 #
1998 #     MEGARAID_SAS module
1999 #
2000 MEGA_SAS_OBJS = megaraid_sas.o

2002 #
2003 #     MR_SAS module
2004 #
2005 MR_SAS_OBJS = ld_pd_map.o mr_sas.o mr_sas_tbolt.o mr_sas_list.o

2007 #
2008 #     ISCSI_INITIATOR module
2009 #
2010 ISCSI_INITIATOR_OBJS = chap.o iscsi_io.o iscsi_thread.o      \
2011                        iscsi_ioctl.o iscsid.o iscsi.o        \
2012                        iscsi_login.o isns_client.o iscsiAuthClient.o \
2013                        iscsi_lun.o iscsiAuthClientGlue.o     \
2014                        iscsi_net.o nvfile.o iscsi_cmd.o      \
2015                        iscsi_queue.o persistent.o iscsi_conn.o \
2016                        iscsi_sess.o radius_auth.o iscsi_crc.o \
2017                        iscsi_stats.o radius_packet.o iscsi_doorclt.o \
2018                        iscsi_targetparam.o utils.o kifconf.o

2020 #
2021 #     ntxn 10Gb/1Gb NIC driver module
2022 #
2023 NTXN_OBJS =     unm_nic_init.o unm_gem.o unm_nic_hw.o unm_ndd.o \
2024                unm_nic_main.o unm_nic_isr.o unm_nic_ctx.o niu.o

2026 #
2027 #     Myricom 10Gb NIC driver module
2028 #
2029 MYRI10GE_OBJS = myril0ge.o myril0ge_lro.o

2031 #
2032 #     nulldriver module
2033 NULLDRIVER_OBJS =     nulldriver.o

2035 TPM_OBJS =     tpm.o tpm_hcall.o
```

```

*****
72752 Thu Mar 28 23:16:18 2013
new/usr/src/uts/common/Makefile.rules
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xfee@gmail.com
*****
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) 2012 Nexenta Systems, Inc. All rights reserved.
25 #
26 #
27 #
28 # uts/common/Makefile.rules
29 #
30 # This Makefile defines all the file build rules for the directory
31 # uts/common and its children. These are the source files which may
32 # be considered common to all SunOS systems.
33 #
34 # The following two-level ordering must be maintained in this file.
35 # Lines are sorted first in order of decreasing specificity based on
36 # the first directory component. That is, sun4u rules come before
37 # sparc rules come before common rules.
38 #
39 # Lines whose initial directory components are equal are sorted
40 # alphabetically by the remaining components.
41 #
42 #
43 # Section 1a: C objects build rules
44 #
45 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/aes/%.c
46 $(COMPILE.c) -o $@ $<
47 $(CTFCONVERT_O)
48 #
49 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/arcfour/%.c
50 $(COMPILE.c) -o $@ $<
51 $(CTFCONVERT_O)
52 #
53 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/blowfish/%.c
54 $(COMPILE.c) -o $@ $<
55 $(CTFCONVERT_O)
56 #
57 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/ecc/%.c
58 $(COMPILE.c) -o $@ $<

```

```

59 $(CTFCONVERT_O)
60 #
61 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/modes/%.c
62 $(COMPILE.c) -o $@ $<
63 $(CTFCONVERT_O)
64 #
65 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/padding/%.c
66 $(COMPILE.c) -o $@ $<
67 $(CTFCONVERT_O)
68 #
69 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/rng/%.c
70 $(COMPILE.c) -o $@ $<
71 $(CTFCONVERT_O)
72 #
73 $(OBJJS_DIR)/%.o: $(COMMONBASE)/crypto/rsa/%.c
74 $(COMPILE.c) -o $@ $<
75 $(CTFCONVERT_O)
76 #
77 $(OBJJS_DIR)/%.o: $(COMMONBASE)/bignum/%.c
78 $(COMPILE.c) -o $@ $<
79 $(CTFCONVERT_O)
80 #
81 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/bignum/%.c
82 $(COMPILE.c) -o $@ $<
83 $(CTFCONVERT_O)
84 #
85 $(OBJJS_DIR)/%.o: $(COMMONBASE)/mpi/%.c
86 $(COMPILE.c) -o $@ $<
87 $(CTFCONVERT_O)
88 #
89 $(OBJJS_DIR)/%.o: $(COMMONBASE)/acl/%.c
90 $(COMPILE.c) -o $@ $<
91 $(CTFCONVERT_O)
92 #
93 $(OBJJS_DIR)/%.o: $(COMMONBASE)/avl/%.c
94 $(COMPILE.c) -o $@ $<
95 $(CTFCONVERT_O)
96 #
97 $(OBJJS_DIR)/%.o: $(COMMONBASE)/ucode/%.c
98 $(COMPILE.c) -o $@ $<
99 $(CTFCONVERT_O)
100 #
101 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/brand/sn1/%.c
102 $(COMPILE.c) -o $@ $<
103 $(CTFCONVERT_O)
104 #
105 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/brand/solaris10/%.c
106 $(COMPILE.c) -o $@ $<
107 $(CTFCONVERT_O)
108 #
109 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/c2/%.c
110 $(COMPILE.c) -o $@ $<
111 $(CTFCONVERT_O)
112 #
113 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/conf/%.c
114 $(COMPILE.c) -o $@ $<
115 $(CTFCONVERT_O)
116 #
117 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/contract/%.c
118 $(COMPILE.c) -o $@ $<
119 $(CTFCONVERT_O)
120 #
121 $(OBJJS_DIR)/%.o: $(UTSBASE)/common/cpr/%.c
122 $(COMPILE.c) -o $@ $<
123 $(CTFCONVERT_O)

```

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

```

125 $(OBJSDIR)/%.o: $(UTSBASE)/common/ctf/%.c
126 $(COMPILE.c) -o $@ $<
127 $(CTFCONVERT_O)

129 $(OBJSDIR)/%.o: $(COMMONBASE)/ctf/%.c
130 $(COMPILE.c) -o $@ $<
131 $(CTFCONVERT_O)

133 $(OBJSDIR)/%.o: $(COMMONBASE)/crypto/des/%.c
134 $(COMPILE.c) -o $@ $<
135 $(CTFCONVERT_O)

137 $(OBJSDIR)/%.o: $(COMMONBASE)/smbios/%.c
138 $(COMPILE.c) -o $@ $<
139 $(CTFCONVERT_O)

141 $(OBJSDIR)/%.o: $(UTSBASE)/common/des/%.c
142 $(COMPILE.c) -o $@ $<
143 $(CTFCONVERT_O)

145 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/api/%.c
146 $(COMPILE.c) -o $@ $<
147 $(CTFCONVERT_O)

149 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/core/%.c
150 $(COMPILE.c) -o $@ $<
151 $(CTFCONVERT_O)

153 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/io/%.c
154 $(COMPILE.c) -o $@ $<
155 $(CTFCONVERT_O)

157 $(OBJSDIR)/%.o: $(UTSBASE)/common/crypto/spi/%.c
158 $(COMPILE.c) -o $@ $<
159 $(CTFCONVERT_O)

161 $(OBJSDIR)/%.o: $(COMMONBASE)/pci/%.c
162 $(COMPILE.c) -o $@ $<
163 $(CTFCONVERT_O)

165 $(OBJSDIR)/%.o: $(COMMONBASE)/devid/%.c
166 $(COMPILE.c) -o $@ $<
167 $(CTFCONVERT_O)

169 $(OBJSDIR)/%.o: $(UTSBASE)/common/disp/%.c
170 $(COMPILE.c) -o $@ $<
171 $(CTFCONVERT_O)

173 $(OBJSDIR)/%.o: $(UTSBASE)/common/dtrace/%.c
174 $(COMPILE.c) -o $@ $<
175 $(CTFCONVERT_O)

177 $(OBJSDIR)/%.o: $(COMMONBASE)/execct/%.c
178 $(COMPILE.c) -o $@ $<
179 $(CTFCONVERT_O)

181 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/aout/%.c
182 $(COMPILE.c) -o $@ $<
183 $(CTFCONVERT_O)

185 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/elf/%.c
186 $(COMPILE.c) -o $@ $<
187 $(CTFCONVERT_O)

189 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/intp/%.c
190 $(COMPILE.c) -o $@ $<

```

3

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

```

191 $(CTFCONVERT_O)

193 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/shbin/%.c
194 $(COMPILE.c) -o $@ $<
195 $(CTFCONVERT_O)

197 $(OBJSDIR)/%.o: $(UTSBASE)/common/exec/java/%.c
198 $(COMPILE.c) -o $@ $<
199 $(CTFCONVERT_O)

201 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/%.c
202 $(COMPILE.c) -o $@ $<
203 $(CTFCONVERT_O)

205 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/autofs/%.c
206 $(COMPILE.c) -o $@ $<
207 $(CTFCONVERT_O)

209 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/cacheofs/%.c
210 $(COMPILE.c) -o $@ $<
211 $(CTFCONVERT_O)

213 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/dcofs/%.c
214 $(COMPILE.c) -o $@ $<
215 $(CTFCONVERT_O)

217 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/devfs/%.c
218 $(COMPILE.c) -o $@ $<
219 $(CTFCONVERT_O)

221 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/ctfs/%.c
222 $(COMPILE.c) -o $@ $<
223 $(CTFCONVERT_O)

225 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/doorfs/%.c
226 $(COMPILE.c) -o $@ $<
227 $(CTFCONVERT_O)

229 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/dev/%.c
230 $(COMPILE.c) -o $@ $<
231 $(CTFCONVERT_O)

233 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/fd/%.c
234 $(COMPILE.c) -o $@ $<
235 $(CTFCONVERT_O)

237 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/fifoofs/%.c
238 $(COMPILE.c) -o $@ $<
239 $(CTFCONVERT_O)

241 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/hsfs/%.c
242 $(COMPILE.c) -o $@ $<
243 $(CTFCONVERT_O)

245 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/lofs/%.c
246 $(COMPILE.c) -o $@ $<
247 $(CTFCONVERT_O)

249 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/mntfs/%.c
250 $(COMPILE.c) -o $@ $<
251 $(CTFCONVERT_O)

253 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/namefs/%.c
254 $(COMPILE.c) -o $@ $<
255 $(CTFCONVERT_O)

```

4

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

5

```

257 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/nfs/%.c
258 $(COMPILE.c) -o $@ $<
259 $(CTFCONVERT_O)

261 $(OBJSDIR)/%.o: $(COMMONBASE)/smbsrv/%.c
262 $(COMPILE.c) -o $@ $<
263 $(CTFCONVERT_O)

265 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbsrv/%.c
266 $(COMPILE.c) -o $@ $<
267 $(CTFCONVERT_O)

269 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/objfs/%.c
270 $(COMPILE.c) -o $@ $<
271 $(CTFCONVERT_O)

273 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/pcfs/%.c
274 $(COMPILE.c) -o $@ $<
275 $(CTFCONVERT_O)

277 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/portfs/%.c
278 $(COMPILE.c) -o $@ $<
279 $(CTFCONVERT_O)

281 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/proc/%.c
282 $(COMPILE.c) -o $@ $<
283 $(CTFCONVERT_O)

285 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/sharefs/%.c
286 $(COMPILE.c) -o $@ $<
287 $(CTFCONVERT_O)

289 $(OBJSDIR)/%.o: $(COMMONBASE)/smbclnt/%.c
290 $(COMPILE.c) -o $@ $<
291 $(CTFCONVERT_O)

293 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbclnt/net smb/%.c
294 $(COMPILE.c) -o $@ $<
295 $(CTFCONVERT_O)

297 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
298 $(COMPILE.c) -o $@ $<
299 $(CTFCONVERT_O)

301 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/sockfs/%.c
302 $(COMPILE.c) -o $@ $<
303 $(CTFCONVERT_O)

305 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/specfs/%.c
306 $(COMPILE.c) -o $@ $<
307 $(CTFCONVERT_O)

309 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/swapfs/%.c
310 $(COMPILE.c) -o $@ $<
311 $(CTFCONVERT_O)

313 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/tmpfs/%.c
314 $(COMPILE.c) -o $@ $<
315 $(CTFCONVERT_O)

317 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/udfs/%.c
318 $(COMPILE.c) -o $@ $<
319 $(CTFCONVERT_O)

321 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/uifs/%.c
322 $(COMPILE.c) -o $@ $<

```

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

6

```

323 $(CTFCONVERT_O)

325 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/vscan/%.c
326 $(COMPILE.c) -o $@ $<
327 $(CTFCONVERT_O)

329 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/zfs/%.c
330 $(COMPILE.c) -o $@ $<
331 $(CTFCONVERT_O)

333 $(OBJSDIR)/%.o: $(UTSBASE)/common/fs/zut/%.c
334 $(COMPILE.c) -o $@ $<
335 $(CTFCONVERT_O)

337 $(OBJSDIR)/%.o: $(COMMONBASE)/xattr/%.c
338 $(COMPILE.c) -o $@ $<
339 $(CTFCONVERT_O)

341 $(OBJSDIR)/%.o: $(COMMONBASE)/zfs/%.c
342 $(COMPILE.c) -o $@ $<
343 $(CTFCONVERT_O)

345 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
346 $(COMPILE.c) -o $@ $<
347 $(CTFCONVERT_O)

349 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/scsi/adapters/pmcs/%.bin
350 $(COMPILE.b) -o $@ $<
351 $(CTFCONVERT_O)

353 $(OBJSDIR)/%.o: $(COMMONBASE)/fsreparse/%.c
354 $(COMPILE.c) -o $@ $<
355 $(CTFCONVERT_O)

357 KMECHKRB5_BASE=$(UTSBASE)/common/gssapi/mechs/krb5

359 KGSSDFLAGS=-I $(UTSBASE)/common/gssapi/include

361 # Note, KRB5_DEFS can be assigned various preprocessor flags,
362 # typically -D defines on the make invocation. The standard compiler
363 # flags will not be overwritten.
364 KGSSDFLAGS += $(KRB5_DEFS)

366 $(OBJSDIR)/%.o: $(UTSBASE)/common/gssapi/%.c
367 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
368 $(CTFCONVERT_O)

370 $(OBJSDIR)/%.o: $(UTSBASE)/common/gssapi/mechs/dummy/%.c
371 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
372 $(CTFCONVERT_O)

374 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/%.c
375 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
376 $(CTFCONVERT_O)

378 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/%.c
379 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
380 $(CTFCONVERT_O)

382 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/des/%.c
383 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
384 $(CTFCONVERT_O)

386 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/arcfour/%.c
387 $(COMPILE.c) $(KGSSDFLAGS) -o $@ $<
388 $(CTFCONVERT_O)

```

```

390 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/dk/%.c
391 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
392 $(CTFCONVERT_O)

394 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
395 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
396 $(CTFCONVERT_O)

398 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
399 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
400 $(CTFCONVERT_O)

402 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
403 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
404 $(CTFCONVERT_O)

406 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/raw/%.c
407 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
408 $(CTFCONVERT_O)

410 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/crypto/old/%.c
411 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
412 $(CTFCONVERT_O)

414 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/krb5/krb/%.c
415 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
416 $(CTFCONVERT_O)

418 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/krb5/os/%.c
419 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
420 $(CTFCONVERT_O)

422 $(OBJSDIR)/ser_sctx.o := CPPFLAGS += -DPROVIDE_KERNEL_IMPORT=1

424 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/mech/%.c
425 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
426 $(CTFCONVERT_O)

428 $(OBJSDIR)/%.o: $(KMECHKRB5_BASE)/profile/%.c
429 $(COMPILE.c) $(KGSSDFLAGS) -o $$@ $<
430 $(CTFCONVERT_O)

432 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ncall/%.c
433 $(COMPILE.c) -o $$@ $<
434 $(CTFCONVERT_O)

436 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/dsw/%.c
437 $(COMPILE.c) -o $$@ $<
438 $(CTFCONVERT_O)

440 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/nsctl/%.c
441 $(COMPILE.c) -o $$@ $<
442 $(CTFCONVERT_O)

444 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/rdc/%.c
445 $(COMPILE.c) -o $$@ $<
446 $(CTFCONVERT_O)

448 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/sdbc/%.c
449 $(COMPILE.c) -o $$@ $<
450 $(CTFCONVERT_O)

452 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/solaris/%.c
453 $(COMPILE.c) -o $$@ $<
454 $(CTFCONVERT_O)

```

```

456 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/sv/%.c
457 $(COMPILE.c) -o $$@ $<
458 $(CTFCONVERT_O)

460 $(OBJSDIR)/%.o: $(UTSBASE)/common/avs/ns/unistat/%.c
461 $(COMPILE.c) -o $$@ $<
462 $(CTFCONVERT_O)

464 $(OBJSDIR)/%.o: $(UTSBASE)/common/idmap/%.c
465 $(COMPILE.c) -o $$@ $<
466 $(CTFCONVERT_O)

468 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/%.c
469 $(COMPILE.c) -o $$@ $<
470 $(CTFCONVERT_O)

472 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/arp/%.c
473 $(COMPILE.c) -o $$@ $<
474 $(CTFCONVERT_O)

476 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/ip/%.c
477 $(COMPILE.c) -o $$@ $<
478 $(CTFCONVERT_O)

480 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/ipnet/%.c
481 $(COMPILE.c) -o $$@ $<
482 $(CTFCONVERT_O)

484 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/iptun/%.c
485 $(COMPILE.c) -o $$@ $<
486 $(CTFCONVERT_O)

488 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/kssl/%.c
489 $(COMPILE.c) -o $$@ $<
490 $(CTFCONVERT_O)

492 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/sctp/%.c
493 $(COMPILE.c) -o $$@ $<
494 $(CTFCONVERT_O)

496 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/tcp/%.c
497 $(COMPILE.c) -o $$@ $<
498 $(CTFCONVERT_O)

500 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/ilb/%.c
501 $(COMPILE.c) -o $$@ $<
502 $(CTFCONVERT_O)

504 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/ipf/%.c
505 $(COMPILE.c) -o $$@ $<
506 $(CTFCONVERT_O)

508 $(OBJSDIR)/%.o: $(COMMONBASE)/net/patricia/%.c
509 $(COMPILE.c) -o $$@ $<
510 $(CTFCONVERT_O)

512 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/udp/%.c
513 $(COMPILE.c) -o $$@ $<
514 $(CTFCONVERT_O)

516 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/nca/%.c
517 $(COMPILE.c) -o $$@ $<
518 $(CTFCONVERT_O)

520 $(OBJSDIR)/%.o: $(UTSBASE)/common/inet/sockmods/%.c

```

```

521     $(COMPILE.c) -o $@ $<
522     $(CTFCONVERT_O)

524 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/inet/dlpistub/%.c
525     $(COMPILE.c) -o $@ $<
526     $(CTFCONVERT_O)

528 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/%.c
529     $(COMPILE.c) -o $@ $<
530     $(CTFCONVERT_O)

532 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/l394/%.c
533     $(COMPILE.c) -o $@ $<
534     $(CTFCONVERT_O)

536 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/l394/adapters/%.c
537     $(COMPILE.c) -o $@ $<
538     $(CTFCONVERT_O)

540 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/l394/targets/avl394/%.c
541     $(COMPILE.c) -o $@ $<
542     $(CTFCONVERT_O)

544 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/l394/targets/dcaml394/%.c
545     $(COMPILE.c) -o $@ $<
546     $(CTFCONVERT_O)

548 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/l394/targets/scsal394/%.c
549     $(COMPILE.c) -o $@ $<
550     $(CTFCONVERT_O)

552 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/sbp2/%.c
553     $(COMPILE.c) -o $@ $<
554     $(CTFCONVERT_O)

556 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/aac/%.c
557     $(COMPILE.c) -o $@ $<
558     $(CTFCONVERT_O)

560 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/afe/%.c
561     $(COMPILE.c) -o $@ $<
562     $(CTFCONVERT_O)

564 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/atge/%.c
565     $(COMPILE.c) -o $@ $<
566     $(CTFCONVERT_O)

568 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/arn/%.c
569     $(COMPILE.c) -o $@ $<
570     $(CTFCONVERT_O)

572 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/ath/%.c
573     $(COMPILE.c) -o $@ $<
574     $(CTFCONVERT_O)

576 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/atu/%.c
577     $(COMPILE.c) -o $@ $<
578     $(CTFCONVERT_O)

580 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/impl/%.c
581     $(COMPILE.c) -o $@ $<
582     $(CTFCONVERT_O)

584 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/ac97/%.c
585     $(COMPILE.c) -o $@ $<
586     $(CTFCONVERT_O)

```

```

588 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioens/%.c
589     $(COMPILE.c) -o $@ $<
590     $(CTFCONVERT_O)

592 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
593     $(COMPILE.c) -o $@ $<
594     $(CTFCONVERT_O)

596 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio1575/%.c
597     $(COMPILE.c) -o $@ $<
598     $(CTFCONVERT_O)

600 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audio810/%.c
601     $(COMPILE.c) -o $@ $<
602     $(CTFCONVERT_O)

604 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
605     $(COMPILE.c) -o $@ $<
606     $(CTFCONVERT_O)

608 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
609     $(COMPILE.c) -o $@ $<
610     $(CTFCONVERT_O)

612 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiohd/%.c
613     $(COMPILE.c) -o $@ $<
614     $(CTFCONVERT_O)

616 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audioixp/%.c
617     $(COMPILE.c) -o $@ $<
618     $(CTFCONVERT_O)

620 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiols/%.c
621     $(COMPILE.c) -o $@ $<
622     $(CTFCONVERT_O)

624 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiopci/%.c
625     $(COMPILE.c) -o $@ $<
626     $(CTFCONVERT_O)

628 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiop16x/%.c
629     $(COMPILE.c) -o $@ $<
630     $(CTFCONVERT_O)

632 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
633     $(COMPILE.c) -o $@ $<
634     $(CTFCONVERT_O)

636 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiotots/%.c
637     $(COMPILE.c) -o $@ $<
638     $(CTFCONVERT_O)

640 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
641     $(COMPILE.c) -o $@ $<
642     $(CTFCONVERT_O)

644 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
645     $(COMPILE.c) -o $@ $<
646     $(CTFCONVERT_O)

648 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bfe/%.c
649     $(COMPILE.c) -o $@ $<
650     $(CTFCONVERT_O)

652 $(OBJS_DIR)/%.o:                $(UTSBASE)/common/io/bge/%.c

```

```

653 $(COMPILE.c) -o $$@ $<
654 $(CTFCONVERT_O)

656 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/blkdev/%.c
657 $(COMPILE.c) -o $$@ $<
658 $(CTFCONVERT_O)

660 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/bpf/%.c
661 $(COMPILE.c) -o $$@ $<
662 $(CTFCONVERT_O)

664 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/cardbus/%.c
665 $(COMPILE.c) -o $$@ $<
666 $(CTFCONVERT_O)

668 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/stmf/%.c
669 $(COMPILE.c) -o $$@ $<
670 $(CTFCONVERT_O)

672 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/fct/%.c
673 $(COMPILE.c) -o $$@ $<
674 $(CTFCONVERT_O)

676 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/qlt/%.c
677 $(COMPILE.c) -o $$@ $<
678 $(CTFCONVERT_O)

680 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/srpt/%.c
681 $(COMPILE.c) -o $$@ $<
682 $(CTFCONVERT_O)

684 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/fcoet/%.c
685 $(COMPILE.c) -o $$@ $<
686 $(CTFCONVERT_O)

688 $(OBJSDIR)/%.o: $(COMMONBASE)/iscsit/%.c
689 $(COMPILE.c) -o $$@ $<
690 $(CTFCONVERT_O)

692 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/iscsit/%.c
693 $(COMPILE.c) -o $$@ $<
694 $(CTFCONVERT_O)

696 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/port/pppt/%.c
697 $(COMPILE.c) -o $$@ $<
698 $(CTFCONVERT_O)

700 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
701 $(COMPILE.c) -o $$@ $<
702 $(CTFCONVERT_O)

704 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dld/%.c
705 $(COMPILE.c) -o $$@ $<
706 $(CTFCONVERT_O)

708 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dls/%.c
709 $(COMPILE.c) -o $$@ $<
710 $(CTFCONVERT_O)

712 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/dmfe/%.c
713 $(COMPILE.c) -o $$@ $<
714 $(CTFCONVERT_O)

716 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/drm/%.c
717 $(COMPILE.c) -o $$@ $<
718 $(CTFCONVERT_O)

```

```

720 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/efe/%.c
721 $(COMPILE.c) -o $$@ $<
722 $(CTFCONVERT_O)

724 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/elxl/%.c
725 $(COMPILE.c) -o $$@ $<
726 $(CTFCONVERT_O)

728 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/fcoe/%.c
729 $(COMPILE.c) -o $$@ $<
730 $(CTFCONVERT_O)

732 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hme/%.c
733 $(COMPILE.c) -o $$@ $<
734 $(CTFCONVERT_O)

736 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pciex/%.c
737 $(COMPILE.c) -o $$@ $<
738 $(CTFCONVERT_O)

740 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
741 $(COMPILE.c) -o $$@ $<
742 $(CTFCONVERT_O)

744 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/pciex/hotplug/%.c
745 $(COMPILE.c) -o $$@ $<
746 $(CTFCONVERT_O)

748 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/hotplug/pcihp/%.c
749 $(COMPILE.c) -o $$@ $<
750 $(CTFCONVERT_O)

752 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/rds/%.c
753 $(COMPILE.c) -o $$@ $<
754 $(CTFCONVERT_O)

756 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
757 $(COMPILE.c) -o $$@ $<
758 $(CTFCONVERT_O)

760 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/iser/%.c
761 $(COMPILE.c) -o $$@ $<
762 $(CTFCONVERT_O)

764 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/ibd/%.c
765 $(COMPILE.c) -o $$@ $<
766 $(CTFCONVERT_O)

768 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/eoib/%.c
769 $(COMPILE.c) -o $$@ $<
770 $(CTFCONVERT_O)

772 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
773 $(COMPILE.c) -o $$@ $<
774 $(CTFCONVERT_O)

776 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
777 $(COMPILE.c) -o $$@ $<
778 $(CTFCONVERT_O)

780 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
781 $(COMPILE.c) -o $$@ $<
782 $(CTFCONVERT_O)

784 $(OBJSDIR)/%.o: $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.

```

```

785     $(COMPILE.c) -o $@ $<
786     $(CTFCONVERT_O)

788 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/clients/sdp/%.c
789     $(COMPILE.c) -o $@ $<
790     $(CTFCONVERT_O)

792 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
793     $(COMPILE.c) -o $@ $<
794     $(CTFCONVERT_O)

796 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
797     $(COMPILE.c) -o $@ $<
798     $(CTFCONVERT_O)

800 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
801     $(COMPILE.c) -o $@ $<
802     $(CTFCONVERT_O)

804 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
805     $(COMPILE.c) -o $@ $<
806     $(CTFCONVERT_O)

808 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/ibnex/%.c
809     $(COMPILE.c) -o $@ $<
810     $(CTFCONVERT_O)

812 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/ibt1/%.c
813     $(COMPILE.c) -o $@ $<
814     $(CTFCONVERT_O)

816 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/tavor/%.c
817     $(COMPILE.c) -o $@ $<
818     $(CTFCONVERT_O)

820 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/adapters/hermon/%.c
821     $(COMPILE.c) -o $@ $<
822     $(CTFCONVERT_O)

824 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ib/clients/daplt/%.c
825     $(COMPILE.c) -o $@ $<
826     $(CTFCONVERT_O)

828 $(OBJSDIR)/%.o:                $(COMMONBASE)/iscsi/%.c
829     $(COMPILE.c) -o $@ $<
830     $(CTFCONVERT_O)

832 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/idm/%.c
833     $(COMPILE.c) -o $@ $<
834     $(CTFCONVERT_O)

836 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ipw/%.c
837     $(COMPILE.c) -o $@ $<
838     $(CTFCONVERT_O)

840 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/iwh/%.c
841     $(COMPILE.c) -o $@ $<
842     $(CTFCONVERT_O)

844 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/iwi/%.c
845     $(COMPILE.c) -o $@ $<
846     $(CTFCONVERT_O)

848 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/iwk/%.c
849     $(COMPILE.c) -o $@ $<
850     $(CTFCONVERT_O)

```

```

852 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/iwp/%.c
853     $(COMPILE.c) -o $@ $<
854     $(CTFCONVERT_O)

856 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/kb8042/%.c
857     $(COMPILE.c) -o $@ $<
858     $(CTFCONVERT_O)

860 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/kbtrans/%.c
861     $(COMPILE.c) -o $@ $<
862     $(CTFCONVERT_O)

864 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ksocket/%.c
865     $(COMPILE.c) -o $@ $<
866     $(CTFCONVERT_O)

868 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/aggr/%.c
869     $(COMPILE.c) -o $@ $<
870     $(CTFCONVERT_O)

872 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lp/%.c
873     $(COMPILE.c) -o $@ $<
874     $(CTFCONVERT_O)

876 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/hotspares/%.c
877     $(COMPILE.c) -o $@ $<
878     $(CTFCONVERT_O)

880 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/md/%.c
881     $(COMPILE.c) -o $@ $<
882     $(CTFCONVERT_O)

884 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/mirror/%.c
885     $(COMPILE.c) -o $@ $<
886     $(CTFCONVERT_O)

888 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/notify/%.c
889     $(COMPILE.c) -o $@ $<
890     $(CTFCONVERT_O)

892 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/raid/%.c
893     $(COMPILE.c) -o $@ $<
894     $(CTFCONVERT_O)

896 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/softpart/%.c
897     $(COMPILE.c) -o $@ $<
898     $(CTFCONVERT_O)

900 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/stripe/%.c
901     $(COMPILE.c) -o $@ $<
902     $(CTFCONVERT_O)

904 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/lvm/trans/%.c
905     $(COMPILE.c) -o $@ $<
906     $(CTFCONVERT_O)

908 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mac/%.c
909     $(COMPILE.c) -o $@ $<
910     $(CTFCONVERT_O)

912 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mac/plugins/%.c
913     $(COMPILE.c) -o $@ $<
914     $(CTFCONVERT_O)

916 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mega_sas/%.c

```



```

917     $(COMPILE.c) -o $@ $<
918     $(CTFCONVERT_O)

920 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mii/%.c
921     $(COMPILE.c) -o $@ $<
922     $(CTFCONVERT_O)

924 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mr_sas/%.c
925     $(COMPILE.c) -o $@ $<
926     $(CTFCONVERT_O)

928 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
929     $(COMPILE.c) -o $@ $<
930     $(CTFCONVERT_O)

932 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mxfe/%.c
933     $(COMPILE.c) -o $@ $<
934     $(CTFCONVERT_O)

936 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mwl/%.c
937     $(COMPILE.c) -o $@ $<
938     $(CTFCONVERT_O)

940 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/mwl/mwl_fw/%.c
941     $(COMPILE.c) -o $@ $<
942     $(CTFCONVERT_O)

944 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/net80211/%.c
945     $(COMPILE.c) -o $@ $<
946     $(CTFCONVERT_O)

948 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/nge/%.c
949     $(COMPILE.c) -o $@ $<
950     $(CTFCONVERT_O)

952 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/nxge/%.c
953     $(COMPILE.c) -o $@ $<
954     $(CTFCONVERT_O)

956 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/nxge/np1/%.c
957     $(COMPILE.c) -o $@ $<
958     $(CTFCONVERT_O)

960 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/nxge/%.s
961     $(COMPILE.s) -o $@ $<

963 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/pci-ide/%.c
964     $(COMPILE.c) -o $@ $<
965     $(CTFCONVERT_O)

967 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/pcmcia/%.c
968     $(COMPILE.c) -o $@ $<
969     $(CTFCONVERT_O)

971 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/pcan/%.c
972     $(COMPILE.c) -o $@ $<
973     $(CTFCONVERT_O)

975 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/pcn/%.c
976     $(COMPILE.c) -o $@ $<
977     $(CTFCONVERT_O)

979 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/pcwl/%.c
980     $(COMPILE.c) -o $@ $<
981     $(CTFCONVERT_O)

```

```

983 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ppp/sppp/%.c
984     $(COMPILE.c) -o $@ $<
985     $(CTFCONVERT_O)

987 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ppp/spppasyn/%.c
988     $(COMPILE.c) -o $@ $<
989     $(CTFCONVERT_O)

991 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ppp/sppptun/%.c
992     $(COMPILE.c) -o $@ $<
993     $(CTFCONVERT_O)

995 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/ral/%.c
996     $(COMPILE.c) -o $@ $<
997     $(CTFCONVERT_O)

999 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rge/%.c
1000    $(COMPILE.c) -o $@ $<
1001    $(CTFCONVERT_O)

1003 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rtls/%.c
1004    $(COMPILE.c) -o $@ $<
1005    $(CTFCONVERT_O)

1007 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rsm/%.c
1008    $(COMPILE.c) -o $@ $<
1009    $(CTFCONVERT_O)

1011 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rtw/%.c
1012    $(COMPILE.c) -o $@ $<
1013    $(CTFCONVERT_O)

1015 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rum/%.c
1016    $(COMPILE.c) -o $@ $<
1017    $(CTFCONVERT_O)

1019 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rwd/%.c
1020    $(COMPILE.c) -o $@ $<
1021    $(CTFCONVERT_O)

1023 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/rwn/%.c
1024    $(COMPILE.c) -o $@ $<
1025    $(CTFCONVERT_O)

1027 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/ahci/%.c
1028    $(COMPILE.c) -o $@ $<
1029    $(CTFCONVERT_O)

1031 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c
1032    $(COMPILE.c) -o $@ $<
1033    $(CTFCONVERT_O)

1035 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/sata/adapters/si3124/%.c
1036    $(COMPILE.c) -o $@ $<
1037    $(CTFCONVERT_O)

1039 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/sata/impl/%.c
1040    $(COMPILE.c) -o $@ $<
1041    $(CTFCONVERT_O)

1043 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/scsi/conf/%.c
1044    $(COMPILE.c) -o $@ $<
1045    $(CTFCONVERT_O)

1047 $(OBJSDIR)/%.o:                $(UTSBASE)/common/io/scsi/impl/%.c
1048    $(COMPILE.c) -o $@ $<

```

```

1049      $(CTFCONVERT_O)

1051 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/scsi/targets/%.c
1052     $(COMPILE.c) -o $@ $<
1053     $(CTFCONVERT_O)

1055 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/scsi/adapters/%.c
1056     $(COMPILE.c) -o $@ $<
1057     $(CTFCONVERT_O)

1059 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
1060     $(COMPILE.c) -o $@ $<
1061     $(CTFCONVERT_O)

1063 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
1064     $(COMPILE.c) -o $@ $<
1065     $(CTFCONVERT_O)

1067 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
1068     $(COMPILE.c) -o $@ $<
1069     $(CTFCONVERT_O)

1071 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/ulp/%.c
1072     $(COMPILE.c) -o $@ $<
1073     $(CTFCONVERT_O)

1075 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/impl/%.c
1076     $(COMPILE.c) -o $@ $<
1077     $(CTFCONVERT_O)

1079 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
1080     $(COMPILE.c) -o $@ $<
1081     $(CTFCONVERT_O)

1083 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
1084     $(COMPILE.c) -o $@ $<
1085     $(CTFCONVERT_O)

1087 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
1088     $(COMPILE.c) -o $@ $<
1089     $(CTFCONVERT_O)

1091 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/fca/ocf/%.c
1092     $(COMPILE.c) -o $@ $<
1093     $(CTFCONVERT_O)

1095 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
1096     $(COMPILE.c) -o $@ $<
1097     $(CTFCONVERT_O)

1099 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
1100     $(COMPILE.c) -o $@ $<
1101     $(CTFCONVERT_O)

1103 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/sdcard/impl/%.c
1104     $(COMPILE.c) -o $@ $<
1105     $(CTFCONVERT_O)

1107 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
1108     $(COMPILE.c) -o $@ $<
1109     $(CTFCONVERT_O)

1111 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/sfe/%.c
1112     $(COMPILE.c) -o $@ $<
1113     $(CTFCONVERT_O)

```

```

1115 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/simnet/%.c
1116     $(COMPILE.c) -o $@ $<
1117     $(CTFCONVERT_O)

1119 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/softmac/%.c
1120     $(COMPILE.c) -o $@ $<
1121     $(CTFCONVERT_O)

1123 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/uath/%.c
1124     $(COMPILE.c) -o $@ $<
1125     $(CTFCONVERT_O)

1127 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/uath/uath_fw/%.c
1128     $(COMPILE.c) -o $@ $<
1129     $(CTFCONVERT_O)

1131 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/ural/%.c
1132     $(COMPILE.c) -o $@ $<
1133     $(CTFCONVERT_O)

1135 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/urtw/%.c
1136     $(COMPILE.c) -o $@ $<
1137     $(CTFCONVERT_O)

1139 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
1140     $(COMPILE.c) -o $@ $<
1141     $(CTFCONVERT_O)

1143 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
1144     $(COMPILE.c) -o $@ $<
1145     $(CTFCONVERT_O)

1147 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
1148     $(COMPILE.c) -o $@ $<
1149     $(CTFCONVERT_O)

1151 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbskel/%.c
1152     $(COMPILE.c) -o $@ $<
1153     $(CTFCONVERT_O)

1155 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
1156     $(COMPILE.c) -o $@ $<
1157     $(CTFCONVERT_O)

1159 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/hwarc/%.c
1160     $(COMPILE.c) -o $@ $<
1161     $(CTFCONVERT_O)

1163 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/hid/%.c
1164     $(COMPILE.c) -o $@ $<
1165     $(CTFCONVERT_O)

1167 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/hidparser/%.c
1168     $(COMPILE.c) -o $@ $<
1169     $(CTFCONVERT_O)

1171 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/printer/%.c
1172     $(COMPILE.c) -o $@ $<
1173     $(CTFCONVERT_O)

1175 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
1176     $(COMPILE.c) -o $@ $<
1177     $(CTFCONVERT_O)

1179 $(OBJS_DIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usblms/%.c
1180     $(COMPILE.c) -o $@ $<

```

```

1181      $(CTFCONVERT_O)

1183 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
1184   $(COMPILE.c) -o $@ $<
1185   $(CTFCONVERT_O)

1187 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/ugen/%.c
1188   $(COMPILE.c) -o $@ $<
1189   $(CTFCONVERT_O)

1191 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbser/%.c
1192   $(COMPILE.c) -o $@ $<
1193   $(CTFCONVERT_O)

1195 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
1196   $(COMPILE.c) -o $@ $<
1197   $(CTFCONVERT_O)

1199 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
1200   $(COMPILE.c) -o $@ $<
1201   $(CTFCONVERT_O)

1203 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbser/usbser_k
1204   $(COMPILE.c) -o $@ $<
1205   $(CTFCONVERT_O)

1207 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
1208   $(COMPILE.c) -o $@ $<
1209   $(CTFCONVERT_O)

1211 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
1212   $(COMPILE.c) -o $@ $<
1213   $(CTFCONVERT_O)

1215 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/hwal480_fw/%.c
1216   $(COMPILE.c) -o $@ $<
1217   $(CTFCONVERT_O)

1219 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
1220   $(COMPILE.c) -o $@ $<
1221   $(CTFCONVERT_O)

1223 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/clients/usbecm/%.c
1224   $(COMPILE.c) -o $@ $<
1225   $(CTFCONVERT_O)

1227 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/hcd/openhci/%.c
1228   $(COMPILE.c) -o $@ $<
1229   $(CTFCONVERT_O)

1231 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/hcd/ehci/%.c
1232   $(COMPILE.c) -o $@ $<
1233   $(CTFCONVERT_O)

1235 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/hcd/uhci/%.c
1236   $(COMPILE.c) -I../..common -o $@ $<
1237   $(CTFCONVERT_O)

1239 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/hubd/%.c
1240   $(COMPILE.c) -o $@ $<
1241   $(CTFCONVERT_O)

1243 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/scsa2usb/%.c
1244   $(COMPILE.c) -o $@ $<
1245   $(CTFCONVERT_O)

```

```

1247 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/usb_mid/%.c
1248   $(COMPILE.c) -o $@ $<
1249   $(CTFCONVERT_O)

1251 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/usb_ia/%.c
1252   $(COMPILE.c) -o $@ $<
1253   $(CTFCONVERT_O)

1255 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/usba/%.c
1256   $(COMPILE.c) -o $@ $<
1257   $(CTFCONVERT_O)

1259 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/usba10/%.c
1260   $(COMPILE.c) -o $@ $<
1261   $(CTFCONVERT_O)

1263 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/usb/hwa/hwahc/%.c
1264   $(COMPILE.c) -o $@ $<
1265   $(CTFCONVERT_O)

1267 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/uwb/uwba/%.c
1268   $(COMPILE.c) -o $@ $<
1269   $(CTFCONVERT_O)

1271 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/vuidmice/%.c
1272   $(COMPILE.c) -o $@ $<
1273   $(CTFCONVERT_O)

1275 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/vnic/%.c
1276   $(COMPILE.c) -o $@ $<
1277   $(CTFCONVERT_O)

1279 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/wpi/%.c
1280   $(COMPILE.c) -o $@ $<
1281   $(CTFCONVERT_O)

1283 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/zyd/%.c
1284   $(COMPILE.c) -o $@ $<
1285   $(CTFCONVERT_O)

1287 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/chxge/com/%.c
1288   $(COMPILE.c) -o $@ $<
1289   $(CTFCONVERT_O)

1291 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/chxge/%.c
1292   $(COMPILE.c) -o $@ $<
1293   $(CTFCONVERT_O)

1295 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/ixgb/%.c
1296   $(COMPILE.c) -o $@ $<
1297   $(CTFCONVERT_O)

1299 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/xge/drv/%.c
1300   $(COMPILE.c) -o $@ $<
1301   $(CTFCONVERT_O)

1303 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/xge/hal/xgehal/%.c
1304   $(COMPILE.c) -o $@ $<
1305   $(CTFCONVERT_O)

1307 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/e1000g/%.c
1308   $(COMPILE.c) -o $@ $<
1309   $(CTFCONVERT_O)

1311 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/igb/%.c
1312   $(COMPILE.c) -o $@ $<

```

```

1313      $(CTFCONVERT_O)

1315 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/iprb/%.c
1316     $(COMPILE.c) -o $@ $<
1317     $(CTFCONVERT_O)

1319 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/ixgbe/%.c
1320     $(COMPILE.c) -o $@ $<
1321     $(CTFCONVERT_O)

1323 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/ntxn/%.c
1324     $(COMPILE.c) -o $@ $<
1325     $(CTFCONVERT_O)

1327 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/myril0ge/drv/%.c
1328     $(COMPILE.c) -o $@ $<
1329     $(CTFCONVERT_O)

1331 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/%.c
1332     $(COMPILE.c) -o $@ $<
1333     $(CTFCONVERT_O)

1335 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/ipgpc/%.c
1336     $(COMPILE.c) -o $@ $<
1337     $(CTFCONVERT_O)

1339 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/dlcosmk/%.c
1340     $(COMPILE.c) -o $@ $<
1341     $(CTFCONVERT_O)

1343 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/flowacct/%.c
1344     $(COMPILE.c) -o $@ $<
1345     $(CTFCONVERT_O)

1347 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/dscpmk/%.c
1348     $(COMPILE.c) -o $@ $<
1349     $(CTFCONVERT_O)

1351 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ipp/meters/%.c
1352     $(COMPILE.c) -o $@ $<
1353     $(CTFCONVERT_O)

1355 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kiconv/kiconv_emea/%.c
1356     $(COMPILE.c) -o $@ $<
1357     $(CTFCONVERT_O)

1359 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kiconv/kiconv_ja/%.c
1360     $(COMPILE.c) -o $@ $<
1361     $(CTFCONVERT_O)

1363 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kiconv/kiconv_ko/%.c
1364     $(COMPILE.c) -o $@ $<
1365     $(CTFCONVERT_O)

1367 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kiconv/kiconv_sc/%.c
1368     $(COMPILE.c) -o $@ $<
1369     $(CTFCONVERT_O)

1371 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kiconv/kiconv_tc/%.c
1372     $(COMPILE.c) -o $@ $<
1373     $(CTFCONVERT_O)

1375 $(OBJSDIR)/%.o:      $(UTSBASE)/common/kmdb/%.c
1376     $(COMPILE.c) -o $@ $<
1377     $(CTFCONVERT_O)

```

```

1379 $(OBJSDIR)/%.o:      $(UTSBASE)/common/ktli/%.c
1380     $(COMPILE.c) -o $@ $<
1381     $(CTFCONVERT_O)

1383 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
1384     $(COMPILE.c) -o $@ $<
1385     $(CTFCONVERT_O)

1387 $(OBJSDIR)/%.o:      $(COMMONBASE)/iscsi/%.c
1388     $(COMPILE.c) -o $@ $<
1389     $(CTFCONVERT_O)

1391 $(OBJSDIR)/%.o:      $(UTSBASE)/common/inet/kifconf/%.c
1392     $(COMPILE.c) -o $@ $<
1393     $(CTFCONVERT_O)

1395 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/vr/%.c
1396     $(COMPILE.c) -o $@ $<
1397     $(CTFCONVERT_O)

1399 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/yge/%.c
1400     $(COMPILE.c) -o $@ $<
1401     $(CTFCONVERT_O)

1403 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/virtio/%.c
1404     $(COMPILE.c) -o $@ $<
1405     $(CTFCONVERT_O)

1407 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/vioblk/%.c
1408     $(COMPILE.c) -o $@ $<
1409     $(CTFCONVERT_O)

1411 $(OBJSDIR)/%.o:      $(UTSBASE)/common/io/vioif/%.c
1412     $(COMPILE.c) -o $@ $<
1413     $(CTFCONVERT_O)
1414 #
1415 # krtld must refer to its own bzero/bcopy until the kernel is fully linked
1416 #
1417 $(OBJSDIR)/bootrd.o      := CPPFLAGS += -DKOBJ_OVERRIDES
1418 $(OBJSDIR)/doreloc.o     := CPPFLAGS += -DKOBJ_OVERRIDES
1419 $(OBJSDIR)/kobj.o        := CPPFLAGS += -DKOBJ_OVERRIDES
1420 $(OBJSDIR)/kobj_boot.o   := CPPFLAGS += -DKOBJ_OVERRIDES
1421 $(OBJSDIR)/kobj_bootflags.o := CPPFLAGS += -DKOBJ_OVERRIDES
1422 $(OBJSDIR)/kobj_convrelstr.o := CPPFLAGS += -DKOBJ_OVERRIDES
1423 $(OBJSDIR)/kobj_isa.o    := CPPFLAGS += -DKOBJ_OVERRIDES
1424 $(OBJSDIR)/kobj_kdi.o    := CPPFLAGS += -DKOBJ_OVERRIDES
1425 $(OBJSDIR)/kobj_lm.o     := CPPFLAGS += -DKOBJ_OVERRIDES
1426 $(OBJSDIR)/kobj_reloc.o  := CPPFLAGS += -DKOBJ_OVERRIDES
1427 $(OBJSDIR)/kobj_stubs.o  := CPPFLAGS += -DKOBJ_OVERRIDES
1428 $(OBJSDIR)/kobj_subr.o   := CPPFLAGS += -DKOBJ_OVERRIDES

1430 $(OBJSDIR)/%.o:      $(UTSBASE)/common/krtld/%.c
1431     $(COMPILE.c) -o $@ $<
1432     $(CTFCONVERT_O)

1434 $(OBJSDIR)/%.o:      $(COMMONBASE)/list/%.c
1435     $(COMPILE.c) -o $@ $<
1436     $(CTFCONVERT_O)

1438 $(OBJSDIR)/%.o:      $(COMMONBASE)/lvm/%.c
1439     $(COMPILE.c) -o $@ $<
1440     $(CTFCONVERT_O)

1442 $(OBJSDIR)/%.o:      $(COMMONBASE)/lzma/%.c
1443     $(COMPILE.c) -o $@ $<
1444     $(CTFCONVERT_O)

```



```

1577      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1579 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/arcfour/%.c
1580   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1582 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/blowfish/%.c
1583   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1585 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/ecc/%.c
1586   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1588 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/modes/%.c
1589   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1591 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/padding/%.c
1592   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1594 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/rng/%.c
1595   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1597 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/rsa/%.c
1598   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1600 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/bignum/%.c
1601   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1603 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/bignum/%.c
1604   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1606 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/mpi/%.c
1607   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1609 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/acl/%.c
1610   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1612 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/avl/%.c
1613   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1615 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/ucode/%.c
1616   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1618 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/brand/snl/%.c
1619   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1621 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/brand/solaris10/%.c
1622   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1624 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/c2/%.c
1625   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1627 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/conf/%.c
1628   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1630 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/contract/%.c
1631   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1633 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/cpr/%.c
1634   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1636 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ctf/%.c
1637   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1639 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/ctf/%.c
1640   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1642 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/pci/%.c

```

```

1643      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1645 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/devid/%.c
1646   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1648 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/des/%.c
1649   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1651 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/smbios/%.c
1652   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1654 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ncall/%.c
1655   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1657 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/dsw/%.c
1658   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1660 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/nsctl/%.c
1661   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1663 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/rdc/%.c
1664   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1666 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/sdbc/%.c
1667   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1669 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/solaris/%.c
1670   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1672 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/sv/%.c
1673   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1675 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/avs/ns/unistat/%.c
1676   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1678 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/des/%.c
1679   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1681 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/crypto/api/%.c
1682   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1684 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/crypto/core/%.c
1685   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1687 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/crypto/io/%.c
1688   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1690 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/crypto/spi/%.c
1691   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1693 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/disp/%.c
1694   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1696 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/dtrace/%.c
1697   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1699 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/exacct/%.c
1700   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1702 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/aout/%.c
1703   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1705 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/elf/%.c
1706   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1708 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/intp/%.c

```

```

1709      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1711 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/shbin/%.c
1712   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1714 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/exec/java/%.c
1715   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1717 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/%.c
1718   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1720 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/autofs/%.c
1721   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1723 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/cacheufs/%.c
1724   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1726 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ctfs/%.c
1727   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1729 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/doorfs/%.c
1730   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1732 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/dcfis/%.c
1733   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1735 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/devfs/%.c
1736   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1738 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/dev/%.c
1739   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1741 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/fd/%.c
1742   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1744 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/fifofs/%.c
1745   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1747 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/hsfs/%.c
1748   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1750 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/lofs/%.c
1751   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1753 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/mntfs/%.c
1754   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1756 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/namefs/%.c
1757   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1759 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/smbsrv/%.c
1760   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1762 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbsrv/%.c
1763   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1765 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/nfs/%.c
1766   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1768 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/objfs/%.c
1769   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1771 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/pcfs/%.c
1772   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1774 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/portfs/%.c

```

```

1775      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1777 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/proc/%.c
1778   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1780 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/sharefs/%.c
1781   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1783 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/smbclnt/%.c
1784   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1786 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbclnt/net smb/%.c
1787   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1789 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/smbclnt/smbfs/%.c
1790   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1792 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/sockfs/%.c
1793   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1795 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/specfs/%.c
1796   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1798 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/swapfs/%.c
1799   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1801 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/tmpfs/%.c
1802   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1804 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/udfs/%.c
1805   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1807 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ufs/%.c
1808   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1810 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/ufs_log/%.c
1811   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1813 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vscan/%.c
1814   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1816 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/zfs/%.c
1817   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1819 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/fs/zut/%.c
1820   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1822 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/xattr/%.c
1823   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1825 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/zfs/%.c
1826   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1828 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/gssapi/%.c
1829   @$(LHEAD) $(LINT.c) $(KGSSDPLAGS) $< $(LTAIL))
1831 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/gssapi/mechs/dummy/%.c
1832   @$(LHEAD) $(LINT.c) $(KGSSDPLAGS) $< $(LTAIL))
1834 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/%.c
1835   @$(LHEAD) $(LINT.c) $(KGSSDPLAGS) $< $(LTAIL))
1837 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/%.c
1838   @$(LHEAD) $(LINT.c) $(KGSSDPLAGS) $< $(LTAIL))
1840 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/des/%.c

```

```

1841      @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1843 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/dk/%.c
1844   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1846 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/os/%.c
1847   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1849 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/arcfour/%.c
1850   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1852 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/enc_provider/%.c
1853   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1855 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/hash_provider/%.c
1856   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1858 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/keyhash_provider/%.c
1859   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1861 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/raw/%.c
1862   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1864 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/crypto/old/%.c
1865   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1867 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/krb5/krb/%.c
1868   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1870 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/krb5/os/%.c
1871   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1873 $(LINTS_DIR)/%.ln:      $(KMECHKRB5_BASE)/mech/%.c
1874   @$(LHEAD) $(LINT.c) $(KGSSDFLAGS) $< $(LTAIL))
1876 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/idmap/%.c
1877   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1879 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/%.c
1880   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1882 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/sockmods/%.c
1883   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1885 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/arp/%.c
1886   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1888 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ip/%.c
1889   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1891 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ipnet/%.c
1892   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1894 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/iptun/%.c
1895   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1897 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ipf/%.c
1898   @$(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))
1900 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/kssl/%.c
1901   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1903 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/net/patricia/%.c
1904   @$(LHEAD) $(LINT.c) $(IPFFLAGS) $< $(LTAIL))
1906 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/udp/%.c

```

```

1907      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1909 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/sctp/%.c
1910   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1912 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/tcp/%.c
1913   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1915 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/ilb/%.c
1916   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1918 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/nca/%.c
1919   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1921 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/dlpistub/%.c
1922   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1924 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/%.c
1925   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1927 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/%.c
1928   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1930 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/adapters/%.c
1931   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1933 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/av1394/%.c
1934   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1936 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/dcam1394/%.c
1937   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1939 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/1394/targets/scsa1394/%.c
1940   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1942 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sbp2/%.c
1943   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1945 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/aac/%.c
1946   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1948 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/afe/%.c
1949   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1951 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/atge/%.c
1952   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1954 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/arn/%.c
1955   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1957 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ath/%.c
1958   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1960 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/atu/%.c
1961   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1963 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/impl/%.c
1964   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1966 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/ac97/%.c
1967   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1969 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audio1575/%.c
1970   @$(LHEAD) $(LINT.c) $< $(LTAIL))
1972 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audio810/%.c

```



```

1973      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1975 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiocmi/%.c
1976      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1978 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiocmihd/%.c
1979      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1981 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioens/%.c
1982      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1984 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioemu10k/%.c
1985      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1987 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiohd/%.c
1988      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1990 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audioixp/%.c
1991      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1993 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiols/%.c
1994      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1996 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiopci/%.c
1997      @$(LHEAD) $(LINT.c) $< $(LTAIL))
1999 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiopl6x/%.c
2000      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2002 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiosolo/%.c
2003      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2005 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiots/%.c
2006      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2008 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiovia823x/%.c
2009      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2011 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/audio/drv/audiovia97/%.c
2012      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2014 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bfe/%.c
2015      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2017 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bpf/%.c
2018      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2020 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/bge/%.c
2021      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2023 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/blkdev/%.c
2024      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2026 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/cardbus/%.c
2027      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2029 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/lu/stmf_sbd/%.c
2030      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2032 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/fct/%.c
2033      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2035 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/qlt/%.c
2036      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2038 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/srpt/%.c

```

```

2039      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2041 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsit/%.c
2042      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2044 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/fcoet/%.c
2045      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2047 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/iscsit/%.c
2048      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2050 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/port/pppt/%.c
2051      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2053 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/comstar/stmf/%.c
2054      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2056 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dld/%.c
2057      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2059 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dls/%.c
2060      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2062 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/dmfe/%.c
2063      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2065 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/drm/%.c
2066      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2068 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/efe/%.c
2069      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2071 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/elxl/%.c
2072      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2074 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fcoe/%.c
2075      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2077 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hme/%.c
2078      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2080 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pciex/%.c
2081      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2083 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hotplug/hpcsvc/%.c
2084      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2086 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pciex/hotplug/%.c
2087      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2089 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hotplug/pcihp/%.c
2090      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2092 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/rds/%.c
2093      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2095 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/rdsv3/%.c
2096      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2098 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/iser/%.c
2099      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2101 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/ibd/%.c
2102      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2104 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/eoib/%.c

```

```

2105      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2107 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_ofs/%.c
2108      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2110 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_ucma/%.c
2111      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2113 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_umad/%.c
2114      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2116 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/of/sol_uverbs/%.
2117      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2119 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/sdp/%.c
2120      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2122 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibcm/%.c
2123      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2125 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibdm/%.c
2126      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2128 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibdma/%.c
2129      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2131 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/mgt/ibmf/%.c
2132      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2134 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/ibnex/%.c
2135      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2137 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/ibt1/%.c
2138      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2140 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/adapters/tavor/%.c
2141      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2143 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/adapters/hermon/%.c
2144      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2146 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ib/clients/daplt/%.c
2147      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2149 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsi/%.c
2150      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2152 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/idm/%.c
2153      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2155 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ipw/%.c
2156      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2158 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwh/%.c
2159      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2161 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwi/%.c
2162      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2164 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwk/%.c
2165      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2167 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iwp/%.c
2168      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2170 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/kb8042/%.c

```

```

2171      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2173 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/kbtrans/%.c
2174      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2176 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ksocket/%.c
2177      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2179 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/aggr/%.c
2180      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2182 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lp/%.c
2183      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2185 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/hotspares/%.c
2186      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2188 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/md/%.c
2189      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2191 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/mirror/%.c
2192      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2194 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/raid/%.c
2195      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2197 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/softpart/%.c
2198      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2200 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/stripe/%.c
2201      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2203 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/notify/%.c
2204      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2206 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/lvm/trans/%.c
2207      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2209 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mac/%.c
2210      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2212 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mac/plugins/%.c
2213      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2215 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mega_sas/%.c
2216      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2218 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mii/%.c
2219      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2221 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mr_sas/%.c
2222      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2224 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/mpt_sas/%.c
2225      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2227 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mxfe/%.c
2228      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2230 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mwl/%.c
2231      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2233 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/mwl/mwl_fw/%.c
2234      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2236 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/net80211/%.c

```

```

2237      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2239 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nge/%.c
2240      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2242 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/%.c
2243      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2245 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/%.s
2246      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2248 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/nxge/mpi/%.c
2249      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2251 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pci-ide/%.c
2252      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2254 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcmcia/%.c
2255      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2257 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcan/%.c
2258      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2260 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcn/%.c
2261      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2263 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/pcwl/%.c
2264      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2266 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/sppp/%.c
2267      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2269 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/spppasyn/%.c
2270      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2272 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ppp/sppptun/%.c
2273      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2275 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ral/%.c
2276      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2278 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rge/%.c
2279      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2281 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rtls/%.c
2282      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2284 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rsm/%.c
2285      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2287 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rtw/%.c
2288      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2290 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rum/%.c
2291      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2293 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rwd/%.c
2294      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2296 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/rwn/%.c
2297      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2299 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/ahci/%.c
2300      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2302 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/nv_sata/%.c

```

```

2303      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2305 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/adapters/si3124/%.c
2306      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2308 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sata/impl/%.c
2309      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2311 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/%.c
2312      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2314 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/blk2scsa/%.c
2315      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2317 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/pmcs/%.c
2318      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2320 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/%.c
2321      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2323 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/scsi_vhci/fop
2324      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2326 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/ulp/%.c
2327      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2329 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/impl/%.c
2330      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2332 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/qlc/%.c
2333      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2335 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/qlge/%.c
2336      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2338 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/emlxs/%.c
2339      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2341 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/oce/%.c
2342      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2344 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/fibre-channel/fca/fcoei/%.c
2345      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2347 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/conf/%.c
2348      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2350 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/impl/%.c
2351      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2353 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/targets/%.c
2354      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2356 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/adapters/sdhost/%.c
2357      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2359 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/impl/%.c
2360      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2362 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sdcard/targets/sdcard/%.c
2363      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2365 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/sfe/%.c
2366      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2368 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/simnet/%.c

```

```

2369         @$(LHEAD) $(LINT.c) $< $(LTAIL))
2371 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/softmac/%.c
2372     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2374 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uath/%.c
2375     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2377 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uath/uath_fw/%.c
2378     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2380 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ural/%.c
2381     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2383 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/urtw/%.c
2384     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2386 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_ac/%.
2387     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2389 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_as/%.
2390     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2392 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/audio/usb_ah/%.
2393     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2395 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbskel/%.c
2396     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2398 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/video/usbvc/%.c
2399     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2401 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hwarc/%.c
2402     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2404 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hid/%.c
2405     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2407 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hidparser/%.c
2408     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2410 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbkbm/%.c
2411     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2413 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbms/%.c
2414     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2416 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbinput/usbwcm
2417     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2419 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/ugen/%.c
2420     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2422 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/printer/%.c
2423     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2425 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/%.c
2426     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2428 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbsacm/
2429     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2431 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbftdi/
2432     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2434 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbser_k

```

```

2435         @$(LHEAD) $(LINT.c) $< $(LTAIL))
2437 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbser/usbsprl/
2438     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2440 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/wusb_df/%.c
2441     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2443 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/hw4480_fw/%.c
2444     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2446 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/wusb_ca/%.c
2447     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2449 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/clients/usbecm/%.c
2450     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2452 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/openhci/%.c
2453     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2455 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/ehci/%.c
2456     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2458 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hcd/uhci/%.c
2459     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2461 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hubd/%.c
2462     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2464 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/scsa2usb/%.c
2465     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2467 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usb_mid/%.c
2468     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2470 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usb_ia/%.c
2471     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2473 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usba/%.c
2474     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2476 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/usba10/%.c
2477     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2479 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/uwb/uwba/%.c
2480     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2482 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/usb/hwa/hwahc/%.c
2483     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2485 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vuidmice/%.c
2486     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2488 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vnic/%.c
2489     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2491 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/wpi/%.c
2492     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2494 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/zyd/%.c
2495     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2497 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/chxge/com/%.c
2498     @$(LHEAD) $(LINT.c) $< $(LTAIL))
2500 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/chxge/%.c

```

```

2501      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2503 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ixgb/%.c
2504   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2506 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/xge/drv/%.c
2507   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2509 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/xge/hal/xgehal/%.c
2510   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2512 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/e1000g/%.c
2513   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2515 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/igb/%.c
2516   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2518 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/iprb/%.c
2519   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2521 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ixgbe/%.c
2522   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2524 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/ntxn/%.c
2525   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2527 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/myril0ge/drv/%.c
2528   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2530 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/%.c
2531   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2533 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/ipgpc/%.c
2534   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2536 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/dlcosmk/%.c
2537   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2539 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/flowacct/%.c
2540   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2542 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/dscpmk/%.c
2543   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2545 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ipp/meters/%.c
2546   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2548 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_emea/%.c
2549   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2551 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_ja/%.c
2552   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2554 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_ko/%.c
2555   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2557 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_sc/%.c
2558   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2560 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kiconv/kiconv_tc/%.c
2561   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2563 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/kmdb/%.c
2564   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2566 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/krtld/%.c

```

```

2567      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2569 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/ktli/%.c
2570   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2572 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/list/%.c
2573   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2575 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/lvm/%.c
2576   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2578 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/lzma/%.c
2579   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2581 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/md4/%.c
2582   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2584 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/md5/%.c
2585   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2587 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/net/dhcp/%.c
2588   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2590 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/nvpair/%.c
2591   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2593 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/os/%.c
2594   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2596 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/%.c
2597   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2599 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/cs/%.c
2600   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2602 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/cis/%.c
2603   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2605 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/nexus/%.c
2606   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2608 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/pcmcia/pcs/%.c
2609   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2611 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/%.c
2612   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2614 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/sec/%.c
2615   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2617 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/rpc/sec_gss/%.c
2618   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2620 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/shal/%.c
2621   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2623 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/crypto/sha2/%.c
2624   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2626 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/syscall/%.c
2627   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2629 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/tnf/%.c
2630   @$(LHEAD) $(LINT.c) $< $(LTAIL))
2632 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/tsol/%.c

```

```
2633      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2635 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/util/%.c
2636      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2638 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/unicode/%.c
2639      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2641 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/vm/%.c
2642      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2644 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/scsi/adapters/iscsi/%.c
2645      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2647 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/iscsi/%.c
2648      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2650 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/inet/kifconf/%.c
2651      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2653 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/virtio/%.c
2654      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2656 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vioblk/%.c
2657      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2659 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vioif/%.c
2660       @$(LHEAD) $(LINT.c) $< $(LTAIL))
2662 ZMODLINTFLAGS = -erroff=E_CONSTANT_CONDITION
2664 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/zmod/%.c
2665      @$(LHEAD) $(LINT.c) $(ZMODLINTFLAGS) $< $(LTAIL))
2667 $(LINTS_DIR)/zlib_obj.ln:  $(ZLIB_OBJS:%.o=$(LINTS_DIR)/%.ln) \
2668      $(UTSBASE)/common/zmod/zlib_lint.c
2669      @$(LHEAD) $(LINT.c) -C $(LINTS_DIR)/zlib_obj \
2670      $(UTSBASE)/common/zmod/zlib_lint.c $(LTAIL))
2672 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/hxge/%.c
2673      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2675 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/tpm/%.c
2676      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2678 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/tpm/%.s
2679      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2681 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/vr/%.c
2682      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2684 $(LINTS_DIR)/%.ln:      $(UTSBASE)/common/io/yge/%.c
2685      @$(LHEAD) $(LINT.c) $< $(LTAIL))
2687 $(LINTS_DIR)/%.ln:      $(COMMONBASE)/fsreparse/%.c
2688      @$(LHEAD) $(LINT.c) $< $(LTAIL))
```

```

*****
48863 Thu Mar 28 23:16:19 2013
new/usr/src/uts/common/io/vioif/vioif.c
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xffea@gmail.com
*****
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 2013 Nexenta Inc. All rights reserved.
14 */

16 /* Based on the NetBSD virtio driver by Minoura Makoto. */
17 /*
18  * Copyright (c) 2010 Minoura Makoto.
19  * All rights reserved.
20  *
21  * Redistribution and use in source and binary forms, with or without
22  * modification, are permitted provided that the following conditions
23  * are met:
24  * 1. Redistributions of source code must retain the above copyright
25  * notice, this list of conditions and the following disclaimer.
26  * 2. Redistributions in binary form must reproduce the above copyright
27  * notice, this list of conditions and the following disclaimer in the
28  * documentation and/or other materials provided with the distribution.
29  *
30  * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
31  * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
32  * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
33  * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
34  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
35  * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
36  * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
37  * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
38  * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
39  * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
40 */

42 #include <sys/types.h>
43 #include <sys/errno.h>
44 #include <sys/param.h>
45 #include <sys/stropts.h>
46 #include <sys/stream.h>
47 #include <sys/strsubr.h>
48 #include <sys/kmem.h>
49 #include <sys/conf.h>
50 #include <sys/devops.h>
51 #include <sys/ksynch.h>
52 #include <sys/stat.h>
53 #include <sys/modctl.h>
54 #include <sys/debug.h>
55 #include <sys/pci.h>
56 #include <sys/ethernet.h>

58 #define VLAN_TAGSZ 4

```

```

60 #include <sys/dlpi.h>
61 #include <sys/taskq.h>
62 #include <sys/cyclic.h>

64 #include <sys/pattr.h>
65 #include <sys/strsun.h>

67 #include <sys/random.h>
68 #include <sys/sysmacros.h>
69 #include <sys/stream.h>

71 #include <sys/mac.h>
72 #include <sys/mac_provider.h>
73 #include <sys/mac_ether.h>

75 #include "virtiovar.h"
76 #include "virtioreg.h"

78 #if !defined(__packed)
79 #define __packed __attribute__((packed))
80 #endif /* __packed */

82 /* Configuration registers */
83 #define VIRTIO_NET_CONFIG_MAC 0 /* 8bit x 6byte */
84 #define VIRTIO_NET_CONFIG_STATUS 6 /* 16bit */

86 /* Feature bits */
87 #define VIRTIO_NET_F_CSUM (1 << 0) /* Host handles pkts w/ partial csum */
88 #define VIRTIO_NET_F_GUEST_CSUM (1 << 1) /* Guest handles pkts w/ part csum */
89 #define VIRTIO_NET_F_MAC (1 << 5) /* Host has given MAC address. */
90 #define VIRTIO_NET_F_GSO (1 << 6) /* Host handles pkts w/ any GSO type */
91 #define VIRTIO_NET_F_GUEST_TSO4 (1 << 7) /* Guest can handle TSOv4 in. */
92 #define VIRTIO_NET_F_GUEST_TSO6 (1 << 8) /* Guest can handle TSOv6 in. */
93 #define VIRTIO_NET_F_GUEST_ECN (1 << 9) /* Guest can handle TSO[6] w/ ECN in */
94 #define VIRTIO_NET_F_GUEST_UFO (1 << 10) /* Guest can handle UFO in. */
95 #define VIRTIO_NET_F_HOST_TSO4 (1 << 11) /* Host can handle TSOv4 in. */
96 #define VIRTIO_NET_F_HOST_TSO6 (1 << 12) /* Host can handle TSOv6 in. */
97 #define VIRTIO_NET_F_HOST_ECN (1 << 13) /* Host can handle TSO[6] w/ ECN in */
98 #define VIRTIO_NET_F_HOST_UFO (1 << 14) /* Host can handle UFO in. */
99 #define VIRTIO_NET_F_MRG_RXBUF (1 << 15) /* Host can merge receive buffers. */
100 #define VIRTIO_NET_F_STATUS (1 << 16) /* Config.status available */
101 #define VIRTIO_NET_F_CTRL_VQ (1 << 17) /* Control channel available */
102 #define VIRTIO_NET_F_CTRL_RX (1 << 18) /* Control channel RX mode support */
103 #define VIRTIO_NET_F_CTRL_VLAN (1 << 19) /* Control channel VLAN filtering */
104 #define VIRTIO_NET_F_CTRL_RX_EXTRA (1 << 20) /* Extra RX mode control support */

106 /* Status */
107 #define VIRTIO_NET_S_LINK_UP 1

109 /* Packet header structure */
110 struct virtio_net_hdr {
111     uint8_t flags;
112     uint8_t gso_type;
113     uint16_t hdr_len;
114     uint16_t gso_size;
115     uint16_t csum_start;
116     uint16_t csum_offset;
117 };

119 #define VIRTIO_NET_HDR_F_NEEDS_CSUM 1 /* flags */
120 #define VIRTIO_NET_HDR_GSO_NONE 0 /* gso_type */
121 #define VIRTIO_NET_HDR_GSO_TCPV4 1 /* gso_type */
122 #define VIRTIO_NET_HDR_GSO_UDP 3 /* gso_type */
123 #define VIRTIO_NET_HDR_GSO_TCPV6 4 /* gso_type */
124 #define VIRTIO_NET_HDR_GSO_ECN 0x80 /* gso_type, |'ed */

```

```

127 /* Control virtqueue */
128 struct virtio_net_ctrl_cmd {
129     uint8_t class;
130     uint8_t command;
131 } __packed;

133 #define VIRTIO_NET_CTRL_RX          0
134 #define VIRTIO_NET_CTRL_RX_PROMISC 0
135 #define VIRTIO_NET_CTRL_RX_ALLMULTI 1

137 #define VIRTIO_NET_CTRL_MAC         1
138 #define VIRTIO_NET_CTRL_MAC_TABLE_SET 0

140 #define VIRTIO_NET_CTRL_VLAN        2
141 #define VIRTIO_NET_CTRL_VLAN_ADD   0
142 #define VIRTIO_NET_CTRL_VLAN_DEL   1

144 struct virtio_net_ctrl_status {
145     uint8_t ack;
146 } __packed;

148 struct virtio_net_ctrl_rx {
149     uint8_t onoff;
150 } __packed;

152 struct virtio_net_ctrl_mac_tbl {
153     uint32_t nentries;
154     uint8_t macs[][ETHERADDRL];
155 } __packed;

157 struct virtio_net_ctrl_vlan {
158     uint16_t id;
159 } __packed;

161 static int vioif_quiesce(dev_info_t *);
162 static int vioif_attach(dev_info_t *, ddi_attach_cmd_t);
163 static int vioif_detach(dev_info_t *, ddi_detach_cmd_t);

165 DDI_DEFINE_STREAM_OPS(vioif_ops,
166     nulldev,          /* identify */
167     nulldev,          /* probe */
168     vioif_attach,     /* attach */
169     vioif_detach,     /* detach */
170     nodev,            /* reset */
171     NULL,              /* cb_ops */
172     D_MP,              /* bus_ops */
173     NULL,              /* power */
174     vioif_quiesce     /* quiesce */
175 );

177 static char vioif_ident[] = "VirtIO ethernet driver";

179 /* Standard Module linkage initialization for a Streams driver */
180 extern struct mod_ops mod_driverops;

182 static struct modldrv modldrv = {
183     &mod_driverops, /* Type of module. This one is a driver */
184     vioif_ident,    /* short description */
185     &vioif_ops      /* driver specific ops */
186 };

188 static struct modlinkage modlinkage = {
189     MODREV_1,
190     {

```

```

191         (void *)&modldrv,
192         NULL,
193     },
194 };

196 ddi_device_acc_attr_t vioif_attr = {
197     DDI_DEVICE_ATTR_V0,
198     DDI_NEVERSWAP_ACC, /* virtio is always native byte order */
199     DDI_STORECACHING_OK_ACC,
200     DDI_DEFAULT_ACC
201 };

203 /*
204  * A mapping represents a binding for a single buffer that is contiguous in the
205  * virtual address space.
206  */
207 struct vioif_buf_mapping {
208     caddr_t          vbm_buf;
209     ddi_dma_handle_t vbm_dmah;
210     ddi_acc_handle_t vbm_acch;
211     ddi_dma_cookie_t vbm_dmac;
212     unsigned int     vbm_ncookies;
213 };

215 /*
216  * Rx buffers can be loaned upstream, so the code has
217  * to allocate them dynamically.
218  */
219 struct vioif_rx_buf {
220     struct vioif_softc *rb_sc;
221     frtn_t             rb_frtn;

223     struct vioif_buf_mapping rb_mapping;
224 };

226 /*
227  * Tx buffers have two mapping types. One, "inline", is pre-allocated and is
228  * used to hold the virtio_net_header. Small packets also get copied there, as
229  * it's faster than mapping them. Bigger packets get mapped using the "external"
230  * mapping array. An array is used, because a packet may consist of multiple
231  * fragments, so each fragment gets bound to an entry. According to my
232  * observations, the number of fragments does not exceed 2, but just in case,
233  * a bigger, up to VIOIF_INDIRECT_MAX - 1 array is allocated. To save resources,
234  * the dma handles are allocated lazily in the tx path.
235  */
236 struct vioif_tx_buf {
237     mblk_t          *tb_mp;

239     /* inline buffer */
240     struct vioif_buf_mapping tb_inline_mapping;

242     /* External buffers */
243     struct vioif_buf_mapping *tb_external_mapping;
244     unsigned int             tb_external_num;
245 };

247 struct vioif_softc {
248     dev_info_t          *sc_dev; /* mirrors virtio_softc->sc_dev */
249     struct virtio_softc sc_virtio;

251     mac_handle_t sc_mac_handle;
252     mac_register_t *sc_macp;

254     struct virtqueue *sc_rx_vq;
255     struct virtqueue *sc_tx_vq;
256     struct virtqueue *sc_ctrl_vq;

```



```

258     unsigned int         sc_tx_stopped:1;

260     /* Feature bits. */
261     unsigned int         sc_rx_csum:1;
262     unsigned int         sc_tx_csum:1;
263     unsigned int         sc_tx_tso4:1;

265     int                   sc_mtu;
266     uint8_t              sc_mac[ETHERADDRL];
267     /*
268     * For rx buffers, we keep a pointer array, because the buffers
269     * can be loaned upstream, and we have to repopulate the array with
270     * new members.
271     */
272     struct vioif_rx_buf   **sc_rxbufs;

274     /*
275     * For tx, we just allocate an array of buffers. The packet can
276     * either be copied into the inline buffer, or the external mapping
277     * could be used to map the packet
278     */
279     struct vioif_tx_buf   *sc_txbufs;

281     kstat_t               *sc_intrstat;
282     /*
283     * We "loan" rx buffers upstream and reuse them after they are
284     * freed. This lets us avoid allocations in the hot path.
285     */
286     kmem_cache_t          *sc_rxbuf_cache;
287     ulong_t               sc_rxloan;

289     /* Copying small packets turns out to be faster then mapping them. */
290     unsigned long         sc_rxcopy_thresh;
291     unsigned long         sc_txcopy_thresh;
292     /* Some statistic coming here */
293     uint64_t              sc_ipackets;
294     uint64_t              sc_opackets;
295     uint64_t              sc_rbytes;
296     uint64_t              sc_obytes;
297     uint64_t              sc_brdcstxmt;
298     uint64_t              sc_brdcstrcv;
299     uint64_t              sc_multixmt;
300     uint64_t              sc_multircv;
301     uint64_t              sc_norecvbuf;
302     uint64_t              sc_notxbuf;
303     uint64_t              sc_toolongerr;
304     uint64_t              sc_ierrors;
305     uint64_t              sc_oerrors;
306 };

308 #define ETHER_HEADER_LEN      sizeof (struct ether_header)

310 /* MTU + the ethernet header. */
311 #define MAX_PAYLOAD          65535
312 #define MAX_MTU              (MAX_PAYLOAD - ETHER_HEADER_LEN)
313 #define DEFAULT_MTU          ETHERMTU

315 /*
316 * Yeah, we spend 8M per device. Turns out, there is no point
317 * being smart and using merged rx buffers (VIRTIO_NET_F_MRG_RXBUF),
318 * because vhost does not support them, and we expect to be used with
319 * vhost in production environment.
320 */
321 /* The buffer keeps both the packet data and the virtio_net_header. */
322 #define VIOIF_RX_SIZE (MAX_PAYLOAD + sizeof (struct virtio_net_hdr))

```

```

324 /*
325 * We win a bit on header alignment, but the host wins a lot
326 * more on moving aligned buffers. Might need more thought.
327 */
328 #define VIOIF_IP_ALIGN 0

330 /* Maximum number of indirect descriptors, somewhat arbitrary. */
331 #define VIOIF_INDIRECT_MAX 128

333 /*
334 * We pre-allocate a reasonably large buffer to copy small packets
335 * there. Bigger packets are mapped, packets with multiple
336 * cookies are mapped as indirect buffers.
337 */
338 #define VIOIF_TX_INLINE_SIZE 2048

340 /* Native queue size for all queues */
341 #define VIOIF_RX_QLEN 0
342 #define VIOIF_TX_QLEN 0
343 #define VIOIF_CTRL_QLEN 0

345 static uchar_t vioif_broadcast[ETHERADDRL] = {
346     0xff, 0xff, 0xff, 0xff, 0xff, 0xff
347 };

349 #define VIOIF_TX_THRESH_MAX    640
350 #define VIOIF_RX_THRESH_MAX    640

352 static char vioif_txcopy_thresh[] =
353     "vioif_txcopy_thresh";
354 static char vioif_rxcopy_thresh[] =
355     "vioif_rxcopy_thresh";

357 static char *vioif_priv_props[] = {
358     vioif_txcopy_thresh,
359     vioif_rxcopy_thresh,
360     NULL
361 };

363 /* Add up to ddi? */
364 static ddi_dma_cookie_t *
365 vioif_dma_curr_cookie(ddi_dma_handle_t dmah)
366 {
367     ddi_dma_impl_t *dmah_impl = (void *) dmah;
368     ASSERT(dmah_impl->dmai_cookie);
369     return (dmah_impl->dmai_cookie);
370 }

372 static void
373 vioif_dma_reset_cookie(ddi_dma_handle_t dmah, ddi_dma_cookie_t *dmac)
374 {
375     ddi_dma_impl_t *dmah_impl = (void *) dmah;
376     dmah_impl->dmai_cookie = dmac;
377 }

379 static link_state_t
380 vioif_link_state(struct vioif_softc *sc)
381 {
382     if (sc->sc_virtio.sc_features & VIRTIO_NET_F_STATUS) {
383         if (virtio_read_device_config_2(&sc->sc_virtio,
384             VIRTIO_NET_CONFIG_STATUS) & VIRTIO_NET_S_LINK_UP) {
386             dev_err(sc->sc_dev, CE_NOTE, "Link up\n");
387             return (LINK_STATE_UP);
388         } else {

```

```

389         dev_err(sc->sc_dev, CE_NOTE, "Link down\n");
390         return (LINK_STATE_DOWN);
391     }
392 }
394 dev_err(sc->sc_dev, CE_NOTE, "Link assumed up\n");
396 return (LINK_STATE_UP);
397 }

399 static ddi_dma_attr_t vioif_inline_buf_dma_attr = {
400     DMA_ATTR_V0,          /* Version number */
401     0,                   /* low address */
402     0xFFFFFFFFFFFFFFFF, /* high address */
403     0xFFFFFFFF,         /* counter register max */
404     1,                   /* page alignment */
405     1,                   /* burst sizes: 1 - 32 */
406     1,                   /* minimum transfer size */
407     0xFFFFFFFF,         /* max transfer size */
408     0xFFFFFFFFFFFFFFFF, /* address register max */
409     1,                   /* scatter-gather capacity */
410     1,                   /* device operates on bytes */
411     0,                   /* attr flag: set to 0 */
412 };

414 static ddi_dma_attr_t vioif_mapped_buf_dma_attr = {
415     DMA_ATTR_V0,          /* Version number */
416     0,                   /* low address */
417     0xFFFFFFFFFFFFFFFF, /* high address */
418     0xFFFFFFFF,         /* counter register max */
419     1,                   /* page alignment */
420     1,                   /* burst sizes: 1 - 32 */
421     1,                   /* minimum transfer size */
422     0xFFFFFFFF,         /* max transfer size */
423     0xFFFFFFFFFFFFFFFF, /* address register max */

425     /* One entry is used for the virtio_net_hdr on the tx path */
426     VIOIF_INDIRECT_MAX - 1, /* scatter-gather capacity */
427     1,                   /* device operates on bytes */
428     0,                   /* attr flag: set to 0 */
429 };

431 static ddi_device_acc_attr_t vioif_bufattr = {
432     DDI_DEVICE_ATTR_V0,
433     DDI_NEVERSWAP_ACC,
434     DDI_STORECACHING_OK_ACC,
435     DDI_DEFAULT_ACC
436 };

438 static void
439 vioif_rx_free(caddr_t free_arg)
440 {
441     struct vioif_rx_buf *buf = (void *) free_arg;
442     struct vioif_softc *sc = buf->rb_sc;

444     kmem_cache_free(sc->sc_rxbuf_cache, buf);
445     atomic_dec_ulong(&sc->sc_rxloan);
446 }

448 static int
449 vioif_rx_construct(void *buffer, void *user_arg, int kmflags)
450 {
451     _NOTE(ARGUNUSED(kmflags));
452     struct vioif_softc *sc = user_arg;
453     struct vioif_rx_buf *buf = buffer;
454     size_t len;

```

```

456     if (ddi_dma_alloc_handle(sc->sc_dev, &vioif_mapped_buf_dma_attr,
457         DDI_DMA_SLEEP, NULL, &buf->rb_mapping.vbm_dmah)) {
458         dev_err(sc->sc_dev, CE_WARN,
459             "Can't allocate dma handle for rx buffer");
460         goto exit_handle;
461     }

463     if (ddi_dma_mem_alloc(buf->rb_mapping.vbm_dmah,
464         VIOIF_RX_SIZE + sizeof (struct virtio_net_hdr),
465         &vioif_bufattr, DDI_DMA_STREAMING, DDI_DMA_SLEEP,
466         NULL, &buf->rb_mapping.vbm_buf, &len, &buf->rb_mapping.vbm_acch)) {
467         dev_err(sc->sc_dev, CE_WARN,
468             "Can't allocate rx buffer");
469         goto exit_alloc;
470     }
471     ASSERT(len >= VIOIF_RX_SIZE);

473     if (ddi_dma_addr_bind_handle(buf->rb_mapping.vbm_dmah, NULL,
474         buf->rb_mapping.vbm_buf, len, DDI_DMA_READ | DDI_DMA_STREAMING,
475         DDI_DMA_SLEEP, NULL, &buf->rb_mapping.vbm_dmac,
476         &buf->rb_mapping.vbm_ncookies)) {
477         dev_err(sc->sc_dev, CE_WARN, "Can't bind tx buffer");
479         goto exit_bind;
480     }

482     ASSERT(buf->rb_mapping.vbm_ncookies <= VIOIF_INDIRECT_MAX);

484     buf->rb_sc = sc;
485     buf->rb_frtn.free_arg = (void *) buf;
486     buf->rb_frtn.free_func = vioif_rx_free;

488     return (0);
489 exit_bind:
490     ddi_dma_mem_free(&buf->rb_mapping.vbm_acch);
491 exit_alloc:
492     ddi_dma_free_handle(&buf->rb_mapping.vbm_dmah);
493 exit_handle:

495     return (ENOMEM);
496 }

498 static void
499 vioif_rx_destruct(void *buffer, void *user_arg)
500 {
501     _NOTE(ARGUNUSED(user_arg));
502     struct vioif_rx_buf *buf = buffer;

504     ASSERT(buf->rb_mapping.vbm_acch);
505     ASSERT(buf->rb_mapping.vbm_acch);

507     (void) ddi_dma_unbind_handle(buf->rb_mapping.vbm_dmah);
508     ddi_dma_mem_free(&buf->rb_mapping.vbm_acch);
509     ddi_dma_free_handle(&buf->rb_mapping.vbm_dmah);
510 }

512 static void
513 vioif_free_mems(struct vioif_softc *sc)
514 {
515     int i;

517     for (i = 0; i < sc->sc_tx_vq->vq_num; i++) {
518         struct vioif_tx_buf *buf = &sc->sc_txbufs[i];
519         int j;

```

```

521      /* Tear down the internal mapping. */
522
523      ASSERT(buf->tb_inline_mapping.vbm_acch);
524      ASSERT(buf->tb_inline_mapping.vbm_dmah);
525
526      (void) ddi_dma_unbind_handle(buf->tb_inline_mapping.vbm_dmah);
527      ddi_dma_mem_free(&buf->tb_inline_mapping.vbm_acch);
528      ddi_dma_free_handle(&buf->tb_inline_mapping.vbm_dmah);
529
530      /* We should not see any in-flight buffers at this point. */
531      ASSERT(!buf->tb_mp);
532
533      /* Free all the dma hdnals we allocated lazily. */
534      for (j = 0; buf->tb_external_mapping[j].vbm_dmah; j++)
535          ddi_dma_free_handle(
536              &buf->tb_external_mapping[j].vbm_dmah);
537      /* Free the external mapping array. */
538      kmem_free(buf->tb_external_mapping,
539              sizeof (struct vioif_tx_buf) * VIOIF_INDIRECT_MAX - 1);
540  }
541
542  kmem_free(sc->sc_txbufs, sizeof (struct vioif_tx_buf) *
543      sc->sc_tx_vq->vq_num);
544
545  for (i = 0; i < sc->sc_rx_vq->vq_num; i++) {
546      struct vioif_rx_buf *buf = sc->sc_rxbufs[i];
547
548      if (buf)
549          kmem_cache_free(sc->sc_rxbuf_cache, buf);
550  }
551  kmem_free(sc->sc_rxbufs, sizeof (struct vioif_rx_buf *) *
552      sc->sc_rx_vq->vq_num);
553  }
554
555  static int
556  vioif_alloc_mems(struct vioif_softc *sc)
557  {
558      int i, txqsize, rxqsize;
559      size_t len;
560      unsigned int nsegments;
561
562      txqsize = sc->sc_tx_vq->vq_num;
563      rxqsize = sc->sc_rx_vq->vq_num;
564
565      sc->sc_txbufs = kmem_zalloc(sizeof (struct vioif_tx_buf) * txqsize,
566          KM_SLEEP);
567      if (sc->sc_txbufs == NULL) {
568          dev_err(sc->sc_dev, CE_WARN,
569              "Failed to allocate the tx buffers array");
570          goto exit_txalloc;
571      }
572
573      /*
574       * We don't allocate the rx vioif_bufs, just the pointers, as
575       * rx vioif_bufs can be loaned upstream, and we don't know the
576       * total number we need.
577       */
578      sc->sc_rxbufs = kmem_zalloc(sizeof (struct vioif_rx_buf *) * rxqsize,
579          KM_SLEEP);
580      if (sc->sc_rxbufs == NULL) {
581          dev_err(sc->sc_dev, CE_WARN,
582              "Failed to allocate the rx buffers pointer array");
583          goto exit_rxalloc;
584      }
585
586      for (i = 0; i < txqsize; i++) {

```

```

587      struct vioif_tx_buf *buf = &sc->sc_txbufs[i];
588
589      /* Allocate and bind an inline mapping. */
590
591      if (ddi_dma_alloc_handle(sc->sc_dev,
592          &vioif_inline_buf_dma_attr,
593          DDI_DMA_SLEEP, NULL, &buf->tb_inline_mapping.vbm_dmah)) {
594
595          dev_err(sc->sc_dev, CE_WARN,
596              "Can't allocate dma handle for tx buffer %d", i);
597          goto exit_tx;
598      }
599
600      if (ddi_dma_mem_alloc(buf->tb_inline_mapping.vbm_dmah,
601          VIOIF_TX_INLINE_SIZE, &vioif_bufattr, DDI_DMA_STREAMING,
602          DDI_DMA_SLEEP, NULL, &buf->tb_inline_mapping.vbm_buf,
603          &len, &buf->tb_inline_mapping.vbm_acch)) {
604
605          dev_err(sc->sc_dev, CE_WARN,
606              "Can't allocate tx buffer %d", i);
607          goto exit_tx;
608      }
609      ASSERT(len >= VIOIF_TX_INLINE_SIZE);
610
611      if (ddi_dma_addr_bind_handle(buf->tb_inline_mapping.vbm_dmah,
612          NULL, buf->tb_inline_mapping.vbm_buf, len,
613          DDI_DMA_WRITE | DDI_DMA_STREAMING, DDI_DMA_SLEEP, NULL,
614          &buf->tb_inline_mapping.vbm_dmac, &nsegments)) {
615
616          dev_err(sc->sc_dev, CE_WARN,
617              "Can't bind tx buffer %d", i);
618          goto exit_tx;
619      }
620
621      /* We asked for a single segment */
622      ASSERT(nsegments == 1);
623
624      /*
625       * We allow up to VIOIF_INDIRECT_MAX - 1 external mappings.
626       * In reality, I don't expect more then 2-3 used, but who
627       * knows.
628       */
629      buf->tb_external_mapping = kmem_zalloc(
630          sizeof (struct vioif_tx_buf) * VIOIF_INDIRECT_MAX - 1,
631          KM_SLEEP);
632
633      /*
634       * The external mapping's dma handles are allocate lazily,
635       * as we don't expect most of them to be used..
636       */
637  }
638
639  return (0);
640
641  exit_tx:
642      for (i = 0; i < txqsize; i++) {
643          struct vioif_tx_buf *buf = &sc->sc_txbufs[i];
644
645          if (buf->tb_inline_mapping.vbm_dmah)
646              (void) ddi_dma_unbind_handle(
647                  buf->tb_inline_mapping.vbm_dmah);
648
649          if (buf->tb_inline_mapping.vbm_acch)
650              ddi_dma_mem_free(
651                  &buf->tb_inline_mapping.vbm_acch);

```

```

653         if (buf->tb_inline_mapping.vbm_dmah)
654             ddi_dma_free_handle(
655                 &buf->tb_inline_mapping.vbm_dmah);

657         if (buf->tb_external_mapping)
658             kmem_free(buf->tb_external_mapping,
659                 sizeof (struct vioif_tx_buf) *
660                 VIOIF_INDIRECT_MAX - 1);
661     }

663     kmem_free(sc->sc_rxbufs, sizeof (struct vioif_rx_buf) * rxqsize);

665 exit_rxalloc:
666     kmem_free(sc->sc_txbufs, sizeof (struct vioif_tx_buf) * txqsize);
667 exit_txalloc:
668     return (ENOMEM);
669 }

671 /* ARGSUSED */
672 int
673 vioif_multicst(void *arg, boolean_t add, const uint8_t *macaddr)
674 {
675     return (DDI_SUCCESS);
676 }

678 /* ARGSUSED */
679 int
680 vioif_promisc(void *arg, boolean_t on)
681 {
682     return (DDI_SUCCESS);
683 }

685 /* ARGSUSED */
686 int
687 vioif_unicst(void *arg, const uint8_t *macaddr)
688 {
689     return (DDI_FAILURE);
690 }

693 static int
694 vioif_add_rx(struct vioif_softc *sc, int kmflag)
695 {
696     struct vq_entry *ve;
697     struct vioif_rx_buf *buf;

699     ve = vq_alloc_entry(sc->sc_rx_vq);
700     if (!ve) {
701         /* Out of free descriptors - ring already full. */
702         /* would be better to update sc_norxdescavail */
703         /* but MAC does not ask for this info */
704         /* hence update sc_norecvbuf */
705         sc->sc_norecvbuf ++;
706         goto exit_vq;
707     }
708     buf = sc->sc_rxbufs[ve->qe_index];

710     if (!buf) {
711         /* First run, allocate the buffer. */
712         buf = kmem_cache_alloc(sc->sc_rxbuf_cache, kmflag);
713         sc->sc_rxbufs[ve->qe_index] = buf;
714     }

716     /* Still nothing? Bye. */
717     if (!buf) {
718         dev_err(sc->sc_dev, CE_WARN, "Can't allocate rx buffer");

```

```

719         sc->sc_norecvbuf ++;
720         goto exit_buf;
721     }

723     ASSERT(buf->rb_mapping.vbm_ncookies >= 1);

725     /*
726      * For an unknown reason, the virtio_net_hdr must be placed
727      * as a separate virtio queue entry.
728      */
729     virtio_ve_add_indirect_buf(ve, buf->rb_mapping.vbm_dmac.dmac_address,
730         sizeof (struct virtio_net_hdr), B_FALSE);

732     /* Add the rest of the first cookie. */
733     virtio_ve_add_indirect_buf(ve,
734         buf->rb_mapping.vbm_dmac.dmac_address +
735         sizeof (struct virtio_net_hdr),
736         buf->rb_mapping.vbm_dmac.dmac_size -
737         sizeof (struct virtio_net_hdr), B_FALSE);

739     /*
740      * If the buffer consists of a single cookie (unlikely for a
741      * 64-k buffer), we are done. Otherwise, add the rest of the cookies
742      * using indirect entries.
743      */
744     if (buf->rb_mapping.vbm_ncookies > 1) {
745         ddi_dma_cookie_t *first_extra_dmac;
746         ddi_dma_cookie_t dmac;
747         first_extra_dmac =
748             vioif_dma_curr_cookie(buf->rb_mapping.vbm_dmah);

750         ddi_dma_nextcookie(buf->rb_mapping.vbm_dmah, &dmac);
751         virtio_ve_add_cookie(ve, buf->rb_mapping.vbm_dmah,
752             dmac, buf->rb_mapping.vbm_ncookies - 1, B_FALSE);
753         vioif_dma_reset_cookie(buf->rb_mapping.vbm_dmah,
754             first_extra_dmac);
755     }

757     virtio_push_chain(ve, B_FALSE);

759     return (DDI_SUCCESS);

761 exit_buf:
762     vq_free_entry(sc->sc_rx_vq, ve);
763 exit_vq:
764     return (DDI_FAILURE);
765 }

767 static int
768 vioif_populate_rx(struct vioif_softc *sc, int kmflag)
769 {
770     int i = 0;
771     int ret;

773     for (;;) {
774         ret = vioif_add_rx(sc, kmflag);
775         if (ret)
776             /*
777              * We could not allocate some memory. Try to work with
778              * what we've got.
779              */
780             break;
781         i++;
782     }

784     if (i)

```

```

785         virtio_sync_vq(sc->sc_rx_vq);
787         return (i);
788     }

790 static int
791 vioif_process_rx(struct vioif_softc *sc)
792 {
793     struct vq_entry *ve;
794     struct vioif_rx_buf *buf;
795     mblk_t *mp;
796     uint32_t len;
797     int i = 0;

799     while ((ve = virtio_pull_chain(sc->sc_rx_vq, &len)) {

801         buf = sc->sc_rxbufs[ve->qe_index];
802         ASSERT(buf);

804         if (len < sizeof(struct virtio_net_hdr)) {
805             dev_err(sc->sc_dev, CE_WARN, "RX: Chain too small: %u",
806                 len - (uint32_t)sizeof(struct virtio_net_hdr));
807             sc->sc_ierrors ++;
808             virtio_free_chain(ve);
809             continue;
810         }

812         len -= sizeof(struct virtio_net_hdr);
813         /*
814          * We copy small packets that happened to fit into a single
815          * cookie and reuse the buffers. For bigger ones, we loan
816          * the buffers upstream.
817          */
818         if (len < sc->sc_rxcopy_thresh) {
819             mp = allocb(len, 0);
820             if (!mp) {
821                 cmn_err(CE_WARN, "Failed to allocate mblock!");
822                 sc->sc_norecvbuf ++;
823                 sc->sc_ierrors ++;

825                 virtio_free_chain(ve);
826                 break;
827             }

829             bcopy((char *)buf->rb_mapping.vbm_buf +
830                 sizeof(struct virtio_net_hdr), mp->b_rptr, len);
831             mp->b_wptr = mp->b_rptr + len;

833         } else {
834             mp = desballoc((unsigned char *)
835                 buf->rb_mapping.vbm_buf +
836                 sizeof(struct virtio_net_hdr) +
837                 VIOIF_IP_ALIGN, len, 0, &buf->rb_frtn);
838             if (!mp) {
839                 cmn_err(CE_WARN, "Failed to allocate mblock!");
840                 sc->sc_norecvbuf ++;
841                 sc->sc_ierrors ++;

843                 virtio_free_chain(ve);
844                 break;
845             }
846             mp->b_wptr = mp->b_rptr + len;

848             atomic_inc_ulong(&sc->sc_rxloan);
849             /*
850              * Buffer loaned, we will have to allocate a new one

```

```

851         * for this slot.
852         */
853         sc->sc_rxbufs[ve->qe_index] = NULL;
854     }
855     /* virtio-net does not provide the info if this packet */
856     /* is multicast or broadcast. So we have to check it */
857     if (mp->b_rptr[0] & 0x1) {
858         if (bcmp(mp->b_rptr, vioif_broadcast, ETHERADDRL) != 0)
859             sc->sc_multircv ++;
860         else
861             sc->sc_brdcstrcv ++;
862     }

864     sc->sc_rbytes += len;
865     sc->sc_ipackets ++;

867     virtio_free_chain(ve);
868     mac_rx(sc->sc_mac_handle, NULL, mp);
869     i++;
870 }

872     return (i);
873 }

875 static void
876 vioif_reclaim_used_tx(struct vioif_softc *sc)
877 {
878     struct vq_entry *ve;
879     struct vioif_tx_buf *buf;
880     uint32_t len;
881     mblk_t *mp;
882     int i = 0;

884     while ((ve = virtio_pull_chain(sc->sc_tx_vq, &len)) {
885         /* We don't chain descriptors for tx, so don't expect any. */
886         ASSERT(!ve->qe_next);

888         buf = &sc->sc_txbufs[ve->qe_index];
889         mp = buf->tb_mp;
890         buf->tb_mp = NULL;

892         if (mp) {
893             for (i = 0; i < buf->tb_external_num; i++)
894                 (void) ddi_dma_unbind_handle(
895                     buf->tb_external_mapping[i].vbm_dmah);
896         }

898         virtio_free_chain(ve);

900         /* External mapping used, mp was not freed in vioif_send() */
901         if (mp)
902             freemsg(mp);
903         i++;
904     }

906     if (sc->sc_tx_stopped && i) {
907         sc->sc_tx_stopped = 0;
908         mac_tx_update(sc->sc_mac_handle);
909     }
910 }

912 /* sc will be used to update stat counters. */
913 /* ARGSUSED */
914 static inline void
915 vioif_tx_inline(struct vioif_softc *sc, struct vq_entry *ve, mblk_t *mp,
916                 size_t msg_size)

```

```

917 {
918     struct vioif_tx_buf *buf;
919     buf = &sc->sc_txbufs[ve->qe_index];

921     ASSERT(buf);

923     /* Frees mp */
924     mcopymsg(mp, buf->tb_inline_mapping.vbm_buf +
925             sizeof (struct virtio_net_hdr));

927     virtio_ve_add_indirect_buf(ve,
928     buf->tb_inline_mapping.vbm_dmac.dmac_laddress +
929     sizeof (struct virtio_net_hdr), msg_size, B_TRUE);
930 }

932 static inline int
933 vioif_tx_lazy_handle_alloc(struct vioif_softc *sc, struct vioif_tx_buf *buf,
934     int i)
935 {
936     int ret = DDI_SUCCESS;

938     if (!buf->tb_external_mapping[i].vbm_dmah) {
939         ret = ddi_dma_alloc_handle(sc->sc_dev,
940             &vioif_mapped_buf_dma_attr, DDI_DMA_SLEEP, NULL,
941             &buf->tb_external_mapping[i].vbm_dmah);
942         if (ret != DDI_SUCCESS) {
943             dev_err(sc->sc_dev, CE_WARN,
944                 "Can't allocate dma handle for external tx buffer");
945         }
946     }

948     return (ret);
949 }

951 static inline int
952 vioif_tx_external(struct vioif_softc *sc, struct vq_entry *ve, mblk_t *mp,
953     size_t msg_size)
954 {
955     _NOTE(ARGUNUSED(msg_size));

957     struct vioif_tx_buf *buf;
958     mblk_t *nmp;
959     int i, j;
960     int ret = DDI_SUCCESS;

962     buf = &sc->sc_txbufs[ve->qe_index];

964     ASSERT(buf);

966     buf->tb_external_num = 0;
967     i = 0;
968     nmp = mp;

970     while (nmp) {
971         size_t len;
972         ddi_dma_cookie_t dmac;
973         unsigned int ncookies;

975         len = MBLKL(nmp);
976         /*
977          * For some reason, the network stack can
978          * actually send us zero-length fragments.
979          */
980         if (len == 0) {
981             nmp = nmp->b_cont;
982             continue;

```

```

983     }

985     ret = vioif_tx_lazy_handle_alloc(sc, buf, i);
986     if (ret != DDI_SUCCESS) {
987         sc->sc_notxbuf ++;
988         sc->sc_oerrors ++;
989         goto exit_lazy_alloc;
990     }
991     ret = ddi_dma_addr_bind_handle(
992     buf->tb_external_mapping[i].vbm_dmah, NULL,
993     (caddr_t)nmp->b_rptr, len,
994     DDI_DMA_WRITE | DDI_DMA_STREAMING,
995     DDI_DMA_SLEEP, NULL, &dmac, &ncookies);

997     if (ret != DDI_SUCCESS) {
998         sc->sc_oerrors ++;
999         dev_err(sc->sc_dev, CE_NOTE,
1000             "TX: Failed to bind external handle");
1001         goto exit_bind;
1002     }

1004     /* Check if we still fit into the indirect table. */
1005     if (virtio_ve_indirect_available(ve) < ncookies) {
1006         dev_err(sc->sc_dev, CE_NOTE,
1007             "TX: Indirect descriptor table limit reached."
1008             " It took %d fragments.", i);
1009         sc->sc_notxbuf ++;
1010         sc->sc_oerrors ++;

1012         ret = DDI_FAILURE;
1013         goto exit_limit;
1014     }

1016     virtio_ve_add_cookie(ve, buf->tb_external_mapping[i].vbm_dmah,
1017         dmac, ncookies, B_TRUE);

1019     nmp = nmp->b_cont;
1020     i++;
1021     }

1023     buf->tb_external_num = i;
1024     /* Save the mp to free it when the packet is sent. */
1025     buf->tb_mp = mp;

1027     return (DDI_SUCCESS);

1029 exit_limit:
1030 exit_bind:
1031 exit_lazy_alloc:

1033     for (j = 0; j < i; j++) {
1034         (void) ddi_dma_unbind_handle(
1035             buf->tb_external_mapping[j].vbm_dmah);
1036     }

1038     return (ret);
1039 }

1041 static boolean_t
1042 vioif_send(struct vioif_softc *sc, mblk_t *mp)
1043 {
1044     struct vq_entry *ve;
1045     struct vioif_tx_buf *buf;
1046     struct virtio_net_hdr *net_header = NULL;
1047     size_t msg_size = 0;
1048     uint32_t csum_start;

```

```

1049     uint32_t csum_stuff;
1050     uint32_t csum_flags;
1051     uint32_t lso_flags;
1052     uint32_t lso_mss;
1053     mblk_t *nmp;
1054     int ret;
1055     boolean_t lso_required = B_FALSE;

1057     for (nmp = mp; nmp; nmp = nmp->b_cont)
1058         msg_size += MBLKL(nmp);

1060     if (sc->sc_tx_tso4) {
1061         mac_lso_get(mp, &lso_mss, &lso_flags);
1062         lso_required = (lso_flags & HW_LSO);
1063     }
1064     if (!lso_required && msg_size > sc->sc_mtu) {
1065         dev_err(sc->sc_dev, CE_WARN, "Message too big");
1066         freemsg(mp);

1068         sc->sc_toolongerr ++;
1069         sc->sc_oerrors ++;
1070         return (B_FALSE);
1071     }
1072     ve = vq_alloc_entry(sc->sc_tx_vq);

1074     if (!ve) {
1075         sc->sc_notxbuf ++;
1076         /* Out of free descriptors - try later. */
1077         return (B_FALSE);
1078     }
1079     buf = &sc->sc_txbufs[ve->qe_index];

1081     /* Use the inline buffer of the first entry for the virtio_net_hdr. */
1082     (void) memset(buf->tb_inline_mapping.vbm_buf, 0,
1083                 sizeof (struct virtio_net_hdr));

1085     /* LINTED E_BAD_PTR_CAST_ALIGN */
1086     net_header = (struct virtio_net_hdr *)
1087         buf->tb_inline_mapping.vbm_buf;

1089     mac_hcksum_get(mp, &csum_start, &csum_stuff, NULL,
1090                  NULL, &csum_flags);

1092     /* They want us to do the TCP/UDP csum calculation. */
1093     if (csum_flags & HCK_PARTIALCKSUM) {
1094         struct ether_header *eth_header;
1095         int eth_hsize;

1097         /* Did we ask for it? */
1098         ASSERT(sc->sc_tx_csum);

1100         /* We only asked for partial csum packets. */
1101         ASSERT(!(csum_flags & HCK_IPV4_HDRCKSUM));
1102         ASSERT(!(csum_flags & HCK_FULLCKSUM));

1104         eth_header = (void *) mp->b_rptr;
1105         if (eth_header->ether_type == htons(ETHERTYPE_VLAN)) {
1106             eth_hsize = sizeof (struct ether_vlan_header);
1107         } else {
1108             eth_hsize = sizeof (struct ether_header);
1109         }
1110         net_header->flags = VIRTIO_NET_HDR_F_NEEDS_CSUM;
1111         net_header->csum_start = eth_hsize + csum_start;
1112         net_header->csum_offset = csum_stuff - csum_start;
1113     }

```

```

1115     /* setup LSO fields if required */
1116     if (lso_required) {
1117         net_header->gso_type = VIRTIO_NET_HDR_GSO_TCPV4;
1118         net_header->gso_size = (uint16_t)lso_mss;
1119     }

1121     virtio_ve_add_indirect_buf(ve,
1122                               buf->tb_inline_mapping.vbm_dmac.dmac_address,
1123                               sizeof (struct virtio_net_hdr), B_TRUE);

1125     /*
1126     * We copy small packets into the inline buffer. The bigger ones
1127     * get mapped using the mapped buffer.
1128     */
1129     if (msg_size < sc->sc_txcopy_thresh) {
1130         vioif_tx_inline(sc, ve, mp, msg_size);
1131     } else {
1132         /* statistic gets updated by vioif_tx_external when fail */
1133         ret = vioif_tx_external(sc, ve, mp, msg_size);
1134         if (ret != DDI_SUCCESS)
1135             goto exit_tx_external;
1136     }

1138     virtio_push_chain(ve, B_TRUE);
1139     /* meanwhile update the statistic */
1140     if (mp->b_rptr[0] & 0x1) {
1141         if (bcmp(mp->b_rptr, vioif_broadcast, ETHERADDRL) != 0)
1142             sc->sc_multixmt ++;
1143         else
1144             sc->sc_brdcstxmt ++;
1145     }
1146     sc->sc_opackets ++;
1147     sc->sc_obytes += msg_size;

1149     return (B_TRUE);

1151 exit_tx_external:

1153     vq_free_entry(sc->sc_tx_vq, ve);
1154     /*
1155     * vioif_tx_external can fail when the buffer does not fit into the
1156     * indirect descriptor table. Free the mp. I don't expect this ever
1157     * to happen.
1158     */
1159     freemsg(mp);

1161     return (B_TRUE);
1162 }

1164 mblk_t *
1165 vioif_tx(void *arg, mblk_t *mp)
1166 {
1167     struct vioif_softc *sc = arg;
1168     mblk_t *nmp;

1170     while (mp != NULL) {
1171         nmp = mp->b_next;
1172         mp->b_next = NULL;

1174         if (!vioif_send(sc, mp)) {
1175             sc->sc_tx_stopped = 1;
1176             mp->b_next = nmp;
1177             break;
1178         }
1179         mp = nmp;
1180     }

```



```

1313         *new_mtu);
1314         return (EINVAL);
1315     }
1317     err = mac_maxsdu_update(sc->sc_mac_handle, *new_mtu);
1318     if (err) {
1319         dev_err(sc->sc_dev, CE_WARN,
1320             "Failed to set the requested mtu (%d)",
1321             *new_mtu);
1322         return (err);
1323     }
1324     break;
1325     case MAC_PROP_PRIVATE:
1326         err = vioif_set_prop_private(sc, pr_name,
1327             pr_valsize, pr_val);
1328         if (err)
1329             return (err);
1330     default:
1331         return (ENOTSUP);
1332 }
1334     return (0);
1335 }
1337 static int
1338 vioif_get_prop_private(struct vioif_softc *sc, const char *pr_name,
1339     uint_t pr_valsize, void *pr_val)
1340 {
1341     int err = ENOTSUP;
1342     int value;
1344     if (strcmp(pr_name, vioif_txcopy_thresh) == 0) {
1346         value = sc->sc_txcopy_thresh;
1347         err = 0;
1348         goto done;
1349     }
1350     if (strcmp(pr_name, vioif_rxcopy_thresh) == 0) {
1352         value = sc->sc_rxcopy_thresh;
1353         err = 0;
1354         goto done;
1355     }
1356 done:
1357     if (err == 0) {
1358         (void) snprintf(pr_val, pr_valsize, "%d", value);
1359     }
1360     return (err);
1361 }
1363 static int
1364 vioif_getprop(void *arg, const char *pr_name, mac_prop_id_t pr_num,
1365     uint_t pr_valsize, void *pr_val)
1366 {
1367     struct vioif_softc *sc = arg;
1368     int err = ENOTSUP;
1370     switch (pr_num) {
1371     case MAC_PROP_PRIVATE:
1372         err = vioif_get_prop_private(sc, pr_name,
1373             pr_valsize, pr_val);
1374         break;
1375     default:
1376         break;
1377     }
1378     return (err);

```

```

1379 }
1381 static void
1382 vioif_propinfo(void *arg, const char *pr_name, mac_prop_id_t pr_num,
1383     mac_prop_info_handle_t prh)
1384 {
1385     struct vioif_softc *sc = arg;
1387     switch (pr_num) {
1389     case MAC_PROP_MTU:
1390         mac_prop_info_set_range_uint32(prh, ETHERMIN, MAX_MTU);
1391         break;
1393     case MAC_PROP_PRIVATE: {
1394         char valstr[64];
1395         int value;
1397         bzero(valstr, sizeof (valstr));
1398         if (strcmp(pr_name, vioif_txcopy_thresh) == 0) {
1400             value = sc->sc_txcopy_thresh;
1401         } else if (strcmp(pr_name,
1402             vioif_rxcopy_thresh) == 0) {
1403             value = sc->sc_rxcopy_thresh;
1404         } else {
1405             return;
1406         }
1407         (void) snprintf(valstr, sizeof (valstr), "%d", value);
1408     }
1409     default:
1410         break;
1411     }
1412 }
1414 static boolean_t
1415 vioif_getcapab(void *arg, mac_capab_t cap, void *cap_data)
1416 {
1417     struct vioif_softc *sc = arg;
1419     switch (cap) {
1420     case MAC_CAPAB_HCKSUM:
1421         if (sc->sc_tx_csum) {
1422             uint32_t *txflags = cap_data;
1424             *txflags = HCKSUM_INET_PARTIAL;
1425             return (B_TRUE);
1426         }
1427         return (B_FALSE);
1428     case MAC_CAPAB_LSO:
1429         if (sc->sc_tx_tso4) {
1430             mac_capab_lso_t *cap_lso = cap_data;
1432             cap_lso->lso_flags = LSO_TX_BASIC_TCP_IPV4;
1433             cap_lso->lso_basic_tcp_ipv4.lso_max = MAX_MTU;
1434             return (B_TRUE);
1435         }
1436         return (B_FALSE);
1437     default:
1438         break;
1439     }
1440     return (B_FALSE);
1441 }
1443 static mac_callbacks_t vioif_m_callbacks = {
1444     .mc_callbacks = MC_GETCAPAB,

```

```

1445     .mc_getstat    = vioif_stat,
1446     .mc_start     = vioif_start,
1447     .mc_stop      = vioif_stop,
1448     .mc_setpromisc = vioif_promisc,
1449     .mc_multicast = vioif_multicast,
1450     .mc_unicast  = vioif_unicast,
1451     .mc_tx       = vioif_tx,
1452     /* Optional callbacks */
1453     .mc_reserved = NULL,          /* reserved */
1454     .mc_ioctl   = NULL,          /* mc_ioctl */
1455     .mc_getcapab = vioif_getcapab, /* mc_getcapab */
1456     .mc_open    = NULL,          /* mc_open */
1457     .mc_close   = NULL,          /* mc_close */
1458     .mc_setprop = vioif_setprop,
1459     .mc_getprop = vioif_getprop,
1460     .mc_propinfo = vioif_propinfo,
1461 };

1463 static void
1464 vioif_show_features(struct vioif_softc *sc, const char *prefix,
1465     uint32_t features)
1466 {
1467     char buf[512];
1468     char *bufp = buf;
1469     char *bufend = buf + sizeof (buf);

1471     /* LINTED E_PTRDIFF_OVERFLOW */
1472     bufp += snprintf(bufp, bufend - bufp, prefix);

1474     /* LINTED E_PTRDIFF_OVERFLOW */
1475     bufp += virtio_show_features(features, bufp, bufend - bufp);

1477     /* LINTED E_PTRDIFF_OVERFLOW */
1478     bufp += snprintf(bufp, bufend - bufp, "Vioif ( ");

1480     if (features & VIRTIO_NET_F_CSUM)
1481         /* LINTED E_PTRDIFF_OVERFLOW */
1482         bufp += snprintf(bufp, bufend - bufp, "CSUM ");
1483     if (features & VIRTIO_NET_F_GUEST_CSUM)
1484         /* LINTED E_PTRDIFF_OVERFLOW */
1485         bufp += snprintf(bufp, bufend - bufp, "GUEST_CSUM ");
1486     if (features & VIRTIO_NET_F_MAC)
1487         /* LINTED E_PTRDIFF_OVERFLOW */
1488         bufp += snprintf(bufp, bufend - bufp, "MAC ");
1489     if (features & VIRTIO_NET_F_GSO)
1490         /* LINTED E_PTRDIFF_OVERFLOW */
1491         bufp += snprintf(bufp, bufend - bufp, "GSO ");
1492     if (features & VIRTIO_NET_F_GUEST_TSO4)
1493         /* LINTED E_PTRDIFF_OVERFLOW */
1494         bufp += snprintf(bufp, bufend - bufp, "GUEST_TSO4 ");
1495     if (features & VIRTIO_NET_F_GUEST_TSO6)
1496         /* LINTED E_PTRDIFF_OVERFLOW */
1497         bufp += snprintf(bufp, bufend - bufp, "GUEST_TSO6 ");
1498     if (features & VIRTIO_NET_F_GUEST_ECN)
1499         /* LINTED E_PTRDIFF_OVERFLOW */
1500         bufp += snprintf(bufp, bufend - bufp, "GUEST_ECN ");
1501     if (features & VIRTIO_NET_F_GUEST_UFO)
1502         /* LINTED E_PTRDIFF_OVERFLOW */
1503         bufp += snprintf(bufp, bufend - bufp, "GUEST_UFO ");
1504     if (features & VIRTIO_NET_F_HOST_TSO4)
1505         /* LINTED E_PTRDIFF_OVERFLOW */
1506         bufp += snprintf(bufp, bufend - bufp, "HOST_TSO4 ");
1507     if (features & VIRTIO_NET_F_HOST_TSO6)
1508         /* LINTED E_PTRDIFF_OVERFLOW */
1509         bufp += snprintf(bufp, bufend - bufp, "HOST_TSO6 ");
1510     if (features & VIRTIO_NET_F_HOST_ECN)

```

```

1511         /* LINTED E_PTRDIFF_OVERFLOW */
1512         bufp += snprintf(bufp, bufend - bufp, "HOST_ECN ");
1513     if (features & VIRTIO_NET_F_HOST_UFO)
1514         /* LINTED E_PTRDIFF_OVERFLOW */
1515         bufp += snprintf(bufp, bufend - bufp, "HOST_UFO ");
1516     if (features & VIRTIO_NET_F_MRG_RXBUF)
1517         /* LINTED E_PTRDIFF_OVERFLOW */
1518         bufp += snprintf(bufp, bufend - bufp, "MRG_RXBUF ");
1519     if (features & VIRTIO_NET_F_STATUS)
1520         /* LINTED E_PTRDIFF_OVERFLOW */
1521         bufp += snprintf(bufp, bufend - bufp, "STATUS ");
1522     if (features & VIRTIO_NET_F_CTRL_VQ)
1523         /* LINTED E_PTRDIFF_OVERFLOW */
1524         bufp += snprintf(bufp, bufend - bufp, "CTRL_VQ ");
1525     if (features & VIRTIO_NET_F_CTRL_RX)
1526         /* LINTED E_PTRDIFF_OVERFLOW */
1527         bufp += snprintf(bufp, bufend - bufp, "CTRL_RX ");
1528     if (features & VIRTIO_NET_F_CTRL_VLAN)
1529         /* LINTED E_PTRDIFF_OVERFLOW */
1530         bufp += snprintf(bufp, bufend - bufp, "CTRL_VLAN ");
1531     if (features & VIRTIO_NET_F_CTRL_RX_EXTRA)
1532         /* LINTED E_PTRDIFF_OVERFLOW */
1533         bufp += snprintf(bufp, bufend - bufp, "CTRL_RX_EXTRA ");

1535     /* LINTED E_PTRDIFF_OVERFLOW */
1536     bufp += snprintf(bufp, bufend - bufp, " ");
1537     *bufp = '\0';

1539     dev_err(sc->sc_dev, CE_NOTE, "%s", buf);
1540 }

1542 /*
1543  * Find out which features are supported by the device and
1544  * choose which ones we wish to use.
1545  */
1546 static int
1547 vioif_dev_features(struct vioif_softc *sc)
1548 {
1549     uint32_t host_features;

1551     host_features = virtio_negotiate_features(&sc->sc_virtio,
1552         VIRTIO_NET_F_CSUM |
1553         VIRTIO_NET_F_HOST_TSO4 |
1554         VIRTIO_NET_F_HOST_ECN |
1555         VIRTIO_NET_F_MAC |
1556         VIRTIO_NET_F_STATUS |
1557         VIRTIO_F_RING_INDIRECT_DESC |
1558         VIRTIO_F_NOTIFY_ON_EMPTY);

1560     vioif_show_features(sc, "Host features: ", host_features);
1561     vioif_show_features(sc, "Negotiated features: ",
1562         sc->sc_virtio.sc_features);

1564     if (!(sc->sc_virtio.sc_features & VIRTIO_F_RING_INDIRECT_DESC)) {
1565         dev_err(sc->sc_dev, CE_NOTE,
1566             "Host does not support RING_INDIRECT_DESC, bye.");
1567         return (DDI_FAILURE);
1568     }

1570     return (DDI_SUCCESS);
1571 }

1573 static int
1574 vioif_has_feature(struct vioif_softc *sc, uint32_t feature)
1575 {
1576     return (virtio_has_feature(&sc->sc_virtio, feature));

```

```

1577 }

1579 static void
1580 vioif_set_mac(struct vioif_softc *sc)
1581 {
1582     int i;

1584     for (i = 0; i < ETHERADDRL; i++) {
1585         virtio_write_device_config_1(&sc->sc_virtio,
1586             VIRTIO_NET_CONFIG_MAC + i, sc->sc_mac[i]);
1587     }
1588 }

1590 /* Get the mac address out of the hardware, or make up one. */
1591 static void
1592 vioif_get_mac(struct vioif_softc *sc)
1593 {
1594     int i;
1595     if (sc->sc_virtio.sc_features & VIRTIO_NET_F_MAC) {
1596         for (i = 0; i < ETHERADDRL; i++) {
1597             sc->sc_mac[i] = virtio_read_device_config_1(
1598                 &sc->sc_virtio,
1599                 VIRTIO_NET_CONFIG_MAC + i);
1600         }
1601         dev_err(sc->sc_dev, CE_NOTE, "Got MAC address from host: %s",
1602             ether_sprintf((struct ether_addr *)sc->sc_mac));
1603     } else {
1604         /* Get a few random bytes */
1605         (void) random_get_pseudo_bytes(sc->sc_mac, ETHERADDRL);
1606         /* Make sure it's a unicast MAC */
1607         sc->sc_mac[0] &= ~1;
1608         /* Set the "locally administered" bit */
1609         sc->sc_mac[1] |= 2;

1611         vioif_set_mac(sc);

1613         dev_err(sc->sc_dev, CE_NOTE,
1614             "Generated a random MAC address: %s",
1615             ether_sprintf((struct ether_addr *)sc->sc_mac));
1616     }
1617 }

1619 /*
1620 * Virtqueue interrupt handlers
1621 */
1622 /* ARGSUSED */
1623 uint_t
1624 vioif_rx_handler(caddr_t arg1, caddr_t arg2)
1625 {
1626     struct virtio_softc *vsc = (void *) arg1;
1627     struct vioif_softc *sc = container_of(vsc,
1628         struct vioif_softc, sc_virtio);

1630     (void) vioif_process_rx(sc);

1632     (void) vioif_populate_rx(sc, KM_NOSLEEP);

1634     return (DDI_INTR_CLAIMED);
1635 }

1637 /* ARGSUSED */
1638 uint_t
1639 vioif_tx_handler(caddr_t arg1, caddr_t arg2)
1640 {
1641     struct virtio_softc *vsc = (void *) arg1;
1642     struct vioif_softc *sc = container_of(vsc,

```

```

1643     struct vioif_softc, sc_virtio);

1645     vioif_reclaim_used_tx(sc);
1646     return (DDI_INTR_CLAIMED);
1647 }

1649 static int
1650 vioif_register_ints(struct vioif_softc *sc)
1651 {
1652     int ret;

1654     struct virtio_int_handler vioif_vq_h[] = {
1655         { vioif_rx_handler },
1656         { vioif_tx_handler },
1657         { NULL }
1658     };

1660     ret = virtio_register_ints(&sc->sc_virtio, NULL, vioif_vq_h);

1662     return (ret);
1663 }

1666 static void
1667 vioif_check_features(struct vioif_softc *sc)
1668 {
1669     if (vioif_has_feature(sc, VIRTIO_NET_F_CSUM)) {
1670         /* The GSO/GRO featured depend on CSUM, check them here. */
1671         sc->sc_tx_csum = 1;
1672         sc->sc_rx_csum = 1;

1674         if (!vioif_has_feature(sc, VIRTIO_NET_F_GUEST_CSUM)) {
1675             sc->sc_rx_csum = 0;
1676         }
1677         cmn_err(CE_NOTE, "Csum enabled.");

1679         if (vioif_has_feature(sc, VIRTIO_NET_F_HOST_TSO4)) {
1681             sc->sc_tx_tso4 = 1;
1682             /*
1683              * We don't seem to have a way to ask the system
1684              * not to send us LSO packets with Explicit
1685              * Congestion Notification bit set, so we require
1686              * the device to support it in order to do
1687              * LSO.
1688              */
1689             if (!vioif_has_feature(sc, VIRTIO_NET_F_HOST_ECN)) {
1690                 dev_err(sc->sc_dev, CE_NOTE,
1691                     "TSO4 supported, but not ECN. "
1692                     "Not using LSO.");
1693                 sc->sc_tx_tso4 = 0;
1694             } else {
1695                 cmn_err(CE_NOTE, "LSO enabled");
1696             }
1697         }
1698     }
1699 }

1701 static int
1702 vioif_attach(dev_info_t *devinfo, ddi_attach_cmd_t cmd)
1703 {
1704     int ret, instance;
1705     struct vioif_softc *sc;
1706     struct virtio_softc *vsc;
1707     mac_register_t *macp;

```

```

1709     instance = ddi_get_instance(devinfo);
1711     switch (cmd) {
1712     case DDI_ATTACH:
1713         break;
1715     case DDI_RESUME:
1716     case DDI_PM_RESUME:
1717         dev_err(devinfo, CE_WARN, "resume not supported yet");
1718         goto exit;
1720     default:
1721         dev_err(devinfo, CE_WARN, "cmd 0x%x unrecognized", cmd);
1722         goto exit;
1723     }
1725     sc = kmem_zalloc(sizeof (struct vioif_softc), KM_SLEEP);
1726     ddi_set_driver_private(devinfo, sc);
1728     vsc = &sc->sc_virtio;
1730     /* Duplicate for less typing */
1731     sc->sc_dev = devinfo;
1732     vsc->sc_dev = devinfo;
1734     /*
1735      * Initialize interrupt kstat.
1736      */
1737     sc->sc_intrstat = kstat_create("vioif", instance, "intr", "controller",
1738     KSTAT_TYPE_INTR, 1, 0);
1739     if (sc->sc_intrstat == NULL) {
1740         dev_err(devinfo, CE_WARN, "kstat_create failed");
1741         goto exit_intrstat;
1742     }
1743     kstat_install(sc->sc_intrstat);
1745     /* map BAR 0 */
1746     ret = ddi_regs_map_setup(devinfo, 1,
1747     (caddr_t *)&sc->sc_virtio.sc_io_addr,
1748     0, 0, &vioif_attr, &sc->sc_virtio.sc_ioh);
1749     if (ret != DDI_SUCCESS) {
1750         dev_err(devinfo, CE_WARN, "unable to map bar 0: %d", ret);
1751         goto exit_map;
1752     }
1754     virtio_device_reset(&sc->sc_virtio);
1755     virtio_set_status(&sc->sc_virtio, VIRTIO_CONFIG_DEVICE_STATUS_ACK);
1756     virtio_set_status(&sc->sc_virtio, VIRTIO_CONFIG_DEVICE_STATUS_DRIVER);
1758     ret = vioif_dev_features(sc);
1759     if (ret)
1760         goto exit_features;
1762     vsc->sc_nvqs = vioif_has_feature(sc, VIRTIO_NET_F_CTRL_VQ) ? 3 : 2;
1764     sc->sc_rxbuf_cache = kmem_cache_create("vioif_rx",
1765     sizeof (struct vioif_rx_buf), 0, vioif_rx_construct,
1766     vioif_rx_destruct, NULL, sc, NULL, KM_SLEEP);
1767     if (sc->sc_rxbuf_cache == NULL) {
1768         cmm_err(CE_NOTE, "Can't allocate the buffer cache");
1769         goto exit_cache;
1770     }
1772     ret = vioif_register_ints(sc);
1773     if (ret) {
1774         dev_err(sc->sc_dev, CE_WARN,

```

```

1775         "Failed to allocate interrupt(s)!");
1776         goto exit_ints;
1777     }
1779     /*
1780      * Register layout determined, can now access the
1781      * device-specific bits
1782      */
1783     vioif_get_mac(sc);
1785     sc->sc_rx_vq = virtio_alloc_vq(&sc->sc_virtio, 0,
1786     VIOIF_RX_QLEN, VIOIF_INDIRECT_MAX, "rx");
1787     if (!sc->sc_rx_vq)
1788         goto exit_alloc1;
1789     virtio_stop_vq_intr(sc->sc_rx_vq);
1791     sc->sc_tx_vq = virtio_alloc_vq(&sc->sc_virtio, 1,
1792     VIOIF_TX_QLEN, VIOIF_INDIRECT_MAX, "tx");
1793     if (!sc->sc_tx_vq)
1794         goto exit_alloc2;
1795     virtio_stop_vq_intr(sc->sc_tx_vq);
1797     if (vioif_has_feature(sc, VIRTIO_NET_F_CTRL_VQ)) {
1798         sc->sc_ctrl_vq = virtio_alloc_vq(&sc->sc_virtio, 2,
1799         VIOIF_CTRL_QLEN, 0, "ctrl");
1800         if (!sc->sc_ctrl_vq) {
1801             goto exit_alloc3;
1802         }
1803         virtio_stop_vq_intr(sc->sc_ctrl_vq);
1804     }
1806     virtio_set_status(&sc->sc_virtio,
1807     VIRTIO_CONFIG_DEVICE_STATUS_DRIVER_OK);
1809     sc->sc_rxloan = 0;
1811     /* set some reasonable-small default values */
1812     sc->sc_rxcopy_thresh = 300;
1813     sc->sc_txcopy_thresh = 300;
1814     sc->sc_mtu = ETHERMTU;
1816     vioif_check_features(sc);
1818     if (vioif_alloc_mems(sc))
1819         goto exit_alloc_mems;
1821     if ((macp = mac_alloc(MAC_VERSION)) == NULL) {
1822         dev_err(devinfo, CE_WARN, "Failed to allocate a mac_register");
1823         goto exit_macalloc;
1824     }
1826     macp->m_type_ident = MAC_PLUGIN_IDENT_ETHER;
1827     macp->m_driver = sc;
1828     macp->m_dip = devinfo;
1829     macp->m_src_addr = sc->sc_mac;
1830     macp->m_callbacks = &vioif_m_callbacks;
1831     macp->m_min_sdu = 0;
1832     macp->m_max_sdu = sc->sc_mtu;
1833     macp->m_margin = VLAN_TAGSZ;
1834     macp->m_priv_props = vioif_priv_props;
1836     sc->sc_macp = macp;
1838     /* Pre-fill the rx ring. */
1839     (void) vioif_populate_rx(sc, KM_SLEEP);

```

```

1841     ret = mac_register(macp, &sc->sc_mac_handle);
1842     if (ret != 0) {
1843         dev_err(devinfo, CE_WARN, "vioif_attach: "
1844             "mac_register() failed, ret=%d", ret);
1845         goto exit_register;
1846     }
1848     ret = virtio_enable_ints(&sc->sc_virtio);
1849     if (ret) {
1850         dev_err(devinfo, CE_WARN, "Failed to enable interrupts");
1851         goto exit_enable_ints;
1852     }
1854     mac_link_update(sc->sc_mac_handle, LINK_STATE_UP);
1855     return (DDI_SUCCESS);
1857 exit_enable_ints:
1858     (void) mac_unregister(sc->sc_mac_handle);
1859 exit_register:
1860     mac_free(macp);
1861 exit_macalloc:
1862     vioif_free_mems(sc);
1863 exit_alloc_mems:
1864     virtio_release_ints(&sc->sc_virtio);
1865     if (sc->sc_ctrl_vq)
1866         virtio_free_vq(sc->sc_ctrl_vq);
1867 exit_alloc3:
1868     virtio_free_vq(sc->sc_tx_vq);
1869 exit_alloc2:
1870     virtio_free_vq(sc->sc_rx_vq);
1871 exit_alloc1:
1872 exit_ints:
1873     kmem_cache_destroy(sc->sc_rxbuf_cache);
1874 exit_cache:
1875 exit_features:
1876     virtio_set_status(&sc->sc_virtio, VIRTIO_CONFIG_DEVICE_STATUS_FAILED);
1877     ddi_regs_map_free(&sc->sc_virtio.sc_iah);
1878 exit_intrstat:
1879 exit_map:
1880     kstat_delete(sc->sc_intrstat);
1881     kmem_free(sc, sizeof (struct vioif_softc));
1882 exit:
1883     return (DDI_FAILURE);
1884 }
1886 static int
1887 vioif_detach(dev_info_t *devinfo, ddi_detach_cmd_t cmd)
1888 {
1889     struct vioif_softc *sc;
1891     if ((sc = ddi_get_driver_private(devinfo)) == NULL)
1892         return (DDI_FAILURE);
1894     switch (cmd) {
1895     case DDI_DETACH:
1896         break;
1898     case DDI_PM_SUSPEND:
1899         cmn_err(CE_WARN, "suspend not supported yet");
1900         return (DDI_FAILURE);
1902     default:
1903         cmn_err(CE_WARN, "cmd 0x%x unrecognized", cmd);
1904         return (DDI_FAILURE);
1905     }

```

```

1907     if (sc->sc_rxloan) {
1908         cmn_err(CE_NOTE, "Some rx buffers are still upstream, "
1909             "Not detaching");
1910         return (DDI_FAILURE);
1911     }
1913     virtio_stop_vq_intr(sc->sc_rx_vq);
1914     virtio_stop_vq_intr(sc->sc_tx_vq);
1916     virtio_release_ints(&sc->sc_virtio);
1918     if (mac_unregister(sc->sc_mac_handle)) {
1919         return (DDI_FAILURE);
1920     }
1922     mac_free(sc->sc_macp);
1924     vioif_free_mems(sc);
1925     virtio_free_vq(sc->sc_rx_vq);
1926     virtio_free_vq(sc->sc_tx_vq);
1928     virtio_device_reset(&sc->sc_virtio);
1930     ddi_regs_map_free(&sc->sc_virtio.sc_iah);
1932     kmem_cache_destroy(sc->sc_rxbuf_cache);
1933     kstat_delete(sc->sc_intrstat);
1934     kmem_free(sc, sizeof (struct vioif_softc));
1936     return (DDI_SUCCESS);
1937 }
1939 static int
1940 vioif_quiesce(dev_info_t *devinfo)
1941 {
1942     struct vioif_softc *sc;
1944     if ((sc = ddi_get_driver_private(devinfo)) == NULL)
1945         return (DDI_FAILURE);
1947     virtio_stop_vq_intr(sc->sc_rx_vq);
1948     virtio_stop_vq_intr(sc->sc_tx_vq);
1949     virtio_device_reset(&sc->sc_virtio);
1951     return (DDI_SUCCESS);
1952 }
1954 int
1955 _init(void)
1956 {
1957     int ret = 0;
1959     mac_init_ops(&vioif_ops, "vioif");
1961     ret = mod_install(&modlinkage);
1962     if (ret != DDI_SUCCESS) {
1963         mac_fini_ops(&vioif_ops);
1964         cmn_err(CE_WARN, "Unable to install the driver");
1965         return (ret);
1966     }
1968     return (0);
1969 }
1971 int
1972 _fini(void)

```

```
1973 {  
1974     int ret;  
  
1976     ret = mod_remove(&modlinkage);  
1977     if (ret == DDI_SUCCESS) {  
1978         mac_fini_ops(&vioif_ops);  
1979     }  
  
1981     return (ret);  
1982 }  
  
1984 int  
1985 _info(struct modinfo *pModinfo)  
1986 {  
1987     return (mod_info(&modlinkage, pModinfo));  
1988 }
```

new/usr/src/uts/common/io/virtio/virtio.c

1

```
*****
34889 Thu Mar 28 23:16:19 2013
new/usr/src/uts/common/io/virtio/virtio.c
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xfee@gmail.com
*****
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 2012 Nexenta Systems, Inc.
24  * Copyright 2012 Alexey Zaytsev <alexey.zaytsev@gmail.com>
25  */
26
27 /* Based on the NetBSD virtio driver by Minoura Makoto. */
28 /*
29  * Copyright (c) 2010 Minoura Makoto.
30  * All rights reserved.
31  *
32  * Redistribution and use in source and binary forms, with or without
33  * modification, are permitted provided that the following conditions
34  * are met:
35  * 1. Redistributions of source code must retain the above copyright
36  * notice, this list of conditions and the following disclaimer.
37  * 2. Redistributions in binary form must reproduce the above copyright
38  * notice, this list of conditions and the following disclaimer in the
39  * documentation and/or other materials provided with the distribution.
40  *
41  * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
42  * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES
43  * OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.
44  * IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT,
45  * INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT
46  * NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
47  * DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
48  * THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
49  * (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF
50  * THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
51  *
52  */
53
54 #include <sys/conf.h>
55 #include <sys/kmem.h>
56 #include <sys/debug.h>
57 #include <sys/modctl.h>
58 #include <sys/autoconf.h>
```

new/usr/src/uts/common/io/virtio/virtio.c

2

```
59 #include <sys/ddi_impldefs.h>
60 #include <sys/ddi.h>
61 #include <sys/sunddi.h>
62 #include <sys/sunndi.h>
63 #include <sys/avintr.h>
64 #include <sys/spl.h>
65 #include <sys/promif.h>
66 #include <sys/list.h>
67 #include <sys/bootconf.h>
68 #include <sys/bootsvcs.h>
69 #include <sys/sysmacros.h>
70 #include <sys/pci.h>
71
72 #include "virtiovar.h"
73 #include "virtioreg.h"
74
75 #define NDEVNAMES (sizeof (virtio_device_name) / sizeof (char *))
76 #define MINSEG_INDIRECT 2 /* use indirect if nsecs >= this value */
77 #define VIRTIOEUE_ALIGN(n) (((n)+(VIRTIO_PAGE_SIZE-1)) & \
78 ~ (VIRTIO_PAGE_SIZE-1))
79
80 void
81 virtio_set_status(struct virtio_softc *sc, unsigned int status)
82 {
83     int old = 0;
84
85     if (status != 0) {
86         old = ddi_get8(sc->sc_ioh, (uint8_t *) (sc->sc_io_addr +
87         if (status != 0)
88             old = ddi_get8(sc->sc_ioh,
89                 (uint8_t *) (sc->sc_io_addr +
90                 VIRTIO_CONFIG_DEVICE_STATUS));
91     }
92
93     ddi_put8(sc->sc_ioh, (uint8_t *) (sc->sc_io_addr +
94         VIRTIO_CONFIG_DEVICE_STATUS), status | old);
95     ddi_put8(sc->sc_ioh,
96         (uint8_t *) (sc->sc_io_addr + VIRTIO_CONFIG_DEVICE_STATUS),
97         status | old);
98 }
99
100 /*
101  * Negotiate features, save the result in sc->sc_features
102  */
103 uint32_t
104 virtio_negotiate_features(struct virtio_softc *sc, uint32_t guest_features)
105 {
106     uint32_t host_features;
107     uint32_t features;
108
109     host_features = ddi_get32(sc->sc_ioh,
110         /* LINTED E_BAD_PTR_CAST_ALIGN */
111         (uint32_t *) (sc->sc_io_addr + VIRTIO_CONFIG_DEVICE_FEATURES));
112
113     dev_debug(sc->sc_dev, CE_NOTE, "host features: %x, guest features: %x",
114         dev_debug(sc->sc_dev, CE_NOTE,
115             "host features: %x, guest features: %x",
116             host_features, guest_features);
117
118     features = host_features & guest_features;
119     ddi_put32(sc->sc_ioh,
120         /* LINTED E_BAD_PTR_CAST_ALIGN */
121         (uint32_t *) (sc->sc_io_addr + VIRTIO_CONFIG_GUEST_FEATURES),
122         features);
123
124     sc->sc_features = features;
125 }
```

```

118     return (host_features);
119 }

121 size_t
122 virtio_show_features(uint32_t features, char *buf, size_t len)
123 virtio_show_features(uint32_t features,
124     char *buf, size_t len)
123 {
124     char *orig_buf = buf;
125     char *bufend = buf + len;

127     /* LINTED E_PTRDIFF_OVERFLOW */
128     buf += snprintf(buf, bufend - buf, "Generic ( ");
129     if (features & VIRTIO_F_RING_INDIRECT_DESC)
130         /* LINTED E_PTRDIFF_OVERFLOW */
131         buf += snprintf(buf, bufend - buf, "INDIRECT_DESC ");

133     /* LINTED E_PTRDIFF_OVERFLOW */
134     buf += snprintf(buf, bufend - buf, " ");

136     /* LINTED E_PTRDIFF_OVERFLOW */
137     return (buf - orig_buf);
138 }
    unchanged_portion_omitted_

194 void
195 virtio_write_device_config_1(struct virtio_softc *sc, unsigned int index,
196     uint8_t value)
197 virtio_write_device_config_1(struct virtio_softc *sc,
198     unsigned int index, uint8_t value)
197 {
198     ASSERT(sc->sc_config_offset);
199     ddi_put8(sc->sc_ioh,
200         (uint8_t *) (sc->sc_io_addr + sc->sc_config_offset + index), value);
201 }

203 void
204 virtio_write_device_config_2(struct virtio_softc *sc, unsigned int index,
205     uint16_t value)
206 virtio_write_device_config_2(struct virtio_softc *sc,
207     unsigned int index, uint16_t value)
206 {
207     ASSERT(sc->sc_config_offset);
208     ddi_put16(sc->sc_ioh,
209         /* LINTED E_BAD_PTR_CAST_ALIGN */
210         (uint16_t *) (sc->sc_io_addr + sc->sc_config_offset + index), value);
211 }

213 void
214 virtio_write_device_config_4(struct virtio_softc *sc, unsigned int index,
215     uint32_t value)
216 virtio_write_device_config_4(struct virtio_softc *sc,
217     unsigned int index, uint32_t value)
216 {
217     ASSERT(sc->sc_config_offset);
218     ddi_put32(sc->sc_ioh,
219         /* LINTED E_BAD_PTR_CAST_ALIGN */
220         (uint32_t *) (sc->sc_io_addr + sc->sc_config_offset + index), value);
221 }

223 void
224 virtio_write_device_config_8(struct virtio_softc *sc, unsigned int index,
225     uint64_t value)
226 virtio_write_device_config_8(struct virtio_softc *sc,
227     unsigned int index, uint64_t value)

```

```

226 {
227     ASSERT(sc->sc_config_offset);
228     ddi_put32(sc->sc_ioh,
229         /* LINTED E_BAD_PTR_CAST_ALIGN */
230         (uint32_t *) (sc->sc_io_addr + sc->sc_config_offset + index),
231         value & 0xFFFFFFFF);
232     ddi_put32(sc->sc_ioh,
233         /* LINTED E_BAD_PTR_CAST_ALIGN */
234         (uint32_t *) (sc->sc_io_addr + sc->sc_config_offset +
235             index + sizeof (uint32_t)), value >> 32);
236 }
    unchanged_portion_omitted_

253 static ddi_dma_attr_t virtio_vq_dma_attr = {
254     DMA_ATTR_V0,          /* Version number */
255     0,                   /* low address */
256     0x0000FFFFFFFFFFFF, /* high address. Has to fit into 32 bits */
257                         /* after page-shifting */
258     /*
259      * high address. Has to fit into 32 bits
260      * after page-shifting
261      */
262     0x0000FFFFFFFFFFFF,
263     0xFFFFFFFF,         /* counter register max */
264     VIRTIO_PAGE_SIZE,   /* page alignment required */
265     0x3F,               /* burst sizes: 1 - 32 */
266     0x1,               /* minimum transfer size */
267     0xFFFFFFFF,        /* max transfer size */
268     0xFFFFFFFF,        /* address register max */
269     1,                 /* no scatter-gather */
270     1,                 /* device operates on bytes */
271     0,                 /* attr flag: set to 0 */
272 };
    unchanged_portion_omitted_

304 static int
305 virtio_alloc_indirect(struct virtio_softc *sc, struct vq_entry *entry)
306 {
307     int allocsize, num;
308     size_t len;
309     unsigned int ncookies;
310     int ret;

312     num = entry->qe_queue->vq_indirect_num;
313     ASSERT(num > 1);

315     allocsize = sizeof (struct vring_desc) * num;

317     ret = ddi_dma_alloc_handle(sc->sc_dev, &virtio_vq_indirect_dma_attr,
318         DDI_DMA_SLEEP, NULL, &entry->qe_indirect_dma_handle);
319     if (ret != DDI_SUCCESS) {
320         dev_err(sc->sc_dev, CE_WARN,
321             "Failed to allocate dma handle for indirect descriptors, "
322             "entry %d, vq %d", entry->qe_index,
323             "Failed to allocate dma handle for indirect descriptors, "
324             "entry %d, vq %d", entry->qe_index,
325             entry->qe_queue->vq_index);
326         goto out_alloc_handle;
327     }

329     ret = ddi_dma_mem_alloc(entry->qe_indirect_dma_handle, allocsize,
330         &virtio_vq_devattr, DDI_DMA_CONSISTENT, DDI_DMA_SLEEP, NULL,
331         ret = ddi_dma_mem_alloc(entry->qe_indirect_dma_handle,
332             allocsize, &virtio_vq_devattr,
333             DDI_DMA_CONSISTENT, DDI_DMA_SLEEP, NULL,

```



```

442  /* size 0 => use native vq size, good for receive queues. */
443  if (size)
444      vq_size = MIN(vq_size, size);

446  /* allocsize1: descriptor table + avail ring + pad */
447  allocsize1 = VIRTQUEUE_ALIGN(sizeof (struct vring_desc) * vq_size +
448      sizeof (struct vring_avail) + sizeof (uint16_t) * vq_size);
449  /* allocsize2: used ring + pad */
450  allocsize2 = VIRTQUEUE_ALIGN(sizeof (struct vring_used) +
451      sizeof (struct vring_used_elem) * vq_size);
452  allocsize2 = VIRTQUEUE_ALIGN(sizeof (struct vring_used)
453      + sizeof (struct vring_used_elem) * vq_size);

455  allocsize = allocsize1 + allocsize2;

457  ret = ddi_dma_alloc_handle(sc->sc_dev, &virtio_vq_dma_attr,
458      DDI_DMA_SLEEP, NULL, &vq->vq_dma_handle);
459  if (ret != DDI_SUCCESS) {
460      dev_err(sc->sc_dev, CE_WARN,
461          "Failed to allocate dma handle for vq %d", index);
462      goto out_alloc_handle;
463  }

465  ret = ddi_dma_mem_alloc(vq->vq_dma_handle, allocsize,
466      &virtio_vq_devattr, DDI_DMA_CONSISTENT, DDI_DMA_SLEEP, NULL,
467      (caddr_t *)&vq->vq_vaddr, &len, &vq->vq_dma_acch);
468  if (ret != DDI_SUCCESS) {
469      dev_err(sc->sc_dev, CE_WARN,
470          "Failed to allocate dma memory for vq %d", index);
471      "Failed to allocate dma memory for vq %d", index);
472      goto out_alloc;
473  }

475  ret = ddi_dma_addr_bind_handle(vq->vq_dma_handle, NULL,
476      (caddr_t)vq->vq_vaddr, len, DDI_DMA_RDWR | DDI_DMA_CONSISTENT,
477      (caddr_t)vq->vq_vaddr, len,
478      DDI_DMA_RDWR | DDI_DMA_CONSISTENT,
479      DDI_DMA_SLEEP, NULL, &vq->vq_dma_cookie, &ncookies);
480  if (ret != DDI_DMA_MAPPED) {
481      dev_err(sc->sc_dev, CE_WARN,
482          "Failed to bind dma memory for vq %d", index);
483      goto out_bind;
484  }

486  /* We asked for a single segment */
487  ASSERT(ncookies == 1);
488  /* and page-aligned buffers. */
489  ASSERT(vq->vq_dma_cookie.dmac_laddress % VIRTIO_PAGE_SIZE == 0);

491  (void) memset(vq->vq_vaddr, 0, allocsize);

493  /* Make sure all zeros hit the buffer before we point the host to it */
494  membar_producer();

496  /* set the vq address */
497  ddi_put32(sc->sc_ioh,
498      /* LINTED E_BAD_PTR_CAST_ALIGN */
499      (uint32_t *) (sc->sc_io_addr + VIRTIO_CONFIG_QUEUE_ADDRESS),
500      (vq->vq_dma_cookie.dmac_laddress / VIRTIO_PAGE_SIZE));

502  /* remember addresses and offsets for later use */
503  vq->vq_owner = sc;
504  vq->vq_num = vq_size;

```

```

506  vq->vq_index = index;
507  vq->vq_descs = vq->vq_vaddr;
508  vq->vq_availoffset = sizeof (struct vring_desc) * vq_size;
509  vq->vq_avail = (void *) (((char *) vq->vq_descs) + vq->vq_availoffset);
510  vq->vq_usedoffset = allocsize1;
511  vq->vq_used = (void *) (((char *) vq->vq_descs) + vq->vq_usedoffset);

513  ASSERT(indirect_num == 0 ||
514      virtio_has_feature(sc, VIRTIO_F_RING_INDIRECT_DESC));
515  vq->vq_indirect_num = indirect_num;

517  /* free slot management */
518  vq->vq_entries = kmem_zalloc(sizeof (struct vq_entry) * vq_size,
519      KM_SLEEP);

521  ret = virtio_init_vq(sc, vq);
522  if (ret)
523      goto out_init;

525  dev_debug(sc->sc_dev, CE_NOTE,
526      "Allocated %d entries for vq %d: %s (%d indirect desc)",
527      "Allocated %d entries for vq %d: %s (%d indirect desc)",
528      vq_size, index, name, indirect_num * vq_size);

530  return (vq);

532 out_init:
533  kmem_free(vq->vq_entries, sizeof (struct vq_entry) * vq_size);
534  (void) ddi_dma_unbind_handle(vq->vq_dma_handle);
535 out_bind:
536  ddi_dma_mem_free(&vq->vq_dma_acch);
537 out_alloc:
538  ddi_dma_free_handle(&vq->vq_dma_handle);
539 out_alloc_handle:
540  kmem_free(vq, sizeof (struct virtqueue));
541 out:
542  return (NULL);
543 }

545 void
546 virtio_free_vq(struct virtqueue *vq)
547 {
548     struct virtio_softc *sc = vq->vq_owner;
549     int i;

551     /* tell device that there's no virtqueue any longer */
552     ddi_put16(sc->sc_ioh,
553         /* LINTED E_BAD_PTR_CAST_ALIGN */
554         (uint16_t *) (sc->sc_io_addr + VIRTIO_CONFIG_QUEUE_SELECT),
555         vq->vq_index);
556     ddi_put32(sc->sc_ioh,
557         /* LINTED E_BAD_PTR_CAST_ALIGN */
558         (uint32_t *) (sc->sc_io_addr + VIRTIO_CONFIG_QUEUE_ADDRESS), 0);

560     /* Free the indirect descriptors, if any. */
561     for (i = 0; i < vq->vq_num; i++) {
562         struct vq_entry *entry = &vq->vq_entries[i];
563         if (entry->qe_indirect_descs)
564             virtio_free_indirect(entry);
565     }

567     kmem_free(vq->vq_entries, sizeof (struct vq_entry) * vq->vq_num);

569     (void) ddi_dma_unbind_handle(vq->vq_dma_handle);
570     ddi_dma_mem_free(&vq->vq_dma_acch);

```

```

564     ddi_dma_free_handle(&vq->vq_dma_handle);

566     mutex_destroy(&vq->vq_used_lock);
567     mutex_destroy(&vq->vq_avail_lock);
568     mutex_destroy(&vq->vq_freelist_lock);

570     kmem_free(vq, sizeof (struct virtqueue));
571 }
    unchanged_portion_omitted_

623 static inline void
624 virtio_ve_set_desc(struct vring_desc *desc, uint64_t paddr, uint32_t len,
625     boolean_t write)
626 {
627     desc->addr = paddr;
628     desc->len = len;
629     desc->next = 0;
630     desc->flags = 0;

632     /* 'write' - from the driver's point of view */
633     if (!write)
634         desc->flags = VRING_DESC_F_WRITE;

635 }
    unchanged_portion_omitted_

644 unsigned int
645 virtio_ve_indirect_available(struct vq_entry *qe)
646 {
647     return (qe->qe_queue->vq_indirect_num - (qe->qe_indirect_next - 1));
648 }

650 void
651 virtio_ve_add_indirect_buf(struct vq_entry *qe, uint64_t paddr, uint32_t len,
652     boolean_t write)
653 {
654     struct vring_desc *indirect_desc;

656     ASSERT(qe->qe_queue->vq_indirect_num);
657     ASSERT(qe->qe_indirect_next < qe->qe_queue->vq_indirect_num);

659     indirect_desc = &qe->qe_indirect_descs[qe->qe_indirect_next];
660     virtio_ve_set_desc(indirect_desc, paddr, len, write);
661     qe->qe_indirect_next++;
662 }
    unchanged_portion_omitted_

677 void
678 virtio_sync_vq(struct virtqueue *vq)
679 {
680     struct virtio_softc *vsc = vq->vq_owner;

682     /* Make sure the avail ring update hit the buffer */
683     membar_producer();

685     vq->vq_avail->idx = vq->vq_avail_idx;

687     /* Make sure the avail idx update hits the buffer */
688     membar_producer();

690     /* Make sure we see the flags update */
691     membar_consumer();

693     if (!(vq->vq_used->flags & VRING_USED_F_NO_NOTIFY)) {
705         if (!(vq->vq_used->flags & VRING_USED_F_NO_NOTIFY)) {

```

```

694         ddi_putl6(vsc->sc_ioh,
695             /* LINTED E_BAD_PTR_CAST_ALIGN */
696             (uint16_t *) (vsc->sc_io_addr +
697                 VIRTIO_CONFIG_QUEUE_NOTIFY),
698             vq->vq_index);
699     }
700 }
    unchanged_portion_omitted_

765 /*
766  * Get a chain of descriptors from the used ring, if one is available.
767  */
776 /* Get a chain of descriptors from the used ring, if one is available. */
778 struct vq_entry *
779 virtio_pull_chain(struct virtqueue *vq, uint32_t *len)
780 {
781     struct vq_entry *head;
782     int slot;
783     int usedidx;

785     mutex_enter(&vq->vq_used_lock);

787     /* No used entries? Bye. */
788     if (vq->vq_used_idx == vq->vq_used->idx) {
789         mutex_exit(&vq->vq_used_lock);
790         return (NULL);
791     }

793     usedidx = vq->vq_used_idx;
794     vq->vq_used_idx++;
795     mutex_exit(&vq->vq_used_lock);

797     usedidx %= vq->vq_num;

799     /* Make sure we do the next step _after_ checking the idx. */
800     membar_consumer();

802     slot = vq->vq_used->ring[usedidx].id;
803     *len = vq->vq_used->ring[usedidx].len;

805     head = &vq->vq_entries[slot];

807     return (head);
808 }

810 void
811 virtio_free_chain(struct vq_entry *qe)
812 {
813     struct vq_entry *tmp;
814     struct virtqueue *vq = qe->qe_queue;

816     ASSERT(qe);

818     do {
819         ASSERT(qe->qe_queue == vq);
820         tmp = qe->qe_next;
821         vq_free_entry(vq, qe);
822         qe = tmp;
823     } while (tmp != NULL);
824 }
    unchanged_portion_omitted_

826 static int
827 virtio_register_msi(struct virtio_softc *sc,
828     struct virtio_int_handler *config_handler,

```

```

825 struct virtio_intr_handler vq_handlers[], int intr_types)
834 struct virtio_intr_handler vq_handlers[],
835 int intr_types)
826 {
827     int count, actual;
828     int int_type;
829     int i;
830     int handler_count;
831     int ret;

833     /* If both MSI and MSI-x are reported, prefer MSI-x. */
834     int_type = DDI_INTR_TYPE_MSI;
835     if (intr_types & DDI_INTR_TYPE_MSIX)
836         int_type = DDI_INTR_TYPE_MSIX;

838     /* Walk the handler table to get the number of handlers. */
839     for (handler_count = 0;
840         vq_handlers && vq_handlers[handler_count].vh_func;
841         handler_count++)
842         ;

844     /* +1 if there is a config change handler. */
845     if (config_handler != NULL)
846         handler_count++;

848     /* Number of MSIs supported by the device. */
849     ret = ddi_intr_get_nintrs(sc->sc_dev, int_type, &count);
850     if (ret != DDI_SUCCESS) {
851         dev_err(sc->sc_dev, CE_WARN, "ddi_intr_get_nintrs failed");
852         return (ret);
853     }

855     /*
856      * Those who try to register more handlers than the device
857      * supports shall suffer.
858      */
859     ASSERT(handler_count <= count);

861     sc->sc_intr_htable = kmem_zalloc(sizeof (ddi_intr_handle_t) *
862     handler_count, KM_SLEEP);
863     sc->sc_intr_htable = kmem_zalloc(
864     sizeof (ddi_intr_handle_t) * handler_count, KM_SLEEP);

864     ret = ddi_intr_alloc(sc->sc_dev, sc->sc_intr_htable, int_type, 0,
865     handler_count, &actual, DDI_INTR_ALLOC_NORMAL);
866     if (ret != DDI_SUCCESS) {
867         dev_err(sc->sc_dev, CE_WARN, "Failed to allocate MSI: %d", ret);
868         goto out_msi_alloc;
869     }

871     if (actual != handler_count) {
872         dev_err(sc->sc_dev, CE_WARN,
873             "Not enough MSI available: need %d, available %d",
874             handler_count, actual);
875         goto out_msi_available;
876     }

878     sc->sc_intr_num = handler_count;
879     sc->sc_intr_config = B_FALSE;
880     if (config_handler != NULL) {
881         if (config_handler) {
882             sc->sc_intr_config = B_TRUE;
883         }
884     }

884     /* Assume they are all same priority */

```

```

885     ret = ddi_intr_get_pri(sc->sc_intr_htable[0], &sc->sc_intr_prio);
886     if (ret != DDI_SUCCESS) {
887         dev_err(sc->sc_dev, CE_WARN, "ddi_intr_get_pri failed");
888         goto out_msi_prio;
889     }

891     /* Add the vq handlers */
892     for (i = 0; vq_handlers[i].vh_func; i++) {
893         ret = ddi_intr_add_handler(sc->sc_intr_htable[i],
894             vq_handlers[i].vh_func, sc, vq_handlers[i].vh_priv);
895         if (ret != DDI_SUCCESS) {
896             dev_err(sc->sc_dev, CE_WARN,
897                 "ddi_intr_add_handler failed");
898             /* Remove the handlers that succeeded. */
899             while (--i >= 0) {
900                 (void) ddi_intr_remove_handler(
901                     sc->sc_intr_htable[i]);
902             }
903             goto out_add_handlers;
904         }
905     }

907     /* Don't forget the config handler */
908     if (config_handler != NULL) {
909         if (config_handler) {
910             ret = ddi_intr_add_handler(sc->sc_intr_htable[i],
911                 config_handler->vh_func, sc, config_handler->vh_priv);
912             if (ret != DDI_SUCCESS) {
913                 dev_err(sc->sc_dev, CE_WARN,
914                     "ddi_intr_add_handler failed");
915                 /* Remove the handlers that succeeded. */
916                 while (--i >= 0) {
917                     (void) ddi_intr_remove_handler(
918                         sc->sc_intr_htable[i]);
919                 }
920                 goto out_add_handlers;
921             }
922         }
923     }

923     /* We know we are using MSI, so set the config offset. */
924     sc->sc_config_offset = VIRTIO_CONFIG_DEVICE_CONFIG_MSI;

926     ret = ddi_intr_get_cap(sc->sc_intr_htable[0], &sc->sc_intr_cap);
927     ret = ddi_intr_get_cap(sc->sc_intr_htable[0],
928     &sc->sc_intr_cap);
929     /* Just in case. */
930     if (ret != DDI_SUCCESS)
931         sc->sc_intr_cap = 0;

931 out_add_handlers:
932 out_msi_prio:
933 out_msi_available:
934     for (i = 0; i < actual; i++)
935         (void) ddi_intr_free(sc->sc_intr_htable[i]);
936 out_msi_alloc:
937     kmem_free(sc->sc_intr_htable, sizeof (ddi_intr_handle_t) * count);

939     return (ret);
940 }

unchanged_portion_omitted

977 /*

```

```

978 * config_handler and vq_handlers may be allocated on stack.
979 * Take precautions not to loose them.
980 */
981 static int
982 virtio_register_intx(struct virtio_softc *sc,
983     struct virtio_int_handler *config_handler,
984     struct virtio_int_handler vq_handlers[])
985 {
986     int vq_handler_count;
987     int config_handler_count = 0;
988     int actual;
989     struct virtio_handler_container *vhc;
990     int ret = DDI_FAILURE;

992     /* Walk the handler table to get the number of handlers. */
993     for (vq_handler_count = 0;
994         vq_handlers && vq_handlers[vq_handler_count].vh_func;
995         vq_handler_count++);
996

998     if (config_handler != NULL)
1011     if (config_handler)
999         config_handler_count = 1;

1001     vhc = kmem_zalloc(sizeof (struct virtio_handler_container) +
1002         sizeof (struct virtio_int_handler) * vq_handler_count, KM_SLEEP);
1015     sizeof (struct virtio_int_handler) * vq_handler_count,
1016     KM_SLEEP);

1004     vhc->nhandlers = vq_handler_count;
1005     (void) memcpy(vhc->vq_handlers, vq_handlers,
1006         sizeof (struct virtio_int_handler) * vq_handler_count);

1008     if (config_handler != NULL) {
1022     if (config_handler) {
1009         (void) memcpy(&vhc->config_handler, config_handler,
1010             sizeof (struct virtio_int_handler));
1011     }

1013     /* Just a single entry for a single interrupt. */
1014     sc->sc_intr_htable = kmem_zalloc(sizeof (ddi_intr_handle_t), KM_SLEEP);

1016     ret = ddi_intr_alloc(sc->sc_dev, sc->sc_intr_htable,
1017         DDI_INTR_TYPE_FIXED, 0, 1, &actual, DDI_INTR_ALLOC_NORMAL);
1031     DDI_INTR_TYPE_FIXED, 0, 1, &actual,
1032     DDI_INTR_ALLOC_NORMAL);
1018     if (ret != DDI_SUCCESS) {
1019         dev_err(sc->sc_dev, CE_WARN,
1020             "Failed to allocate a fixed interrupt: %d", ret);
1021         goto out_int_alloc;
1022     }

1024     ASSERT(actual == 1);
1025     sc->sc_intr_num = 1;

1027     ret = ddi_intr_get_pri(sc->sc_intr_htable[0], &sc->sc_intr_prio);
1028     if (ret != DDI_SUCCESS) {
1029         dev_err(sc->sc_dev, CE_WARN, "ddi_intr_get_pri failed");
1030         goto out_prio;
1031     }

1033     ret = ddi_intr_add_handler(sc->sc_intr_htable[0],
1034         virtio_intx_dispatch, sc, vhc);
1035     if (ret != DDI_SUCCESS) {
1036         dev_err(sc->sc_dev, CE_WARN, "ddi_intr_add_handler failed");
1037         goto out_add_handlers;

```

```

1038     }

1040     /* We know we are not using MSI, so set the config offset. */
1041     sc->sc_config_offset = VIRTIO_CONFIG_DEVICE_CONFIG_NOMSI;

1043     return (DDI_SUCCESS);

1045 out_add_handlers:
1046 out_prio:
1047     (void) ddi_intr_free(sc->sc_intr_htable[0]);
1048 out_int_alloc:
1049     kmem_free(sc->sc_intr_htable, sizeof (ddi_intr_handle_t));
1050     kmem_free(vhc, sizeof (struct virtio_int_handler) *
1051         (vq_handler_count + config_handler_count));
1052     return (ret);
1053 }
----- unchanged portion omitted -----

1099 static int
1100 virtio_enable_msi(struct virtio_softc *sc)
1101 {
1102     int ret, i;
1103     int vq_handler_count = sc->sc_intr_num;

1105     /* Number of handlers, not counting the counfig. */
1106     if (sc->sc_intr_config)
1107         vq_handler_count--;

1109     /* Enable the interrupts. Either the whole block, or one by one. */
1110     if (sc->sc_intr_cap & DDI_INTR_FLAG_BLOCK) {
1111         ret = ddi_intr_block_enable(sc->sc_intr_htable,
1112             sc->sc_intr_num);
1113         if (ret != DDI_SUCCESS) {
1114             dev_err(sc->sc_dev, CE_WARN,
1115                 "Failed to enable MSI, falling back to INTx");
1116             goto out_enable;
1117         }
1118     } else {
1119         for (i = 0; i < sc->sc_intr_num; i++) {
1120             ret = ddi_intr_enable(sc->sc_intr_htable[i]);
1121             if (ret != DDI_SUCCESS) {
1122                 dev_err(sc->sc_dev, CE_WARN,
1123                     "Failed to enable MSI %d, "
1124                     "falling back to INTx", i);

1126                 while (--i >= 0) {
1127                     (void) ddi_intr_disable(
1128                         sc->sc_intr_htable[i]);
1129                 }
1130                 goto out_enable;
1131             }
1132         }
1133     }

1135     /* Bind the allocated MSI to the queues and config */
1136     for (i = 0; i < vq_handler_count; i++) {
1137         int check;

1139         ddi_put16(sc->sc_ioh,
1140             /* LINTED E_BAD_PTR_CAST_ALIGN */
1141             (uint16_t *) (sc->sc_io_addr +
1142                 VIRTIO_CONFIG_QUEUE_SELECT), i);

1144         ddi_put16(sc->sc_ioh,
1145             /* LINTED E_BAD_PTR_CAST_ALIGN */

```

```

1146         (uint16_t *) (sc->sc_io_addr +
1147         VIRTIO_CONFIG_QUEUE_VECTOR), i);

1149     check = ddi_get16(sc->sc_ioh,
1150     /* LINTED E_BAD_PTR_CAST_ALIGN */
1151     (uint16_t *) (sc->sc_io_addr +
1152     VIRTIO_CONFIG_QUEUE_VECTOR));
1153     if (check != i) {
1154         dev_err(sc->sc_dev, CE_WARN, "Failed to bind handler "
1155         dev_err(sc->sc_dev, CE_WARN, "Failed to bind handler"
1156         "for VQ %d, MSI %d. Check = %x", i, i, check);
1157         ret = ENODEV;
1158         goto out_bind;
1159     }

1161     if (sc->sc_intr_config) {
1162         int check;

1164         ddi_put16(sc->sc_ioh,
1165         /* LINTED E_BAD_PTR_CAST_ALIGN */
1166         (uint16_t *) (sc->sc_io_addr +
1167         VIRTIO_CONFIG_CONFIG_VECTOR), i);

1169         check = ddi_get16(sc->sc_ioh,
1170         /* LINTED E_BAD_PTR_CAST_ALIGN */
1171         (uint16_t *) (sc->sc_io_addr +
1172         VIRTIO_CONFIG_CONFIG_VECTOR));
1173         if (check != i) {
1174             dev_err(sc->sc_dev, CE_WARN, "Failed to bind handler "
1175             "for Config updates, MSI %d", i);
1176             ret = ENODEV;
1177             goto out_bind;
1178         }
1179     }

1181     return (DDI_SUCCESS);

1183 out_bind:
1184     /* Unbind the vqs */
1185     for (i = 0; i < vq_handler_count - 1; i++) {
1186         ddi_put16(sc->sc_ioh,
1187         /* LINTED E_BAD_PTR_CAST_ALIGN */
1188         (uint16_t *) (sc->sc_io_addr +
1189         VIRTIO_CONFIG_QUEUE_SELECT), i);

1191         ddi_put16(sc->sc_ioh,
1192         /* LINTED E_BAD_PTR_CAST_ALIGN */
1193         (uint16_t *) (sc->sc_io_addr +
1194         VIRTIO_CONFIG_QUEUE_VECTOR),
1195         VIRTIO_MSI_NO_VECTOR);
1196     }
1197     /* And the config */
1198     /* LINTED E_BAD_PTR_CAST_ALIGN */
1199     ddi_put16(sc->sc_ioh, (uint16_t *) (sc->sc_io_addr +
1200     VIRTIO_CONFIG_CONFIG_VECTOR), VIRTIO_MSI_NO_VECTOR);

1202     ret = DDI_FAILURE;

1204 out_enable:
1205     return (ret);
1206 }

1208 static int
1209 virtio_enable_intx(struct virtio_softc *sc)
1222 static int virtio_enable_intx(struct virtio_softc *sc)

```

```

1210 {
1211     int ret;

1213     ret = ddi_intr_enable(sc->sc_intr_htable[0]);
1214     if (ret != DDI_SUCCESS) {
1227     if (ret != DDI_SUCCESS)
1225         dev_err(sc->sc_dev, CE_WARN,
1216         "Failed to enable interrupt: %d", ret);
1217     }

1219     return (ret);
1220 }
    unchanged_portion_omitted

1239 void
1240 virtio_release_ints(struct virtio_softc *sc)
1241 {
1242     int i;
1243     int ret;

1245     /* We were running with MSI, unbind them. */
1246     if (sc->sc_config_offset == VIRTIO_CONFIG_DEVICE_CONFIG_MSI) {
1247         /* Unbind all vqs */
1248         for (i = 0; i < sc->sc_nvqs; i++) {
1249             ddi_put16(sc->sc_ioh,
1250             /* LINTED E_BAD_PTR_CAST_ALIGN */
1251             (uint16_t *) (sc->sc_io_addr +
1252             VIRTIO_CONFIG_QUEUE_SELECT), i);

1254             ddi_put16(sc->sc_ioh,
1255             /* LINTED E_BAD_PTR_CAST_ALIGN */
1256             (uint16_t *) (sc->sc_io_addr +
1257             VIRTIO_CONFIG_QUEUE_VECTOR),
1258             VIRTIO_MSI_NO_VECTOR);
1259         }
1260         /* And the config */
1261         /* LINTED E_BAD_PTR_CAST_ALIGN */
1262         ddi_put16(sc->sc_ioh, (uint16_t *) (sc->sc_io_addr +
1263         VIRTIO_CONFIG_CONFIG_VECTOR),
1264         VIRTIO_MSI_NO_VECTOR);

1266     }

1268     /* Disable the interrupts. Either the whole block, or one by one. */
1269     if (sc->sc_intr_cap & DDI_INTR_FLAG_BLOCK) {
1270         ret = ddi_intr_block_disable(sc->sc_intr_htable,
1271         sc->sc_intr_num);
1272         if (ret != DDI_SUCCESS) {
1273             dev_err(sc->sc_dev, CE_WARN,
1274             "Failed to disable MSIs, won't be able to "
1275             "Failed to disable MSIs, won't be able to "
1276             "reuse next time");
1277         }
1278     } else {
1279         for (i = 0; i < sc->sc_intr_num; i++) {
1280             ret = ddi_intr_disable(sc->sc_intr_htable[i]);
1281             if (ret != DDI_SUCCESS) {
1282                 dev_err(sc->sc_dev, CE_WARN,
1283                 "Failed to disable interrupt %d,"
1284                 "won't be able to reuse", i);
1285             }
1286         }
    }

```

```
1289     for (i = 0; i < sc->sc_intr_num; i++) {
1290         (void) ddi_intr_remove_handler(sc->sc_intr_htable[i]);
1291     }
1293     for (i = 0; i < sc->sc_intr_num; i++)
1294         (void) ddi_intr_free(sc->sc_intr_htable[i]);
1296     kmem_free(sc->sc_intr_htable, sizeof (ddi_intr_handle_t) *
1297             sc->sc_intr_num);
1308     kmem_free(sc->sc_intr_htable,
1309             sizeof (ddi_intr_handle_t) * sc->sc_intr_num);
1299     /* After disabling interrupts, the config offset is non-MSI. */
1300     sc->sc_config_offset = VIRTIO_CONFIG_DEVICE_CONFIG_NOMSI;
1301 }
```

unchanged_portion_omitted_

```
*****
7589 Thu Mar 28 23:16:19 2013
new/usr/src/uts/common/io/virtio/virtiovar.h
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xfee@gmail.com
*****
_____unchanged_portion_omitted_____

155 /* public interface */
156 uint32_t virtio_negotiate_features(struct virtio_softc *, uint32_t);
157 size_t virtio_show_features(uint32_t features, char *buffer, size_t len);
158 boolean_t virtio_has_feature(struct virtio_softc *sc, uint32_t feature);
159 void virtio_set_status(struct virtio_softc *sc, unsigned int);
160 #define virtio_device_reset(sc) virtio_set_status((sc), 0)

162 uint8_t virtio_read_device_config_1(struct virtio_softc *sc,
163     unsigned int index);
164 uint16_t virtio_read_device_config_2(struct virtio_softc *sc,
165     unsigned int index);
166 uint32_t virtio_read_device_config_4(struct virtio_softc *sc,
167     unsigned int index);
168 uint64_t virtio_read_device_config_8(struct virtio_softc *sc,
169     unsigned int index);
170 void virtio_write_device_config_1(struct virtio_softc *sc,
171     unsigned int index, uint8_t value);
172 void virtio_write_device_config_2(struct virtio_softc *sc,
173     unsigned int index, uint16_t value);
174 void virtio_write_device_config_4(struct virtio_softc *sc,
175     unsigned int index, uint32_t value);
176 void virtio_write_device_config_8(struct virtio_softc *sc,
177     unsigned int index, uint64_t value);

179 struct virtqueue *virtio_alloc_vq(struct virtio_softc *sc,
180     unsigned int index, unsigned int size,
181     unsigned int indirect_num, const char *name);
182 void virtio_free_vq(struct virtqueue *);
183 void virtio_reset(struct virtio_softc *);
184 struct vq_entry *vq_alloc_entry(struct virtqueue *vq);
185 void vq_free_entry(struct virtqueue *vq, struct vq_entry *qe);
186 uint_t vq_num_used(struct virtqueue *vq);
187 unsigned int virtio_ve_indirect_available(struct vq_entry *qe);

189 void virtio_stop_vq_intr(struct virtqueue *);
190 void virtio_start_vq_intr(struct virtqueue *);

192 void virtio_ve_add_cookie(struct vq_entry *qe, ddi_dma_handle_t dma_handle,
193     ddi_dma_cookie_t dma_cookie, unsigned int ncookies, boolean_t write);
194 void virtio_ve_add_indirect_buf(struct vq_entry *qe, uint64_t paddr,
195     uint32_t len, boolean_t write);
196 void virtio_ve_set(struct vq_entry *qe, uint64_t paddr, uint32_t len,
197     boolean_t write);

199 void virtio_push_chain(struct vq_entry *qe, boolean_t sync);
200 struct vq_entry *virtio_pull_chain(struct virtqueue *vq, uint32_t *len);
201 void virtio_free_chain(struct vq_entry *ve);
202 void virtio_sync_vq(struct virtqueue *vq);

204 int virtio_register_ints(struct virtio_softc *sc,
205     struct virtio_int_handler *config_handler,
206     struct virtio_int_handler vq_handlers[]);
207 void virtio_release_ints(struct virtio_softc *sc);
208 int virtio_enable_ints(struct virtio_softc *sc);

210 #endif /* __VIRTIOVAR_H__ */
```



```

*****
16947 Thu Mar 28 23:16:19 2013
new/usr/src/uts/intel/Makefile.intel.shared
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xffea@gmail.com
*****
1 # CDDL HEADER START
2 #
3 # The contents of this file are subject to the terms of the
4 # Common Development and Distribution License (the "License").
5 # You may not use this file except in compliance with the License.
6 #
7 # You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
8 # or http://www.opensolaris.org/os/licensing.
9 # See the License for the specific language governing permissions
10 # and limitations under the License.
11 #
12 # When distributing Covered Code, include this CDDL HEADER in each
13 # file and include the License file at usr/src/OPENSOLARIS.LICENSE.
14 # If applicable, add the following below this CDDL HEADER, with the
15 # fields enclosed by brackets "[]" replaced with your own identifying
16 # information: Portions Copyright [yyyy] [name of copyright owner]
17 #
18 # CDDL HEADER END
19 #
20 #
21 # Copyright (c) 2005, 2010, Oracle and/or its affiliates. All rights reserved.
22 # Copyright (c) 2012 Nexenta Systems, Inc. All rights reserved.
23 #
24 #
25 # This makefile contains the common definitions for all intel
26 # implementation architecture independent modules.
27 #
28 #
29 #
30 # Machine type (implementation architecture):
31 #
32 PLATFORM = i86pc
33 #
34 #
35 # Everybody needs to know how to build modstubs.o and to locate unix.o.
36 # Note that unix.o must currently be selected from among the possible
37 # "implementation architectures". Note further, that unix.o is only
38 # used as an optional error check for undefines so (theoretically)
39 # any "implementation architectures" could be used. We choose i86pc
40 # because it is the reference port.
41 #
42 UNIX_DIR = $(UTSBASE)/i86pc/unix
43 GENLIB_DIR = $(UTSBASE)/intel/genunix
44 IPDRV_DIR = $(UTSBASE)/intel/ip
45 MODSTUBS_DIR = $(UNIX_DIR)
46 DSF_DIR = $(UTSBASE)/$(PLATFORM)/genassym
47 LINTS_DIR = $(OBJS_DIR)
48 LINT_LIB_DIR = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)
49 #
50 UNIX_O = $(UNIX_DIR)/$(OBJS_DIR)/unix.o
51 GENLIB = $(GENLIB_DIR)/$(OBJS_DIR)/libgenunix.so
52 MODSTUBS_O = $(MODSTUBS_DIR)/$(OBJS_DIR)/modstubs.o
53 LINT_LIB = $(UTSBASE)/i86pc/lint-libs/$(OBJS_DIR)/llib-lunix.ln
54 GEN_LINT_LIB = $(UTSBASE)/intel/lint-libs/$(OBJS_DIR)/llib-lgenunix.ln
55 #
56 #
57 # Include the makefiles which define build rule templates, the
58 # collection of files per module, and a few specific flags. Note

```

```

59 # that order is significant, just as with an include path. The
60 # first build rule template which matches the files name will be
61 # used. By including these in order from most machine dependent
62 # to most machine independent, we allow a machine dependent file
63 # to be used in preference over a machine independent version
64 # (Such as a machine specific optimization, which preserves the
65 # interfaces.)
66 #
67 include $(UTSTREE)/intel/Makefile.files
68 include $(UTSTREE)/common/Makefile.files
69 #
70 #
71 # ----- TRANSITIONAL SECTION -----
72 #
73 #
74 #
75 # Not everything which *should* be a module is a module yet. The
76 # following is a list of such objects which are currently part of
77 # genunix but which might someday become kmods. This must be
78 # defined before we include Makefile.uts, or else genunix's build
79 # won't be as parallel as we might like.
80 #
81 NOT_YET_KMODS = $(OLDPTY_OBJS) $(PTY_OBJS) $(VCONS_CONF_OBJS) $(MOD_OBJS)
82 #
83 #
84 # ----- END OF TRANSITIONAL SECTION -----
85 #
86 #
87 # Include machine independent rules. Note that this does not imply
88 # that the resulting module from rules in Makefile.uts is machine
89 # independent. Only that the build rules are machine independent.
90 #
91 include $(UTSBASE)/Makefile.uts
92 #
93 # The following must be defined for all implementations:
94 #
95 MODSTUBS = $(UTSBASE)/intel/ia32/ml/modstubs.s
96 #
97 #
98 # Define supported builds
99 #
100 DEF_BUILDS = $(DEF_BUILDS64) $(DEF_BUILDS32)
101 ALL_BUILDS = $(ALL_BUILDS64) $(ALL_BUILDS32)
102 #
103 #
104 # x86 or amd64 inline templates
105 #
106 INLINES_32 = $(UTSBASE)/intel/ia32/ml/ia32.il
107 INLINES_64 = $(UTSBASE)/intel/amd64/ml/amd64.il
108 INLINES += $(INLINES_$(CLASS))
109 #
110 #
111 # kernel-specific optimizations; override default in Makefile.master
112 #
113 #
114 CFLAGS_XARCH_32 = $(i386_CFLAGS)
115 CFLAGS_XARCH_64 = $(amd64_CFLAGS)
116 CFLAGS_XARCH = $(CFLAGS_XARCH_$(CLASS))
117 #
118 COPTFLAG_32 = $(COPTFLAG)
119 COPTFLAG_64 = $(COPTFLAG64)
120 COPTIMIZE = $(COPTFLAG_$(CLASS))
121 #
122 CFLAGS = $(CFLAGS_XARCH)
123 CFLAGS += $(COPTIMIZE)
124 CFLAGS += $(INLINES) -D_ASM_INLINES

```

```

125 CFLAGS          += $(CCMODE)
126 CFLAGS          += $(SPACEFLAG)
127 CFLAGS          += $(CCUNBOUND)
128 CFLAGS          += $(CFLAGS_uts)
129 CFLAGS          += -xstrconst

131 ASFLAGS_XARCH_32 = $(i386_ASFLAGS)
132 ASFLAGS_XARCH_64 = $(amd64_ASFLAGS)
133 ASFLAGS_XARCH    = $(ASFLAGS_XARCH_$(CLASS))

135 ASFLAGS          += $(ASFLAGS_XARCH)

137 #
138 #       Define the base directory for installation.
139 #
140 BASE_INS_DIR     = $(ROOT)

142 #
143 #       Debugging level
144 #
145 #       Special knowledge of which special debugging options affect which
146 #       file is used to optimize the build if these flags are changed.
147 #
148 DEBUG_DEFS_OBJ32 =
149 DEBUG_DEFS_DBG32 = -DDEBUG
150 DEBUG_DEFS_OBJ64 =
151 DEBUG_DEFS_DBG64 = -DDEBUG
152 DEBUG_DEFS       = $(DEBUG_DEFS_$(BUILD_TYPE))

154 DEBUG_COND_OBJ32 :sh = echo \\043
155 DEBUG_COND_DBG32 =
156 DEBUG_COND_OBJ64 :sh = echo \\043
157 DEBUG_COND_DBG64 =
158 IF_DEBUG_OBJ     = $(DEBUG_COND_$(BUILD_TYPE))$(OBJS_DIR)/

160 $(IF_DEBUG_OBJ)syscall.o := DEBUG_DEFS += -DSYSCALLTRACE
161 $(IF_DEBUG_OBJ)clock.o  := DEBUG_DEFS += -DKSLICE=1

163 #
164 #       Collect the preprocessor definitions to be associated with *all*
165 #       files.
166 #
167 ALL_DEFS         = $(DEBUG_DEFS) $(OPTION_DEFS)

169 #
170 #       The kernels modules which are "implementation architecture"
171 #       specific for this machine are enumerated below. Note that most
172 #       of these modules must exist (in one form or another) for each
173 #       architecture.
174 #
175 #       Common Drivers (usually pseudo drivers) (/kernel/drv)
176 #       DRV_KMODS are built both 32-bit and 64-bit
177 #       DRV_KMODS_32 are built only 32-bit
178 #       DRV_KMODS_64 are built only 64-bit
179 #
180 DRV_KMODS        += aac
181 DRV_KMODS        += aggr
182 DRV_KMODS        += ahci
183 DRV_KMODS        += amd64_gart
184 DRV_KMODS        += amr
185 DRV_KMODS        += agpgart
186 DRV_KMODS        += srn
187 DRV_KMODS        += agptarget
188 DRV_KMODS        += arn
189 DRV_KMODS        += arp
190 DRV_KMODS        += asy

```

```

191 DRV_KMODS        += ata
192 DRV_KMODS        += ath
193 DRV_KMODS        += atu
194 DRV_KMODS        += audio
195 DRV_KMODS        += audio1575
196 DRV_KMODS        += audio810
197 DRV_KMODS        += audiocmi
198 DRV_KMODS        += audiocmihd
199 DRV_KMODS        += audioemul0k
200 DRV_KMODS        += audioens
201 DRV_KMODS        += audiohd
202 DRV_KMODS        += audioixp
203 DRV_KMODS        += audiols
204 DRV_KMODS        += audiopl6x
205 DRV_KMODS        += audiopci
206 DRV_KMODS        += audiosolo
207 DRV_KMODS        += audiotst
208 DRV_KMODS        += audiovia823x
209 DRV_KMODS_32    += audiovia97
210 DRV_KMODS        += bl
211 DRV_KMODS        += blkdev
212 DRV_KMODS        += bge
213 DRV_KMODS        += bofi
214 DRV_KMODS        += bpf
215 DRV_KMODS        += bridge
216 DRV_KMODS        += bsdbus
217 DRV_KMODS        += bscv
218 DRV_KMODS        += chxge
219 DRV_KMODS        += ntxn
220 DRV_KMODS        += myril0ge
221 DRV_KMODS        += clone
222 DRV_KMODS        += cmdk
223 DRV_KMODS        += cn
224 DRV_KMODS        += conskbd
225 DRV_KMODS        += consms
226 DRV_KMODS        += cpuid
227 DRV_KMODS        += cpunex
228 DRV_KMODS        += crypto
229 DRV_KMODS        += cryptoadm
230 DRV_KMODS        += dca
231 DRV_KMODS        += devinfo
232 DRV_KMODS        += dld
233 DRV_KMODS        += dlpistub
234 DRV_KMODS_32    += dnet
235 DRV_KMODS        += dump
236 DRV_KMODS        += ecpp
237 DRV_KMODS        += emlxs
238 DRV_KMODS        += fd
239 DRV_KMODS        += fdc
240 DRV_KMODS        += fm
241 DRV_KMODS        += fssnap
242 DRV_KMODS        += hxge
243 DRV_KMODS        += i8042
244 DRV_KMODS        += i915
245 DRV_KMODS        += icmp
246 DRV_KMODS        += icmp6
247 DRV_KMODS        += intel_nb5000
248 DRV_KMODS        += intel_nhm
249 DRV_KMODS        += ip
250 DRV_KMODS        += ip6
251 DRV_KMODS        += ipf
252 DRV_KMODS        += ipnet
253 DRV_KMODS        += ippctl
254 DRV_KMODS        += ipsecah
255 DRV_KMODS        += ipsecesp
256 DRV_KMODS        += ipw

```

```

257 DRV_KMODS      += iwh
258 DRV_KMODS      += iwi
259 DRV_KMODS      += iwk
260 DRV_KMODS      += iwp
261 DRV_KMODS      += iwscn
262 DRV_KMODS      += kb8042
263 DRV_KMODS      += keysock
264 DRV_KMODS      += kssl
265 DRV_KMODS      += kstat
266 DRV_KMODS      += ksyms
267 DRV_KMODS      += kmdb
268 DRV_KMODS      += llcl
269 DRV_KMODS      += lofi
270 DRV_KMODS      += log
271 DRV_KMODS      += logindmux
272 DRV_KMODS      += mega_sas
273 DRV_KMODS      += mc-amd
274 DRV_KMODS      += mm
275 DRV_KMODS      += mouse8042
276 DRV_KMODS      += mpt_sas
277 DRV_KMODS      += mr_sas
278 DRV_KMODS      += mw1
279 DRV_KMODS      += nca
280 DRV_KMODS      += nsmb
281 DRV_KMODS      += nulldriver
282 DRV_KMODS      += nv_sata
283 DRV_KMODS      += nxge
284 DRV_KMODS      += oce
285 DRV_KMODS      += openeep
286 DRV_KMODS      += pci_pci
287 DRV_KMODS      += pcic
288 DRV_KMODS      += pcieb
289 DRV_KMODS      += physmem
290 DRV_KMODS      += pcan
291 DRV_KMODS      += pcwl
292 DRV_KMODS      += pit_beeper
293 DRV_KMODS      += pm
294 DRV_KMODS      += poll
295 DRV_KMODS      += pool
296 DRV_KMODS      += power
297 DRV_KMODS      += pseudo
298 DRV_KMODS      += ptc
299 DRV_KMODS      += ptm
300 DRV_KMODS      += pts
301 DRV_KMODS      += ptsl
302 DRV_KMODS      += qlge
303 DRV_KMODS      += radeon
304 DRV_KMODS      += ral
305 DRV_KMODS      += ramdisk
306 DRV_KMODS      += random
307 DRV_KMODS      += rds
308 DRV_KMODS      += rdsv3
309 DRV_KMODS      += rpcib
310 DRV_KMODS      += rsm
311 DRV_KMODS      += rts
312 DRV_KMODS      += rtw
313 DRV_KMODS      += rum
314 DRV_KMODS      += rwd
315 DRV_KMODS      += rwn
316 DRV_KMODS      += sad
317 DRV_KMODS      += sd
318 DRV_KMODS      += sdhost
319 DRV_KMODS      += sgen
320 DRV_KMODS      += si3124
321 DRV_KMODS      += sbios
322 DRV_KMODS      += softmac

```

```

323 DRV_KMODS      += spdssock
324 DRV_KMODS      += sbsrv
325 DRV_KMODS      += smp
326 DRV_KMODS      += spps
327 DRV_KMODS      += sppptun
328 DRV_KMODS      += srpt
329 DRV_KMODS      += st
330 DRV_KMODS      += sy
331 DRV_KMODS      += sysevent
332 DRV_KMODS      += sysmsg
333 DRV_KMODS      += tcp
334 DRV_KMODS      += tcp6
335 DRV_KMODS      += tl
336 DRV_KMODS      += tnf
337 DRV_KMODS      += tpm
338 DRV_KMODS      += trill
339 DRV_KMODS      += udp
340 DRV_KMODS      += udp6
341 DRV_KMODS      += ucode
342 DRV_KMODS      += ural
343 DRV_KMODS      += uath
344 DRV_KMODS      += urtw
345 DRV_KMODS      += vgatext
346 DRV_KMODS      += heci
347 DRV_KMODS      += vnic
348 DRV_KMODS      += vscan
349 DRV_KMODS      += wc
350 DRV_KMODS      += winlock
351 DRV_KMODS      += wpi
352 DRV_KMODS      += xge
353 DRV_KMODS      += yge
354 DRV_KMODS      += zcons
355 DRV_KMODS      += zyd
356 DRV_KMODS      += simnet
357 DRV_KMODS      += stmf
358 DRV_KMODS      += stmf_sbd
359 DRV_KMODS      += fct
360 DRV_KMODS      += fcoe
361 DRV_KMODS      += fcoet
362 DRV_KMODS      += fcoei
363 DRV_KMODS      += qlt
364 DRV_KMODS      += iscsit
365 DRV_KMODS      += pppt
366 DRV_KMODS      += ncall nsctl sdbc nskern sv
367 DRV_KMODS      += ii rdc rdcsrv rdcstub
368 DRV_KMODS      += iptun

370 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += bmc
371 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += glm
372 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += intel_nhmex
373 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += cpqary3
374 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += marvell88sx
375 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += bcm_sata
376 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += memtest
377 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += mpt
378 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += atiatom
379 $(CLOSED_BUILD)CLOSED_DRV_KMODS      += acpi_toshiba

381 #
382 #      Common code drivers
383 #

384 DRV_KMODS      += afe
385 DRV_KMODS      += atge
386 DRV_KMODS      += bfe
387 DRV_KMODS      += dmfe

```

```

388 DRV_KMODS      += e1000g
389 DRV_KMODS      += efe
390 DRV_KMODS      += elxl
391 DRV_KMODS      += hme
392 DRV_KMODS      += mxfe
393 DRV_KMODS      += nge
394 DRV_KMODS      += pcn
395 DRV_KMODS      += rge
396 DRV_KMODS      += rtls
397 DRV_KMODS      += sfe
398 DRV_KMODS      += amd8111s
399 DRV_KMODS      += igb
400 DRV_KMODS      += ipmi
401 DRV_KMODS      += iprb
402 DRV_KMODS      += ixgbe
403 DRV_KMODS      += vr
404 $(CLOSED_BUILD)CLOSED_DRV_KMODS += ixgb

406 #
407 #      Virtio drivers
408 #

411 # Virtio core
409 DRV_KMODS      += virtio

414 # Virtio block driver
410 DRV_KMODS      += vioblk
411 DRV_KMODS      += vioif

413 # Virtio networking driver
414 DRV_KMODS      += vioif

415 #
416 #      DTrace and DTrace Providers
417 #
418 DRV_KMODS      += dtrace
419 DRV_KMODS      += fbt
420 DRV_KMODS      += lockstat
421 DRV_KMODS      += profile
422 DRV_KMODS      += sdt
423 DRV_KMODS      += systrace
424 DRV_KMODS      += fasttrap
425 DRV_KMODS      += dcpc

427 #
428 #      I/O framework test drivers
429 #
430 DRV_KMODS      += pshot
431 DRV_KMODS      += gen_drv
432 DRV_KMODS      += tvhci tphci tclient
433 DRV_KMODS      += emul64

435 #
436 #      Machine Specific Driver Modules (/kernel/drv):
437 #
438 DRV_KMODS      += options
439 DRV_KMODS      += scsi_vhci
440 DRV_KMODS      += pmcs
441 DRV_KMODS      += pmcs800lfw
442 DRV_KMODS      += arcmsr
443 DRV_KMODS      += fcp
444 DRV_KMODS      += fcip
445 DRV_KMODS      += fcsn
446 DRV_KMODS      += fp
447 DRV_KMODS      += qlc
448 DRV_KMODS      += iscsi

```

```

450 #
451 #      PCMCIA specific module(s)
452 #
453 DRV_KMODS      += pcs
454 DRV_KMODS      += pcata
455 MISC_KMODS     += cardbus
456 $(CLOSED_BUILD)CLOSED_DRV_KMODS += pcser

458 #
459 #      SCSI Enclosure Services driver
460 #
461 DRV_KMODS      += ses

463 #
464 #      USB specific modules
465 #
466 DRV_KMODS      += hid
467 DRV_KMODS      += hwarc hwahc
468 DRV_KMODS      += hubd
469 DRV_KMODS      += uhci
470 DRV_KMODS      += ehci
471 DRV_KMODS      += ohci
472 DRV_KMODS      += usb_mid
473 DRV_KMODS      += usb_ia
474 DRV_KMODS      += scsa2usb
475 DRV_KMODS      += usbprn
476 DRV_KMODS      += ugen
477 DRV_KMODS      += usbser
478 DRV_KMODS      += usbsacm
479 DRV_KMODS      += usbsksp
480 DRV_KMODS      += usbsprl
481 DRV_KMODS      += usb_ac
482 DRV_KMODS      += usb_as
483 DRV_KMODS      += usbskel
484 DRV_KMODS      += usbvc
485 DRV_KMODS      += usbftdi
486 DRV_KMODS      += wusb_df
487 DRV_KMODS      += wusb_ca
488 DRV_KMODS      += usbecm

490 $(CLOSED_BUILD)CLOSED_DRV_KMODS += usbser_edge

492 #
493 #      1394 modules
494 #
495 MISC_KMODS     += s1394 sbp2
496 DRV_KMODS      += hci1394 scsa1394
497 DRV_KMODS      += av1394
498 DRV_KMODS      += dcam1394

500 #
501 #      InfiniBand pseudo drivers
502 #
503 DRV_KMODS      += ib ibp eibnx eoib rdsib sdp iser daplt hermon tavor sol_ucma
504 DRV_KMODS      += sol_umad

506 #
507 #      LVM modules
508 #
509 DRV_KMODS      += md
510 MISC_KMODS     += md_stripe md_hotspares md_mirror md_raid md_trans md_notify
511 MISC_KMODS     += md_sp

513 #
514 #      Brand modules
515 #

```

```

516 BRAND_KMODS      += snl_brand s10_brand

518 #
519 #       Exec Class Modules (/kernel/exec):
520 #
521 EXEC_KMODS        += elfexec intpexec shbinexec javaexec

523 #
524 #       Scheduling Class Modules (/kernel/sched):
525 #
526 SCHED_KMODS      += IA RT TS RT_DPTBL TS_DPTBL FSS FX FX_DPTBL SDC

528 #
529 #       File System Modules (/kernel/fs):
530 #
531 FS_KMODS          += autofs cachefs ctfs dcfs dev devfs fdfs fifofs hsfs lofs
532 FS_KMODS          += mntfs namefs nfs objfs zfs zut
533 FS_KMODS          += pcfs procfs sockfs specfs tmpfs udfs ufs sharefs
534 FS_KMODS          += smbfs

536 #
537 #       Streams Modules (/kernel/strmod):
538 #
539 STRMOD_KMODS     += bufmod conlld dedump ldterm pckt pfmod pipemod
540 STRMOD_KMODS     += ptem redirmod rpcmod rlmod telmod timod
541 STRMOD_KMODS     += sppsasyn sppscomp
542 STRMOD_KMODS     += tirdwr ttcompat
543 STRMOD_KMODS     += usbkbm
544 STRMOD_KMODS     += usbms
545 STRMOD_KMODS     += usbwcm
546 STRMOD_KMODS     += usb_ah
547 STRMOD_KMODS     += dircompat
548 STRMOD_KMODS     += cryptmod
549 STRMOD_KMODS     += vuid2ps2
550 STRMOD_KMODS     += vuid3ps2
551 STRMOD_KMODS     += vuidm3p
552 STRMOD_KMODS     += vuidm4p
553 STRMOD_KMODS     += vuidm5p

555 #
556 #       'System' Modules (/kernel/sys):
557 #
558 SYS_KMODS        += c2audit
559 SYS_KMODS        += doorfs
560 SYS_KMODS        += exacctsys
561 SYS_KMODS        += inst_sync
562 SYS_KMODS        += kaio
563 SYS_KMODS        += msgsys
564 SYS_KMODS        += pipe
565 SYS_KMODS        += portfs
566 SYS_KMODS        += pset
567 SYS_KMODS        += semsys
568 SYS_KMODS        += shmsys
569 SYS_KMODS        += sysacct
570 SYS_KMODS        += acctctl

572 #
573 #       'Misc' Modules (/kernel/misc)
574 #       MISC_KMODS are built both 32-bit and 64-bit
575 #       MISC_KMODS_32 are built only 32-bit
576 #       MISC_KMODS_64 are built only 64-bit
577 #
578 MISC_KMODS       += ac97
579 MISC_KMODS       += acpica
580 MISC_KMODS       += agpmaster
581 MISC_KMODS       += bignum

```

```

582 MISC_KMODS      += bootdev
583 MISC_KMODS      += busra
584 MISC_KMODS      += cmlb
585 MISC_KMODS      += consconfig
586 MISC_KMODS      += ctf
587 MISC_KMODS      += dadk
588 MISC_KMODS      += dcopy
589 MISC_KMODS      += dls
590 MISC_KMODS      += drm
591 MISC_KMODS      += fssnap_if
592 MISC_KMODS      += gda
593 MISC_KMODS      += gld
594 MISC_KMODS      += hidparser
595 MISC_KMODS      += hook
596 MISC_KMODS      += hpcsvc
597 MISC_KMODS      += ibcm
598 MISC_KMODS      += ibdm
599 MISC_KMODS      += ibdma
600 MISC_KMODS      += ibmf
601 MISC_KMODS      += ibt1
602 MISC_KMODS      += idm
603 MISC_KMODS      += idmap
604 MISC_KMODS      += iomulib
605 MISC_KMODS      += ipc
606 MISC_KMODS      += kbtrans
607 MISC_KMODS      += kcf
608 MISC_KMODS      += kgssapi
609 MISC_KMODS      += kmech_dummy
610 MISC_KMODS      += kmech_krb5
611 MISC_KMODS      += ksocket
612 MISC_KMODS      += mac
613 MISC_KMODS      += mii
614 MISC_KMODS      += mwlfw
615 MISC_KMODS      += net80211
616 MISC_KMODS      += nfs_dlboot
617 MISC_KMODS      += nfssrv
618 MISC_KMODS      += neti
619 MISC_KMODS      += pci_autoconfig
620 MISC_KMODS      += pcicfg
621 MISC_KMODS      += pcihp
622 MISC_KMODS      += pcmcia
623 MISC_KMODS      += rpcsec
624 MISC_KMODS      += rpcsec_gss
625 MISC_KMODS      += rsmops
626 MISC_KMODS      += sata
627 MISC_KMODS      += scsi
628 MISC_KMODS      += sda
629 MISC_KMODS      += sol_ofs
630 MISC_KMODS      += spuni
631 MISC_KMODS      += strategy
632 MISC_KMODS      += strplumb
633 MISC_KMODS      += tem
634 MISC_KMODS      += tlimod
635 MISC_KMODS      += usba usba10 usbs49_fw
636 MISC_KMODS      += scsi_vhci_f_sym_hds
637 MISC_KMODS      += scsi_vhci_f_sym
638 MISC_KMODS      += scsi_vhci_f_tpgs
639 MISC_KMODS      += scsi_vhci_f_asym_sun
640 MISC_KMODS      += scsi_vhci_f_tape
641 MISC_KMODS      += scsi_vhci_f_tpgs_tape
642 MISC_KMODS      += fctl
643 MISC_KMODS      += emlxs_fw
644 MISC_KMODS      += qlc_fw_2200
645 MISC_KMODS      += qlc_fw_2300
646 MISC_KMODS      += qlc_fw_2400
647 MISC_KMODS      += qlc_fw_2500

```

```

648 MISC_KMODS      += qlc_fw_6322
649 MISC_KMODS      += qlc_fw_8100
650 MISC_KMODS      += hwal480_fw
651 MISC_KMODS      += uathfw
652 MISC_KMODS      += uwba

654 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += klmmod klmops
655 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_asym_lsi
656 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_asym_emc
657 $(CLOSED_BUILD)CLOSED_MISC_KMODS      += scsi_vhci_f_sym_emc

659 #
660 #       Software Cryptographic Providers (/kernel/crypto):
661 #
662 CRYPTO_KMODS     += aes
663 CRYPTO_KMODS     += arcfour
664 CRYPTO_KMODS     += blowfish
665 CRYPTO_KMODS     += des
666 CRYPTO_KMODS     += ecc
667 CRYPTO_KMODS     += md4
668 CRYPTO_KMODS     += md5
669 CRYPTO_KMODS     += rsa
670 CRYPTO_KMODS     += sha1
671 CRYPTO_KMODS     += sha2
672 CRYPTO_KMODS     += swrand

674 #
675 #       IP Policy Modules (/kernel/ipp)
676 #
677 IPP_KMODS        += dlcosmk
678 IPP_KMODS        += flowacct
679 IPP_KMODS        += ipgpc
680 IPP_KMODS        += dsccpmk
681 IPP_KMODS        += tokenmt
682 IPP_KMODS        += tswtclmt

684 #
685 #       generic-unix module (/kernel/genunix):
686 #
687 GENUNIX_KMODS    += genunix

689 #
690 #       SVVS Testing Modules (/kernel/strmod):
691 #
692 #       These are streams and driver modules which are not to be
693 #       delivered with a released system. However, during development
694 #       it is convenient to build and install the SVVS kernel modules.
695 #
696 SVVS_KMODS       += lmodb lmode lmodr lmodt svvslo tidg tivc tmux

698 $(CLOSED_BUILD)SVVS      += svvs

700 #
701 #       Modules eXcluded from the product:
702 #
703 $(CLOSED_BUILD)CLOSED_XMODS =      \
704     adpu320      \
705     bnx          \
706     bnx         \
707     lsimega     \
708     sdpiib

711 #
712 #       'Dacf' Modules (/kernel/dacf):
713 #

```

```

715 #
716 #       Performance Counter BackEnd modules (/usr/kernel/pcbe)
717 #
718 PCBE_KMODS       += pl23_pcbe p4_pcbe opteron_pcbe core_pcbe

720 #
721 #       MAC-Type Plugin Modules (/kernel/mac)
722 #
723 MAC_KMODS        += mac_6to4
724 MAC_KMODS        += mac_ether
725 MAC_KMODS        += mac_ipv4
726 MAC_KMODS        += mac_ipv6
727 MAC_KMODS        += mac_wifi
728 MAC_KMODS        += mac_ib

730 #
731 #       socketmod (kernel/socketmod)
732 #
733 SOCKET_KMODS     += sockpfp
734 SOCKET_KMODS     += socksctp
735 SOCKET_KMODS     += socksdp
736 SOCKET_KMODS     += sockrds
737 SOCKET_KMODS     += kssif

739 #
740 #       kiconv modules (/kernel/kiconv):
741 #
742 KICONV_KMODS     += kiconv_emea kiconv_ja kiconv_ko kiconv_sc kiconv_tc

744 #
745 #       'Dacf' Modules (/kernel/dacf):
746 #
747 DACF_KMODS       += net_dacf

```

```
*****
```

```
1434 Thu Mar 28 23:16:19 2013
new/usr/src/uts/intel/vioif/Makefile
3644 Add virtio-net support into the Illumos
Reviewed by: Alexey Zaytsev, alexey.zaytsev@gmail.com
Reviewed by: Yuri Pankov, yuri.pankov@nexenta.com
Reviewed by: David Hoppner, 0xfee@gmail.com
*****
```

```
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 #
11 #
12 #
13 # Copyright 2013 Nexenta Inc. All rights reserved.
14 #
15 #
16 #
17 # Path to the base of the uts directory tree (usually /usr/src/uts).
18 #
19 UTSBASE = ../..
20 #
21 #
22 # Define the module and object file sets.
23 #
24 MODULE = vioif
25 OBJECTS = $(VIOIF_OBJS:%=$(OBJS_DIR)/%)
26 LINTS = $(VIOIF_OBJS:%.o=$(LINTS_DIR)/%.ln)
27 ROOTMODULE = $(ROOT_DRV_DIR)/$(MODULE)
28 #
29 #
30 # Include common rules.
31 #
32 include $(UTSBASE)/intel/Makefile.intel
33 #
34 #
35 # Define targets
36 #
37 ALL_TARGET = $(BINARY)
38 LINT_TARGET = $(MODULE).lint
39 INSTALL_TARGET = $(BINARY) $(ROOTMODULE)
40 #
41 #
42 # Overrides
43 #
44 #
45 INC_PATH += -I$(UTSBASE)/common/io/virtio
46 #
47 #
48 # lint pass one enforcement
49 #
50 CFLAGS += $(CCVERBOSE)
51 #
52 # Driver depends on virtio and blkdev
53 #
54 LDFLAGS += -dy -N misc/virtio -N misc/mac
55 #
56 #
57 # Default build targets.
58 #
```

```
59 .KEEP_STATE:
60 #
61 def: $(DEF_DEPS)
62 #
63 all: $(ALL_DEPS)
64 #
65 clean: $(CLEAN_DEPS)
66 #
67 clobber: $(CLOBBER_DEPS)
68 #
69 lint: $(LINT_DEPS)
70 #
71 modlintlib: $(MODLINTLIB_DEPS)
72 #
73 clean.lint: $(CLEAN_LINT_DEPS)
74 #
75 install: $(INSTALL_DEPS)
76 #
77 #
78 # Include common targets.
79 #
80 include $(UTSBASE)/intel/Makefile.targ
```