
5200 Thu Feb 2 13:24:10 2017

new/usr/src/cmd/mandoc/msec.in

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```

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 2015 Nexenta Systems, Inc. All rights reserved.
14 */

16 LINE("1", "User Commands")
17 LINE("1B", "BSD Compatibility Package Commands")
18 LINE("1C", "Communication Commands")
19 LINE("1HAS", "User Commands")
20 LINE("1M", "Maintenance Commands")
21 LINE("1ONBLD", "illumos Build Tools")
22 LINE("1S", "illumos Specific Commands")
23 LINE("2", "System Calls")
24 LINE("3", "Introduction to Library Functions")
25 LINE("3AVL", "AVL Tree Library Functions")
26 LINE("3BSDMALLOC", "BSD Memory Allocation Library")
27 LINE("3BSM", "Security and Auditing Library Functions")
28 LINE("3C", "Standard C Library Functions")
29 LINE("3C+", "C++ Library Functions")
30 LINE("3C_DB", "Threads Debugging Library Functions")
31 LINE("3CFGADM", "Configuration Administration Library Functions")
32 LINE("3COMMUTIL", "Communication Protocol Parser Utilities Library Functions")
33 LINE("3CONTRACT", "Contract Management Library Functions")
34 LINE("3CPC", "CPU Performance Counters Library Functions")
35 LINE("3CURSES", "Curses Library Functions")
36 LINE("3DAT", "Direct Access Transport Library Functions")
37 LINE("3DEVID", "Device ID Library Functions")
38 LINE("3DEVINFO", "Device Information Library Functions")
39 LINE("3DLPI", "Data Link Provider Interface Library Functions")
40 LINE("3DNS_SD", "DNS Service Discovery Library Functions")
41 LINE("3DOOR", "Door Library Functions")
42 LINE("3ELF", "ELF Library Functions")
43 LINE("3EXACCT", "Extended Accounting File Access Library Functions")
44 LINE("3EXT", "Extended Library Functions")
45 LINE("3FCOE", "FCoE Port Management Library Functions")
46 LINE("3FSTYP", "File System Type Identification Library Functions")
47 LINE("3GEN", "String Pattern-Matching Library Functions")
48 LINE("3GSS", "Generic Security Services API Library Functions")
49 LINE("3HEAD", "Headers")
50 LINE("3ISCSI", "iSCSI Management Library Functions")
51 LINE("3KRB", "Kerberos Library Functions")
52 LINE("3KRBS", "MIT Kerberos 5 Library Functions")
53 LINE("3KSTAT", "Kernel Statistics Library Functions")
54 LINE("3KVM", "Kernel VM Library Functions")
55 LINE("3LDAP", "LDAP Library Functions")
56 LINE("3LGRP", "Locality Group Library Functions")
57 LINE("3LIB", "Interface Libraries")
58 LINE("3M", "Mathematical Library Functions")
59 LINE("3MAIL", "User Mailbox Library Functions")
60 LINE("3MALLOC", "Memory Allocation Library Functions")

```

```

61 LINE("3MP", "Multiple Precision Library Functions")
62 LINE("3MPAPI", "Common Multipath Management Library Functions")
63 LINE("3NSL", "Networking Services Library Functions")
64 LINE("3NVPAIR", "Name-value Pair Library Functions")
65 LINE("3PAM", "PAM Library Functions")
66 LINE("3PAPI", "PAPI Library Functions")
67 LINE("3PERL", "Perl Library Functions")
68 LINE("3PICL", "PICL Library Functions")
69 LINE("3PICLTREE", "PICL Plug-In Library Functions")
70 LINE("3POOL", "Pool Configuration Manipulation Library Functions")
71 LINE("3PROC", "Process Control Library Functions")
72 LINE("3PROJECT", "Project Database Access Library Functions")
73 LINE("3RAC", "Remote Asynchronous Calls Library Functions")
74 LINE("3RESOLV", "Resolver Library Functions")
75 LINE("3RPC", "RPC Library Functions")
76 LINE("3RSM", "Remote Shared Memory Library Functions")
77 LINE("3RT", "Realtime Library Functions")
78 LINE("3SASL", "Simple Authentication Security Layer Library Functions")
79 LINE("3SCF", "Service Configuration Facility Library Functions")
80 LINE("3SEC", "File Access Control Library Functions")
81 LINE("3SECDB", "Security Attributes Database Library Functions")
82 LINE("3SIP", "Session Initiation Protocol Library Functions")
83 LINE("3SLP", "Service Location Protocol Library Functions")
84 LINE("3SOCKET", "Sockets Library Functions")
85 LINE("3STMF", "SCSI Target Mode Framework Library Functions")
86 LINE("3SYSEVENT", "System Event Library Functions")
87 LINE("3TECLA", "Interactive Command-line Input Library Functions")
88 LINE("3TNF", "TNF Library Functions")
89 LINE("3TSOL", "Trusted Extensions Library Functions")
90 LINE("3UUID", "Universally Unique Identifier Library Functions")
91 LINE("3VOLMGT", "Volume Management Library Functions")
92 LINE("3XCURSES", "X/Open Curses Library Functions")
93 LINE("3XNET", "X/Open Networking Services Library Functions")
94 LINE("3F", "Fortran Library Routines")
95 LINE("3X", "Miscellaneous Library Functions")
96 LINE("4", "File Formats and Configurations")
97 LINE("5", "Standards, Environments, and Macros")
98 LINE("6", "Games and Demos")
99 LINE("7", "Device and Network Interfaces")
100 LINE("7D", "Devices")
101 LINE("7FS", "File Systems")
102 LINE("7I", "Ioctl Requests")
103 LINE("7IPP", "IP Quality of Service Modules")
104 LINE("7M", "STREAMS Modules")
105 LINE("7P", "Protocols")
106 LINE("8", "Maintenance Procedures")
107 LINE("9", "Kernel Concepts")
107 LINE("9", "Device Driver Interfaces")
108 LINE("9E", "Driver Entry Points")
109 LINE("9F", "Kernel Functions for Drivers")
110 LINE("9P", "Kernel Properties for Drivers")
111 LINE("9S", "Data Structures for Drivers")

```

new/usr/src/man/man9/Intro.9

1

```
*****
483 Thu Feb  2 13:24:12 2017
new/usr/src/man/man9/Intro.9
7831 want vmem manual pages
7832 big theory statements need a place in the manual
*****
1 .Dd Jan 25, 2017
2 .Dt INTRO 9
3 .Os
4 .Sh NAME
5 .Nm Intro ,
6 .Nm intro
7 .Nd introduction to kernel concepts
8 .Sh DESCRIPTION
9 This section outlines high-level concepts of the illumos kernel and its
10 subsystems.  Specific documentation can be found in this section's
11 sub\sections.
12 .Bl -tag -width Ds
13 .It Section 9E
14 Driver Entry Points
15 .It Section 9F
16 Kernel Functions for Drivers
17 .It Section 9P
18 Kernel Properties for Drivers
19 .It Section 9S
20 Data Structures for Drivers
21 .El
22
23 #endif /* ! codereview */
```

new/usr/src/man/man9/Makefile

1

716 Thu Feb 2 13:24:12 2017

new/usr/src/man/man9/Makefile

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```
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
9 # at http://www.illumos.org/license/CDDL.
10 #
```

```
12 #
13 # Copyright 2017, Richard Lowe
13 # Copyright 2011, Richard Lowe
14 # Copyright 2013 Nexenta Systems, Inc. All rights reserved.
15 #
```

```
17 include $(SRC)/Makefile.master
```

```
19 MANSECT= 9
20 MANFILES= Intro.9 \
21 vmem.9
22 #endif /* ! codereview */
```

```
20 MANLINKS= Intro.9 \
21 intro.9
```

```
25 MANLINKS= intro.9
```

```
27 intro.9 := LINKSRC = Intro.9
23 Intro.9 := LINKSRC = ../man9e/Intro.9e
24 intro.9 := LINKSRC = ../man9e/Intro.9e
```

```
29 .KEEP_STATE:
```

```
31 include $(SRC)/man/Makefile.man
```

```
33 install: $(ROOTMANFILES) $(ROOTMANLINKS)
30 install: $(ROOTMANLINKS)
```

```

*****
5813 Thu Feb  2 13:24:13 2017
new/usr/src/man/man9/vmem.9
7831 want vmem manual pages
7832 big theory statements need a place in the manual
*****
1  .\"
2  .\" CDDL HEADER START
3  .\"
4  .\" The contents of this file are subject to the terms of the
5  .\" Common Development and Distribution License (the "License").
6  .\" You may not use this file except in compliance with the License.
7  .\"
8  .\" You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  .\" or http://www.opensolaris.org/os/licensing.
10 .\" See the License for the specific language governing permissions
11 .\" and limitations under the License.
12 .\"
13 .\" When distributing Covered Code, include this CDDL HEADER in each
14 .\" file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 .\" If applicable, add the following below this CDDL HEADER, with the
16 .\" fields enclosed by brackets "[]" replaced with your own identifying
17 .\" information: Portions Copyright [yyyy] [name of copyright owner]
18 .\"
19 .\" CDDL HEADER END
20 .\"
21 .\"
22 .\" Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23 .\" Use is subject to license terms.
24 .\"
25 .\" Copyright (c) 2012, 2015 by Delphix. All rights reserved.
26 .\" Copyright (c) 2012, Joyent, Inc. All rights reserved.
27 .\"
28 .\" The text of this is derived from section 1 of the big theory statement in
29 .\" usr/src/uts/common/os/vmem.c, the traditional location of this text. They
30 .\" should largely be updated in tandem.
31 .Dd Jan 18, 2017
32 .Dt VMEM 9
33 .Os
34 .Sh NAME
35 .Nm vmem
36 .Nd virtual memory allocator
37 .Sh DESCRIPTION
38 .Ss Overview
39 An address space is divided into a number of logically distinct pieces, or
40 .Em arenas :
41 text, data, heap, stack, and so on.
42 Within these
43 arenas we often subdivide further; for example, we use heap addresses
44 not only for the kernel heap
45 .Po
46 .Fn kmem_alloc
47 space
48 .Pc ,
49 but also for DVMA,
50 .Fn bp_mapin ,
51 .Pa /dev/kmem ,
52 and even some device mappings.
53 .Pp
54 The kernel address space, therefore, is most accurately described as
55 a tree of arenas in which each node of the tree
56 .Em imports
57 some subset of its parent.
58 The virtual memory allocator manages these arenas
59 and supports their natural hierarchical structure.
60 .Ss Arenas

```

```

61 An arena is nothing more than a set of integers. These integers most
62 commonly represent virtual addresses, but in fact they can represent
63 anything at all. For example, we could use an arena containing the
64 integers minpid through maxpid to allocate process IDs. For uses of this
65 nature, prefer
66 .Xr id_space 9F
67 instead.
68 .Pp
69 .Fn vmem_create
70 and
71 .Fn vmem_destroy
72 create and destroy vmem arenas. In order to differentiate between arenas used
73 for adresses and arenas used for identifiers, the
74 .Dv VMC_IDENTIFIER
75 flag is passed to
76 .Fn vmem_create .
77 This prevents identifier exhaustion from being diagnosed as general memory
78 failure.
79 .Ss Spans
80 We represent the integers in an arena as a collection of
81 .Em spans ,
82 or contiguous ranges of integers. For example, the kernel heap consists of
83 just one span:
84 .Li "[kernelheap, ekernelheap]" .
85 Spans can be added to an arena in two ways: explicitly, by
86 .Fn vmem_add ,
87 or implicitly, by importing, as described in
88 .Sx Imported Memory
89 below.
90 .Ss Segments
91 Spans are subdivided into
92 .Em segments ,
93 each of which is either allocated or free. A segment, like a span, is a
94 contiguous range of integers. Each allocated segment
95 .Li "[addr, addr + size)"
96 represents exactly one
97 .Li "vmem_alloc(size)"
98 that returned
99 .Sy addr .
100 Free segments represent the space between allocated segments. If two free
101 segments are adjacent, we coalesce them into one larger segment; that is, if
102 segments
103 .Li "[a, b)"
104 and
105 .Li "[b, c)"
106 are both free, we merge them into a single segment
107 .Li "[a, c)" .
108 The segments within a span are linked together in increasing-address
109 order so we can easily determine whether coalescing is possible.
110 .Pp
111 Segments never cross span boundaries. When all segments within an imported
112 span become free, we return the span to its source.
113 .Ss Imported Memory
114 As mentioned in the overview, some arenas are logical subsets of
115 other arenas. For example,
116 .Sy kmem_va_arena
117 (a virtual address cache
118 that satisfies most
119 .Fn kmem_slab_create
120 requests) is just a subset of
121 .Sy heap_arena
122 (the kernel heap) that provides caching for the most common slab sizes. When
123 .Sy kmem_va_arena
124 runs out of virtual memory, it
125 .Em imports more from the heap; we
126 say that

```

```
127 .Sy heap_arena
128 is the
129 .Em "vmem source" for
130 .Sy kmem_va_arena.
131 .Fn vmem_create
132 allows you to specify any existing vmem arena as the source for your new
133 arena. Topologically, since every arena is a child of at most one source, the
134 set of all arenas forms a collection of trees.
135 .Ss Constrained Allocations
136 Some vmem clients are quite picky about the kind of address they want.
137 For example, the DVMA code may need an address that is at a particular
138 phase with respect to some alignment (to get good cache coloring), or
139 that lies within certain limits (the addressable range of a device),
140 or that doesn't cross some boundary (a DMA counter restriction) \(\em
141 or all of the above.
142 .Fn vmem_xalloc
143 allows the client to specify any or all of these constraints.
144 .Ss The Vmem Quantum
145 Every arena has a notion of
146 .Sq quantum ,
147 specified at
148 .Fn vmem_create
149 time, that defines the arena's minimum unit of currency. Most commonly the
150 quantum is either 1 or
151 .Dv PAGESIZE ,
152 but any power of 2 is legal. All vmem allocations are guaranteed to be
153 quantum\(-aligned.
154 .Ss Relationship to the Kernel Memory Allocator
155 Every kmem cache has a vmem arena as its slab supplier. The kernel memory
156 allocator uses
157 .Fn vmem_alloc
158 and
159 .Fn vmem_free
160 to create and destroy slabs.
161 .Sh SEE ALSO
162 .Xr id_space 9F ,
163 .Xr vmem_add 9F ,
164 .Xr vmem_alloc 9F ,
165 .Xr vmem_contains 9F ,
166 .Xr vmem_create 9F ,
167 .Xr vmem_walk 9F
168 .Pp
169 .Rs
170 .%A Jeff Bonwick
171 .%A Jonathan Adams
172 .%T Magazines and vmem: Extending the Slab Allocator to Many CPUs and Arbitrary
173 .%J Proceedings of the 2001 Usenix Conference
174 .%U http://www.usenix.org/event/usenix01/bonwick.html
175 .Re
176 #endif /* ! codereview */
```

```

*****
66677 Thu Feb  2 13:24:14 2017
new/usr/src/man/man9f/Makefile
7831 want vmem manual pages
7832 big theory statements need a place in the manual
*****
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 2017, Richard Lowe
14 # Copyright 2011, Richard Lowe
15 # Copyright 2014 Garrett D'Amore <garrett@damore>
16 # Copyright 2016 Joyent, Inc.
17 # Copyright 2016 Nexenta Systems, Inc.
18 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>

20 include      $(SRC)/Makefile.master

22 MANSECT=     9f

24 MANFILES=    ASSERT.9f
25               Intro.9f
26               OTHERQ.9f
27               RD.9f
28               SAMESTR.9f
29               STRUCT_DECL.9f
30               WR.9f
31               adjmsg.9f
32               allocb.9f
33               atomic_add.9f
34               atomic_and.9f
35               atomic_bits.9f
36               atomic_cas.9f
37               atomic_dec.9f
38               atomic_inc.9f
39               atomic_ops.9f
40               atomic_or.9f
41               atomic_swap.9f
42               avl.9f
43               backq.9f
44               bcanput.9f
45               bcmp.9f
46               bcopy.9f
47               bioclone.9f
48               biodone.9f
49               bioerror.9f
50               biofini.9f
51               bioinit.9f
52               biomodified.9f
53               bioreset.9f
54               biosize.9f
55               biowait.9f
56               bp_copyin.9f
57               bp_copyout.9f
58               bp_mapin.9f
59               bp_mapout.9f

```

```

60               btop.9f
61               btopr.9f
62               bufcall.9f
63               bzero.9f
64               canput.9f
65               canputnext.9f
66               clrbuf.9f
67               cmn_err.9f
68               condvar.9f
69               copyb.9f
70               copyin.9f
71               copymsg.9f
72               copyout.9f
73               csx_AccessConfigurationRegister.9f
74               csx_CS_DDI_Info.9f
75               csx_ConvertSize.9f
76               csx_ConvertSpeed.9f
77               csx_DeregisterClient.9f
78               csx_DupHandle.9f
79               csx_Error2Text.9f
80               csx_Event2Text.9f
81               csx_FreeHandle.9f
82               csx_Get8.9f
83               csx_GetFirstClient.9f
84               csx_GetFirstTuple.9f
85               csx_GetHandleOffset.9f
86               csx_GetMappedAddr.9f
87               csx_GetStatus.9f
88               csx_GetTupleData.9f
89               csx_MakeDeviceNode.9f
90               csx_MapLogSocket.9f
91               csx_MapMemPage.9f
92               csx_ModifyConfiguration.9f
93               csx_ModifyWindow.9f
94               csx_ParseTuple.9f
95               csx_Parse_CISTPL_BATTERY.9f
96               csx_Parse_CISTPL_BYTEORDER.9f
97               csx_Parse_CISTPL_CFTABLE_ENTRY.9f
98               csx_Parse_CISTPL_CONFIG.9f
99               csx_Parse_CISTPL_DATE.9f
100              csx_Parse_CISTPL_DEVICE.9f
101              csx_Parse_CISTPL_DEVICEGEO.9f
102              csx_Parse_CISTPL_DEVICEGEO_A.9f
103              csx_Parse_CISTPL_FORMAT.9f
104              csx_Parse_CISTPL_FUNCE.9f
105              csx_Parse_CISTPL_FUNCID.9f
106              csx_Parse_CISTPL_GEOMETRY.9f
107              csx_Parse_CISTPL_JEDEC_C.9f
108              csx_Parse_CISTPL_LINKTARGET.9f
109              csx_Parse_CISTPL_LONGLINK_A.9f
110              csx_Parse_CISTPL_LONGLINK_MFC.9f
111              csx_Parse_CISTPL_MANFID.9f
112              csx_Parse_CISTPL_ORG.9f
113              csx_Parse_CISTPL_SPCL.9f
114              csx_Parse_CISTPL_SWIL.9f
115              csx_Parse_CISTPL_VERS_1.9f
116              csx_Parse_CISTPL_VERS_2.9f
117              csx_Put8.9f
118              csx_RegisterClient.9f
119              csx_ReleaseConfiguration.9f
120              csx_RepGet8.9f
121              csx_RepPut8.9f
122              csx_RequestConfiguration.9f
123              csx_RequestIO.9f
124              csx_RequestIRQ.9f
125              csx_RequestSocketMask.9f

```

```

126 csx_RequestWindow.9f //
127 csx_ResetFunction.9f //
128 csx_SetEventMask.9f //
129 csx_SetHandleOffset.9f //
130 csx_ValidateCIS.9f //
131 datamsg.9f //
132 ddi_add_event_handler.9f //
133 ddi_add_intr.9f //
134 ddi_add_softintr.9f //
135 ddi_binding_name.9f //
136 ddi_btop.9f //
137 ddi_can_receive_sig.9f //
138 ddi_cb_register.9f //
139 ddi_check_acc_handle.9f //
140 ddi_copyin.9f //
141 ddi_copyout.9f //
142 ddi_create_minor_node.9f //
143 ddi_cred.9f //
144 ddi_dev_is_needed.9f //
145 ddi_dev_is_sid.9f //
146 ddi_dev_nintrs.9f //
147 ddi_dev_nregs.9f //
148 ddi_dev_regsize.9f //
149 ddi_dev_report_fault.9f //
150 ddi_device_copy.9f //
151 ddi_device_zero.9f //
152 ddi_devid_compare.9f //
153 ddi_dma_addr_bind_handle.9f //
154 ddi_dma_alloc_handle.9f //
155 ddi_dma_buf_bind_handle.9f //
156 ddi_dma_burstsizes.9f //
157 ddi_dma_free_handle.9f //
158 ddi_dma_getwin.9f //
159 ddi_dma_mem_alloc.9f //
160 ddi_dma_mem_free.9f //
161 ddi_dma_nextcookie.9f //
162 ddi_dma_numwin.9f //
163 ddi_dma_set_sbus64.9f //
164 ddi_dma_sync.9f //
165 ddi_dma_unbind_handle.9f //
166 ddi_dmae.9f //
167 ddi_driver_major.9f //
168 ddi_driver_name.9f //
169 ddi_enter_critical.9f //
170 ddi_ffs.9f //
171 ddi_fm_acc_err_clear.9f //
172 ddi_fm_acc_err_get.9f //
173 ddi_fm_ereport_post.9f //
174 ddi_fm_handler_register.9f //
175 ddi_fm_init.9f //
176 ddi_fm_service_impact.9f //
177 ddi_get8.9f //
178 ddi_get_cred.9f //
179 ddi_get_devstate.9f //
180 ddi_get_driver_private.9f //
181 ddi_get_eventcookie.9f //
182 ddi_get_instance.9f //
183 ddi_get_kt_did.9f //
184 ddi_get_lbolt.9f //
185 ddi_get_parent.9f //
186 ddi_get_pid.9f //
187 ddi_get_time.9f //
188 ddi_getiminor.9f //
189 ddi_in_panic.9f //
190 ddi_intr_add_handler.9f //
191 ddi_intr_add_softint.9f //

```

```

192 ddi_intr_alloc.9f //
193 ddi_intr_dup_handler.9f //
194 ddi_intr_enable.9f //
195 ddi_intr_get_cap.9f //
196 ddi_intr_get_hilevel_pri.9f //
197 ddi_intr_get_nintrs.9f //
198 ddi_intr_get_pending.9f //
199 ddi_intr_get_pri.9f //
200 ddi_intr_get_supported_types.9f //
201 ddi_intr_hilevel.9f //
202 ddi_intr_set_mask.9f //
203 ddi_intr_set_nreq.9f //
204 ddi_io_get8.9f //
205 ddi_io_put8.9f //
206 ddi_io_rep_get8.9f //
207 ddi_io_rep_put8.9f //
208 ddi_log_sysevent.9f //
209 ddi_map_regs.9f //
210 ddi_mem_get8.9f //
211 ddi_mem_put8.9f //
212 ddi_mem_rep_get8.9f //
213 ddi_mem_rep_put8.9f //
214 ddi_mmap_get_model.9f //
215 ddi_model_convert_from.9f //
216 ddi_modopen.9f //
217 ddi_no_info.9f //
218 ddi_node_name.9f //
219 ddi_peek.9f //
220 ddi_periodic_add.9f //
221 ddi_periodic_delete.9f //
222 ddi_poke.9f //
223 ddi_prop_create.9f //
224 ddi_prop_exists.9f //
225 ddi_prop_get_int.9f //
226 ddi_prop_lookup.9f //
227 ddi_prop_op.9f //
228 ddi_prop_update.9f //
229 ddi_put8.9f //
230 ddi_regs_map_free.9f //
231 ddi_regs_map_setup.9f //
232 ddi_remove_event_handler.9f //
233 ddi_remove_minor_node.9f //
234 ddi_removing_power.9f //
235 ddi_rep_get8.9f //
236 ddi_rep_put8.9f //
237 ddi_report_dev.9f //
238 ddi_root_node.9f //
239 ddi_segmap.9f //
240 ddi_slaveonly.9f //
241 ddi_soft_state.9f //
242 ddi_strtol.9f //
243 ddi_strtoll.9f //
244 ddi_strtoul.9f //
245 ddi_umem_alloc.9f //
246 ddi_umem_iosetup.9f //
247 ddi_umem_lock.9f //
248 delay.9f //
249 devfs_clean.9f //
250 devmap_default_access.9f //
251 devmap_devmem_setup.9f //
252 devmap_do_ctxmgt.9f //
253 devmap_set_ctx_timeout.9f //
254 devmap_setup.9f //
255 devmap_unload.9f //
256 disksort.9f //
257 dlbindack.9f //

```

```

258     drv_getparm.9f
259     drv_hztousec.9f
260     drv_priv.9f
261     drv_uctohz.9f
262     drv_usecwait.9f
263     dupb.9f
264     dupmsg.9f
265     enableok.9f
266     esballoc.9f
267     esbcall.9f
268     firmload.9f
269     flushband.9f
270     flushq.9f
271     freeb.9f
272     freemsg.9f
273     freerbuf.9f
274     freezestr.9f
275     get_pktiopb.9f
276     geterror.9f
277     gethrtime.9f
278     getmajor.9f
279     getminor.9f
280     getq.9f
281     getrbuf.9f
282     gld.9f
283     hook_alloc.9f
284     hook_free.9f
285     id32_alloc.9f
286     id_space.9f
287     inb.9f
288     insq.9f
289     kiconv.9f
290     kiconv_close.9f
291     kiconv_open.9f
292     kiconvstr.9f
293     kmem_alloc.9f
294     kmem_cache_create.9f
295     kstat_create.9f
296     kstat_delete.9f
297     kstat_install.9f
298     kstat_named_init.9f
299     kstat_queue.9f
300     ldi_add_event_handler.9f
301     ldi_aread.9f
302     ldi_devmap.9f
303     ldi_dump.9f
304     ldi_ev_finalize.9f
305     ldi_ev_get_cookie.9f
306     ldi_ev_get_type.9f
307     ldi_ev_notify.9f
308     ldi_ev_register_callbacks.9f
309     ldi_ev_remove_callbacks.9f
310     ldi_get_dev.9f
311     ldi_get_eventcookie.9f
312     ldi_get_size.9f
313     ldi_ident_from_dev.9f
314     ldi_ioctl.9f
315     ldi_open_by_dev.9f
316     ldi_poll.9f
317     ldi_prop_exists.9f
318     ldi_prop_get_int.9f
319     ldi_prop_lookup_int_array.9f
320     ldi_putmsg.9f
321     ldi_read.9f
322     ldi_remove_event_handler.9f
323     ldi_strategy.9f

```

```

324     linkb.9f
325     list_create.9f
326     mac_alloc.9f
327     mac_hcksum_get.9f
328     mac_init_ops.9f
329     mac_link_update.9f
330     mac_lso_get.9f
331     mac_maxsdu_update.9f
332     mac_prop_info.9f
333     mac_register.9f
334     mac_rx.9f
335     mac_tx_update.9f
336     makecom.9f
337     makedevice.9f
338     max.9f
339     mcopyin.9f
340     mcopymsg.9f
341     mcopyout.9f
342     membar_ops.9f
343     memchr.9f
344     merror.9f
345     mexchange.9f
346     min.9f
347     mioc2ack.9f
348     miocack.9f
349     miocnak.9f
350     miocpullup.9f
351     mkiocb.9f
352     mod_install.9f
353     msgdsizes.9f
354     msgpullup.9f
355     msgsize.9f
356     mt-streams.9f
357     mutex.9f
358     net_event_notify_register.9f
359     net_getifname.9f
360     net_getlifaddr.9f
361     net_getmtu.9f
362     net_getnetid.9f
363     net_getpmtuenabled.9f
364     net_hook_register.9f
365     net_hook_unregister.9f
366     net_inject.9f
367     net_inject_alloc.9f
368     net_inject_free.9f
369     net_instance_alloc.9f
370     net_instance_free.9f
371     net_instance_notify_register.9f
372     net_instance_register.9f
373     net_instance_unregister.9f
374     net_ispartialchecksum.9f
375     net_isvalidchecksum.9f
376     net_kstat_create.9f
377     net_kstat_delete.9f
378     net_lifgetnext.9f
379     net_netidtozonid.9f
380     net_phygetnext.9f
381     net_phylookup.9f
382     net_protocol_lookup.9f
383     net_protocol_notify_register.9f
384     net_protocol_release.9f
385     net_protocol_walk.9f
386     net_routeto.9f
387     net_zoneidtonetid.9f
388     netinfo.9f
389     nochpoll.9f

```



```

390     nodev.9f          \
391     noenable.9f      \
392     nulldev.9f       \
393     nvlist_add_boolean.9f \
394     nvlist_alloc.9f  \
395     nvlist_lookup_boolean.9f \
396     nvlist_lookup_nvpair.9f \
397     nvlist_next_nvpair.9f \
398     nvlist_remove.9f \
399     nvpair_value_byte.9f \
400     outb.9f          \
401     pci_config_get8.9f \
402     pci_config_setup.9f \
403     pci_ereport_setup.9f \
404     pci_report_pmcaps.9f \
405     pci_save_config_regs.9f \
406     physio.9f        \
407     pm_busy_component.9f \
408     pm_power_has_changed.9f \
409     pm_raise_power.9f \
410     pm_trans_check.9f \
411     pollhead_clean.9f \
412     pollwakeups.9f \
413     priv_getbyname.9f \
414     priv_policy.9f \
415     proc_signal.9f \
416     ptob.9f          \
417     pullupmsg.9f    \
418     put.9f           \
419     putbq.9f         \
420     putctl.9f        \
421     putctl1.9f       \
422     putnext.9f       \
423     putnextctl.9f   \
424     putnextctl1.9f \
425     putq.9f          \
426     qassociate.9f    \
427     qbufcall.9f     \
428     qenable.9f      \
429     qprocson.9f     \
430     qreply.9f       \
431     qsize.9f         \
432     qtimeout.9f     \
433     qnbufcall.9f    \
434     qntimeout.9f    \
435     qwait.9f        \
436     qwriter.9f      \
437     rmalloc.9f      \
438     rmalloc_wait.9f \
439     rmallocmap.9f   \
440     rmfree.9f       \
441     rmvb.9f         \
442     rmvq.9f         \
443     rwlock.9f       \
444     scsi_abort.9f   \
445     scsi_alloc_consistent_buf.9f \
446     scsi_cname.9f   \
447     scsi_destroy_pkt.9f \
448     scsi_dmaget.9f  \
449     scsi_errmsg.9f  \
450     scsi_ext_sense_fields.9f \
451     scsi_find_sense_descr.9f \
452     scsi_free_consistent_buf.9f \
453     scsi_get_device_type_scsi_options.9f \
454     scsi_get_device_type_string.9f \
455     scsi_hba_attach_setup.9f \

```

```

456     scsi_hba_init.9f \
457     scsi_hba_lookup_capstr.9f \
458     scsi_hba_pkt_alloc.9f \
459     scsi_hba_pkt_comp.9f \
460     scsi_hba_probe.9f \
461     scsi_hba_tran_alloc.9f \
462     scsi_ifgetcap.9f \
463     scsi_init_pkt.9f \
464     scsi_log.9f      \
465     scsi_pktalloc.9f \
466     scsi_poll.9f    \
467     scsi_probe.9f   \
468     scsi_reset.9f   \
469     scsi_reset_notify.9f \
470     scsi_sense_key.9f \
471     scsi_setup_cdb.9f \
472     scsi_slave.9f   \
473     scsi_sync_pkt.9f \
474     scsi_transport.9f \
475     scsi_unprobe.9f \
476     scsi_validate_sense.9f \
477     scsi_vu_errmsg.9f \
478     semaphore.9f   \
479     sprintf.9f     \
480     stoi.9f        \
481     string.9f      \
482     strlog.9f      \
483     strqget.9f     \
484     strqset.9f     \
485     swab.9f        \
486     taskq.9f       \
487     testb.9f       \
488     timeout.9f     \
489     u8_strcmp.9f   \
490     u8_textprep_str.9f \
491     u8_validate.9f \
492     uconv_ul6tou32.9f \
493     uiomove.9f     \
494     unbufcall.9f   \
495     unlinkb.9f     \
496     untimeout.9f   \
497     ureadc.9f      \
498     usb_alloc_request.9f \
499     usb_client_attach.9f \
500     usb_clr_feature.9f \
501     usb_create_pm_components.9f \
502     usb_get_addr.9f \
503     usb_get_alt_if.9f \
504     usb_get_cfg.9f \
505     usb_get_current_frame_number.9f \
506     usb_get_dev_data.9f \
507     usb_get_max_pkts_per_isoc_request.9f \
508     usb_get_status.9f \
509     usb_get_string_descr.9f \
510     usb_handle_remote_wakeup.9f \
511     usb_lookup_ep_data.9f \
512     usb_parse_data.9f \
513     usb_pipe_bulk_xfer.9f \
514     usb_pipe_close.9f \
515     usb_pipe_ctrl_xfer.9f \
516     usb_pipe_drain_reqs.9f \
517     usb_pipe_get_max_bulk_transfer_size.9f \
518     usb_pipe_get_state.9f \
519     usb_pipe_intr_xfer.9f \
520     usb_pipe_isoc_xfer.9f \
521     usb_pipe_open.9f \

```

```

522         usb_pipe_reset.9f          \|
523         usb_pipe_set_private.9f    \|
524         usb_register_hotplug_obs.9f \|
525         usb_reset_device.9f       \|
526         uwritel.9f                 \|
527         va_arg.9f                   \|
528         vmem_add.9f                 \|
529         vmem_alloc.9f              \|
530         vmem_contains.9f          \|
531         vmem_create.9f            \|
532         vmem_walk.9f              \|
527         va_arg.9f

```

```

534 MANLINKS=      AVL_NEXT.9f          \|
535                 AVL_PREV.9f          \|
536                 SIZEOF_PTR.9f        \|
537                 SIZEOF_STRUCT.9f     \|
538                 STRUCT_BUF.9f        \|
539                 STRUCT_FADDR.9f      \|
540                 STRUCT_FGET.9f       \|
541                 STRUCT_FGETP.9f      \|
542                 STRUCT_FSET.9f       \|
543                 STRUCT_FSETP.9f      \|
544                 STRUCT_HANDLE.9f     \|
545                 STRUCT_INIT.9f       \|
546                 STRUCT_SET_HANDLE.9f \|
547                 STRUCT_SIZE.9f       \|
548                 assert.9f           \|
549                 atomic_add_16.9f     \|
550                 atomic_add_16_nv.9f  \|
551                 atomic_add_32.9f     \|
552                 atomic_add_32_nv.9f  \|
553                 atomic_add_64.9f     \|
554                 atomic_add_64_nv.9f  \|
555                 atomic_add_8.9f      \|
556                 atomic_add_8_nv.9f   \|
557                 atomic_add_char.9f   \|
558                 atomic_add_char_nv.9f \|
559                 atomic_add_int.9f    \|
560                 atomic_add_int_nv.9f  \|
561                 atomic_add_long.9f   \|
562                 atomic_add_long_nv.9f \|
563                 atomic_add_ptr.9f     \|
564                 atomic_add_ptr_nv.9f  \|
565                 atomic_add_short.9f   \|
566                 atomic_add_short_nv.9f \|
567                 atomic_and_16.9f     \|
568                 atomic_and_16_nv.9f  \|
569                 atomic_and_32.9f     \|
570                 atomic_and_32_nv.9f  \|
571                 atomic_and_64.9f     \|
572                 atomic_and_64_nv.9f  \|
573                 atomic_and_8.9f      \|
574                 atomic_and_8_nv.9f   \|
575                 atomic_and_uchar.9f  \|
576                 atomic_and_uchar_nv.9f \|
577                 atomic_and_uint.9f   \|
578                 atomic_and_uint_nv.9f \|
579                 atomic_and_ulong.9f  \|
580                 atomic_and_ulong_nv.9f \|
581                 atomic_and_ushort.9f  \|
582                 atomic_and_ushort_nv.9f \|
583                 atomic_cas_16.9f     \|
584                 atomic_cas_32.9f     \|
585                 atomic_cas_64.9f     \|
586                 atomic_cas_8.9f

```

```

587         atomic_cas_ptr.9f          \|
588         atomic_cas_uchar.9f        \|
589         atomic_cas_uint.9f         \|
590         atomic_cas_ulong.9f        \|
591         atomic_cas_ushort.9f       \|
592         atomic_clear_long_excl.9f  \|
593         atomic_dec_16.9f           \|
594         atomic_dec_16_nv.9f        \|
595         atomic_dec_32.9f           \|
596         atomic_dec_32_nv.9f        \|
597         atomic_dec_64.9f           \|
598         atomic_dec_64_nv.9f        \|
599         atomic_dec_8.9f            \|
600         atomic_dec_8_nv.9f         \|
601         atomic_dec_ptr.9f          \|
602         atomic_dec_ptr_nv.9f       \|
603         atomic_dec_uchar.9f        \|
604         atomic_dec_uchar_nv.9f     \|
605         atomic_dec_uint.9f         \|
606         atomic_dec_uint_nv.9f      \|
607         atomic_dec_ulong.9f        \|
608         atomic_dec_ulong_nv.9f     \|
609         atomic_dec_ushort.9f       \|
610         atomic_dec_ushort_nv.9f    \|
611         atomic_inc_16.9f           \|
612         atomic_inc_16_nv.9f        \|
613         atomic_inc_32.9f           \|
614         atomic_inc_32_nv.9f        \|
615         atomic_inc_64.9f           \|
616         atomic_inc_64_nv.9f        \|
617         atomic_inc_8.9f            \|
618         atomic_inc_8_nv.9f         \|
619         atomic_inc_ptr.9f          \|
620         atomic_inc_ptr_nv.9f       \|
621         atomic_inc_uchar.9f        \|
622         atomic_inc_uchar_nv.9f     \|
623         atomic_inc_uint.9f         \|
624         atomic_inc_uint_nv.9f      \|
625         atomic_inc_ulong.9f        \|
626         atomic_inc_ulong_nv.9f     \|
627         atomic_inc_ushort.9f       \|
628         atomic_inc_ushort_nv.9f    \|
629         atomic_or_16.9f            \|
630         atomic_or_16_nv.9f         \|
631         atomic_or_32.9f            \|
632         atomic_or_32_nv.9f         \|
633         atomic_or_64.9f            \|
634         atomic_or_64_nv.9f         \|
635         atomic_or_8.9f             \|
636         atomic_or_8_nv.9f          \|
637         atomic_or_uchar.9f         \|
638         atomic_or_uchar_nv.9f     \|
639         atomic_or_uint.9f          \|
640         atomic_or_uint_nv.9f       \|
641         atomic_or_ulong.9f         \|
642         atomic_or_ulong_nv.9f      \|
643         atomic_or_ushort.9f        \|
644         atomic_or_ushort_nv.9f     \|
645         atomic_set_long_excl.9f    \|
646         atomic_swap_16.9f          \|
647         atomic_swap_32.9f          \|
648         atomic_swap_64.9f          \|
649         atomic_swap_8.9f           \|
650         atomic_swap_ptr.9f         \|
651         atomic_swap_uchar.9f       \|
652         atomic_swap_uint.9f

```

```

653         atomic_swap_ulong.9f
654         atomic_swap_ushort.9f
655         avl_add.9f
656         avl_create.9f
657         avl_destroy.9f
658         avl_destroy_nodes.9f
659         avl_find.9f
660         avl_first.9f
661         avl_insert.9f
662         avl_insert_here.9f
663         avl_is_empty.9f
664         avl_last.9f
665         avl_nearest.9f
666         avl_numnodes.9f
667         avl_remove.9f
668         avl_swap.9f
669         bcanputnext.9f
670         crgetgid.9f
671         crgetgroups.9f
672         crgetngroups.9f
673         crgetrgid.9f
674         crgetruid.9f
675         crgetsgid.9f
676         crgetsuid.9f
677         crgetuid.9f
678         crgetzoneid.9f
679         csx_Get16.9f
680         csx_Get32.9f
681         csx_Get64.9f
682         csx_GetEventMask.9f
683         csx_GetNextClient.9f
684         csx_GetNextTuple.9f
685         csx_Parse_CISTPL_DEVICE_A.9f
686         csx_Parse_CISTPL_DEVICE_OA.9f
687         csx_Parse_CISTPL_DEVICE_OC.9f
688         csx_Parse_CISTPL_JEDEC_A.9f
689         csx_Parse_CISTPL_LONGLINK_C.9f
690         csx_Put16.9f
691         csx_Put32.9f
692         csx_Put64.9f
693         csx_ReleaseIO.9f
694         csx_ReleaseIRQ.9f
695         csx_ReleaseSocketMask.9f
696         csx_ReleaseWindow.9f
697         csx_RemoveDeviceNode.9f
698         csx_RepGet16.9f
699         csx_RepGet32.9f
700         csx_RepGet64.9f
701         csx_RepPut16.9f
702         csx_RepPut32.9f
703         csx_RepPut64.9f
704         cv_broadcast.9f
705         cv_destroy.9f
706         cv_init.9f
707         cv_reltimedwait.9f
708         cv_reltimedwait_sig.9f
709         cv_signal.9f
710         cv_timedwait.9f
711         cv_timedwait_sig.9f
712         cv_wait.9f
713         cv_wait_sig.9f
714         ddi_btopr.9f
715         ddi_cb_unregister.9f
716         ddi_check_dma_handle.9f
717         ddi_devid_free.9f
718         ddi_devid_get.9f

```

```

719         ddi_devid_init.9f
720         ddi_devid_register.9f
721         ddi_devid_sizeof.9f
722         ddi_devid_str_decode.9f
723         ddi_devid_str_encode.9f
724         ddi_devid_str_free.9f
725         ddi_devid_unregister.9f
726         ddi_devid_valid.9f
727         ddi_devmap_segmap.9f
728         ddi_dmae_lstparty.9f
729         ddi_dmae_alloc.9f
730         ddi_dmae_disable.9f
731         ddi_dmae_enable.9f
732         ddi_dmae_getattr.9f
733         ddi_dmae_getcnt.9f
734         ddi_dmae_prog.9f
735         ddi_dmae_release.9f
736         ddi_dmae_stop.9f
737         ddi_exit_critical.9f
738         ddi_fls.9f
739         ddi_fm_capable.9f
740         ddi_fm_dma_err_clear.9f
741         ddi_fm_dma_err_get.9f
742         ddi_fm_fini.9f
743         ddi_fm_handler_unregister.9f
744         ddi_get16.9f
745         ddi_get32.9f
746         ddi_get64.9f
747         ddi_get_iblock_cookie.9f
748         ddi_get_lbolt64.9f
749         ddi_get_name.9f
750         ddi_get_soft_iblock_cookie.9f
751         ddi_get_soft_state.9f
752         ddi_getb.9f
753         ddi_getl.9f
754         ddi_getll.9f
755         ddi_getlongprop.9f
756         ddi_getlongprop_buf.9f
757         ddi_getprop.9f
758         ddi_getproplen.9f
759         ddi_getw.9f
760         ddi_intr_block_disable.9f
761         ddi_intr_block_enable.9f
762         ddi_intr_clr_mask.9f
763         ddi_intr_disable.9f
764         ddi_intr_free.9f
765         ddi_intr_get_navail.9f
766         ddi_intr_get_softint_pri.9f
767         ddi_intr_remove_handler.9f
768         ddi_intr_remove_softint.9f
769         ddi_intr_set_cap.9f
770         ddi_intr_set_pri.9f
771         ddi_intr_set_softint_pri.9f
772         ddi_intr_trigger_softint.9f
773         ddi_io_get16.9f
774         ddi_io_get32.9f
775         ddi_io_getb.9f
776         ddi_io_getl.9f
777         ddi_io_getw.9f
778         ddi_io_put16.9f
779         ddi_io_put32.9f
780         ddi_io_putb.9f
781         ddi_io_putl.9f
782         ddi_io_putw.9f
783         ddi_io_rep_get16.9f
784         ddi_io_rep_get32.9f

```

```

785 ddi_io_rep_getb.9f //
786 ddi_io_rep_getl.9f //
787 ddi_io_rep_getw.9f //
788 ddi_io_rep_put16.9f //
789 ddi_io_rep_put32.9f //
790 ddi_io_rep_putb.9f //
791 ddi_io_rep_putl.9f //
792 ddi_io_rep_putw.9f //
793 ddi_mem_get16.9f //
794 ddi_mem_get32.9f //
795 ddi_mem_get64.9f //
796 ddi_mem_getb.9f //
797 ddi_mem_getl.9f //
798 ddi_mem_getll.9f //
799 ddi_mem_getw.9f //
800 ddi_mem_put16.9f //
801 ddi_mem_put32.9f //
802 ddi_mem_put64.9f //
803 ddi_mem_putb.9f //
804 ddi_mem_putl.9f //
805 ddi_mem_putll.9f //
806 ddi_mem_putw.9f //
807 ddi_mem_rep_get16.9f //
808 ddi_mem_rep_get32.9f //
809 ddi_mem_rep_get64.9f //
810 ddi_mem_rep_getb.9f //
811 ddi_mem_rep_getl.9f //
812 ddi_mem_rep_getll.9f //
813 ddi_mem_rep_getw.9f //
814 ddi_mem_rep_put16.9f //
815 ddi_mem_rep_put32.9f //
816 ddi_mem_rep_put64.9f //
817 ddi_mem_rep_putb.9f //
818 ddi_mem_rep_putl.9f //
819 ddi_mem_rep_putll.9f //
820 ddi_mem_rep_putw.9f //
821 ddi_modclose.9f //
822 ddi_modsym.9f //
823 ddi_peek16.9f //
824 ddi_peek32.9f //
825 ddi_peek64.9f //
826 ddi_peek8.9f //
827 ddi_peekc.9f //
828 ddi_peekd.9f //
829 ddi_peekl.9f //
830 ddi_peeks.9f //
831 ddi_poke16.9f //
832 ddi_poke32.9f //
833 ddi_poke64.9f //
834 ddi_poke8.9f //
835 ddi_pokec.9f //
836 ddi_poked.9f //
837 ddi_pokel.9f //
838 ddi_pokes.9f //
839 ddi_prop_free.9f //
840 ddi_prop_get_int64.9f //
841 ddi_prop_lookup_byte_array.9f //
842 ddi_prop_lookup_int64_array.9f //
843 ddi_prop_lookup_int_array.9f //
844 ddi_prop_lookup_string.9f //
845 ddi_prop_lookup_string_array.9f //
846 ddi_prop_modify.9f //
847 ddi_prop_remove.9f //
848 ddi_prop_remove_all.9f //
849 ddi_prop_undefine.9f //
850 ddi_prop_update_byte_array.9f //

```

```

851 ddi_prop_update_int.9f //
852 ddi_prop_update_int64.9f //
853 ddi_prop_update_int64_array.9f //
854 ddi_prop_update_int_array.9f //
855 ddi_prop_update_string.9f //
856 ddi_prop_update_string_array.9f //
857 ddi_ptob.9f //
858 ddi_put16.9f //
859 ddi_put32.9f //
860 ddi_put64.9f //
861 ddi_putb.9f //
862 ddi_putl.9f //
863 ddi_putll.9f //
864 ddi_putw.9f //
865 ddi_remove_intr.9f //
866 ddi_remove_softintr.9f //
867 ddi_rep_get16.9f //
868 ddi_rep_get32.9f //
869 ddi_rep_get64.9f //
870 ddi_rep_getb.9f //
871 ddi_rep_getl.9f //
872 ddi_rep_getll.9f //
873 ddi_rep_getw.9f //
874 ddi_rep_put16.9f //
875 ddi_rep_put32.9f //
876 ddi_rep_put64.9f //
877 ddi_rep_putb.9f //
878 ddi_rep_putl.9f //
879 ddi_rep_putll.9f //
880 ddi_rep_putw.9f //
881 ddi_segmap_setup.9f //
882 ddi_set_driver_private.9f //
883 ddi_soft_state_fini.9f //
884 ddi_soft_state_free.9f //
885 ddi_soft_state_init.9f //
886 ddi_soft_state_zalloc.9f //
887 ddi_strdup.9f //
888 ddi_strtoull.9f //
889 ddi_taskq_create.9f //
890 ddi_taskq_destroy.9f //
891 ddi_taskq_dispatch.9f //
892 ddi_taskq_resume.9f //
893 ddi_taskq_suspend.9f //
894 ddi_taskq_wait.9f //
895 ddi_trigger_softintr.9f //
896 ddi_umem_free.9f //
897 ddi_umem_unlock.9f //
898 ddi_unmap_regs.9f //
899 desballoc.9f //
900 dev_err.9f //
901 devmap_load.9f //
902 devmap_umem_setup.9f //
903 dlerrorack.9f //
904 dlkack.9f //
905 dlphysaddrack.9f //
906 dluderrorind.9f //
907 firmware_close.9f //
908 firmware_free.9f //
909 firmware_get_size.9f //
910 firmware_malloc.9f //
911 firmware_open.9f //
912 firmware_read.9f //
913 free_pktiopb.9f //
914 gld_intr.9f //
915 gld_mac_alloc.9f //
916 gld_mac_free.9f //

```

```

917      gld_rcv.9f
918      gld_register.9f
919      gld_sched.9f
920      gld_unregister.9f
921      id32_free.9f
922      id32_lookup.9f
923      id_alloc.9f
924      id_alloc_nosleep.9f
925      id_alloc_specific_nosleep.9f
926      id_allocff.9f
927      id_allocff_nosleep.9f
928      id_free.9f
929      id_space_create.9f
930      id_space_destroy.9f
931      id_space_extend.9f
932      inl.9f
933      intro.9f
934      inw.9f
935      kmem_cache_alloc.9f
936      kmem_cache_destroy.9f
937      kmem_cache_free.9f
938      kmem_cache_set_move.9f
939      kmem_free.9f
940      kmem_zalloc.9f
941      kstat_named_setstr.9f
942      kstat_runq_back_to_waitq.9f
943      kstat_runq_enter.9f
944      kstat_runq_exit.9f
945      kstat_waitq_enter.9f
946      kstat_waitq_exit.9f
947      kstat_waitq_to_runq.9f
948      ldi_awrite.9f
949      ldi_close.9f
950      ldi_get_devid.9f
951      ldi_get_minor_name.9f
952      ldi_get_otyp.9f
953      ldi_getmsg.9f
954      ldi_ident_from_dip.9f
955      ldi_ident_from_stream.9f
956      ldi_ident_release.9f
957      ldi_open_by_devid.9f
958      ldi_open_by_name.9f
959      ldi_prop_get_int64.9f
960      ldi_prop_lookup_byte_array.9f
961      ldi_prop_lookup_int64_array.9f
962      ldi_prop_lookup_string.9f
963      ldi_prop_lookup_string_array.9f
964      ldi_write.9f
965      list_destroy.9f
966      list_head.9f
967      list_insert_after.9f
968      list_insert_before.9f
969      list_insert_head.9f
970      list_insert_tail.9f
971      list_is_empty.9f
972      list_link_active.9f
973      list_link_init.9f
974      list_link_replace.9f
975      list_move_tail.9f
976      list_next.9f
977      list_prev.9f
978      list_remove.9f
979      list_remove_head.9f
980      list_remove_tail.9f
981      list_tail.9f
982      mac_fini_ops.9f

```

```

983      mac_free.9f
984      mac_hcksum_set.9f
985      mac_prop_info_set_default_link_flowctrl.9f
986      mac_prop_info_set_default_str.9f
987      mac_prop_info_set_default_uint32.9f
988      mac_prop_info_set_default_uint64.9f
989      mac_prop_info_set_default_uint8.9f
990      mac_prop_info_set_perm.9f
991      mac_prop_info_set_range_uint32.9f
992      mac_unregister.9f
993      makecom_g0.9f
994      makecom_g0_s.9f
995      makecom_g1.9f
996      makecom_g5.9f
997      membar_consumer.9f
998      membar_enter.9f
999      membar_exit.9f
1000     membar_producer.9f
1001     memcmp.9f
1002     memcpy.9f
1003     memmove.9f
1004     memset.9f
1005     minphys.9f
1006     mod_info.9f
1007     mod_modname.9f
1008     mod_remove.9f
1009     mutex_destroy.9f
1010     mutex_enter.9f
1011     mutex_exit.9f
1012     mutex_init.9f
1013     mutex_owned.9f
1014     mutex_tryenter.9f
1015     net_event_notify_unregister.9f
1016     net_instance_notify_unregister.9f
1017     net_instance_protocol_unregister.9f
1018     numtos.9f
1019     nv_alloc_fini.9f
1020     nv_alloc_init.9f
1021     nvlist_add_boolean_array.9f
1022     nvlist_add_boolean_value.9f
1023     nvlist_add_byte.9f
1024     nvlist_add_byte_array.9f
1025     nvlist_add_int16.9f
1026     nvlist_add_int16_array.9f
1027     nvlist_add_int32.9f
1028     nvlist_add_int32_array.9f
1029     nvlist_add_int64.9f
1030     nvlist_add_int64_array.9f
1031     nvlist_add_int8.9f
1032     nvlist_add_int8_array.9f
1033     nvlist_add_nvlist.9f
1034     nvlist_add_nvlist_array.9f
1035     nvlist_add_nvpair.9f
1036     nvlist_add_string.9f
1037     nvlist_add_string_array.9f
1038     nvlist_add_uint16.9f
1039     nvlist_add_uint16_array.9f
1040     nvlist_add_uint32.9f
1041     nvlist_add_uint32_array.9f
1042     nvlist_add_uint64.9f
1043     nvlist_add_uint64_array.9f
1044     nvlist_add_uint8.9f
1045     nvlist_add_uint8_array.9f
1046     nvlist_dup.9f
1047     nvlist_exists.9f
1048     nvlist_free.9f

```

```

1049          nvlist_lookup_boolean_array.9f
1050          nvlist_lookup_boolean_value.9f
1051          nvlist_lookup_byte.9f
1052          nvlist_lookup_byte_array.9f
1053          nvlist_lookup_int16.9f
1054          nvlist_lookup_int16_array.9f
1055          nvlist_lookup_int32.9f
1056          nvlist_lookup_int32_array.9f
1057          nvlist_lookup_int64.9f
1058          nvlist_lookup_int64_array.9f
1059          nvlist_lookup_int8.9f
1060          nvlist_lookup_int8_array.9f
1061          nvlist_lookup_nvlist.9f
1062          nvlist_lookup_nvlist_array.9f
1063          nvlist_lookup_pairs.9f
1064          nvlist_lookup_string.9f
1065          nvlist_lookup_string_array.9f
1066          nvlist_lookup_uint16.9f
1067          nvlist_lookup_uint16_array.9f
1068          nvlist_lookup_uint32.9f
1069          nvlist_lookup_uint32_array.9f
1070          nvlist_lookup_uint64.9f
1071          nvlist_lookup_uint64_array.9f
1072          nvlist_lookup_uint8.9f
1073          nvlist_lookup_uint8_array.9f
1074          nvlist_merge.9f
1075          nvlist_pack.9f
1076          nvlist_remove_all.9f
1077          nvlist_size.9f
1078          nvlist_t.9f
1079          nvlist_unpack.9f
1080          nvlist_xalloc.9f
1081          nvlist_xdup.9f
1082          nvlist_xpack.9f
1083          nvlist_xunpack.9f
1084          nvpair_name.9f
1085          nvpair_type.9f
1086          nvpair_value_boolean_array.9f
1087          nvpair_value_byte_array.9f
1088          nvpair_value_int16.9f
1089          nvpair_value_int16_array.9f
1090          nvpair_value_int32.9f
1091          nvpair_value_int32_array.9f
1092          nvpair_value_int64.9f
1093          nvpair_value_int64_array.9f
1094          nvpair_value_int8.9f
1095          nvpair_value_int8_array.9f
1096          nvpair_value_nvlist.9f
1097          nvpair_value_nvlist_array.9f
1098          nvpair_value_string.9f
1099          nvpair_value_string_array.9f
1100          nvpair_value_uint16.9f
1101          nvpair_value_uint16_array.9f
1102          nvpair_value_uint32.9f
1103          nvpair_value_uint32_array.9f
1104          nvpair_value_uint64.9f
1105          nvpair_value_uint64_array.9f
1106          nvpair_value_uint8.9f
1107          nvpair_value_uint8_array.9f
1108          otherq.9f
1109          outl.9f
1110          outw.9f
1111          pci_config_get16.9f
1112          pci_config_get32.9f
1113          pci_config_get64.9f
1114          pci_config_getb.9f

```

```

1115          pci_config_getl.9f
1116          pci_config_getll.9f
1117          pci_config_getw.9f
1118          pci_config_put16.9f
1119          pci_config_put32.9f
1120          pci_config_put64.9f
1121          pci_config_put8.9f
1122          pci_config_putb.9f
1123          pci_config_putl.9f
1124          pci_config_putll.9f
1125          pci_config_putw.9f
1126          pci_config_teardown.9f
1127          pci_ereport_post.9f
1128          pci_ereport_teardown.9f
1129          pci_restore_config_regs.9f
1130          pm_idle_component.9f
1131          pm_lower_power.9f
1132          priv_policy_choice.9f
1133          priv_policy_only.9f
1134          proc_ref.9f
1135          proc_unref.9f
1136          qprocsoff.9f
1137          qwait_sig.9f
1138          rd.9f
1139          repinsb.9f
1140          repinsd.9f
1141          repinsw.9f
1142          repoutsb.9f
1143          repoutsd.9f
1144          repoutsw.9f
1145          rmallocmap_wait.9f
1146          rmfreemap.9f
1147          rw_destroy.9f
1148          rw_downgrade.9f
1149          rw_enter.9f
1150          rw_exit.9f
1151          rw_init.9f
1152          rw_read_locked.9f
1153          rw_tryenter.9f
1154          rw_tryupgrade.9f
1155          samestr.9f
1156          scsi_dmafree.9f
1157          scsi_dname.9f
1158          scsi_hba_detach.9f
1159          scsi_hba_fini.9f
1160          scsi_hba_pkt_free.9f
1161          scsi_hba_tran_free.9f
1162          scsi_ifsetcap.9f
1163          scsi_mname.9f
1164          scsi_pktfree.9f
1165          scsi_resalloc.9f
1166          scsi_resfree.9f
1167          scsi_rname.9f
1168          scsi_sense_asc.9f
1169          scsi_sense_ascq.9f
1170          scsi_sense_cmdspecific_uint64.9f
1171          scsi_sense_info_uint64.9f
1172          scsi_sname.9f
1173          scsi_unslave.9f
1174          sema_destroy.9f
1175          sema_init.9f
1176          sema_p.9f
1177          sema_p_sig.9f
1178          sema_try.9f
1179          sema_v.9f
1180          snprintf.9f

```

```

1181         strcasecmp.9f
1182         strcat.9f
1183         strchr.9f
1184         strcmp.9f
1185         strcpy.9f
1186         strdup.9f
1187         strfree.9f
1188         strlcat.9f
1189         strlcpy.9f
1190         strlen.9f
1191         strncasecmp.9f
1192         strncat.9f
1193         strncmp.9f
1194         strncpy.9f
1195         strnlen.9f
1196         strrchr.9f
1197         strspn.9f
1198         taskq_suspended.9f
1199         uconv_u16tou8.9f
1200         uconv_u32tou16.9f
1201         uconv_u32tou8.9f
1202         uconv_u8tou16.9f
1203         uconv_u8tou32.9f
1204         unfreezestr.9f
1205         usb_alloc_bulk_req.9f
1206         usb_alloc_ctrl_req.9f
1207         usb_alloc_intr_req.9f
1208         usb_alloc_isoc_req.9f
1209         usb_client_detach.9f
1210         usb_free_bulk_req.9f
1211         usb_free_ctrl_req.9f
1212         usb_free_descr_tree.9f
1213         usb_free_dev_data.9f
1214         usb_free_intr_req.9f
1215         usb_free_isoc_req.9f
1216         usb_get_if_number.9f
1217         usb_owns_device.9f
1218         usb_pipe_ctrl_xfer_wait.9f
1219         usb_pipe_get_private.9f
1220         usb_pipe_stop_intr_polling.9f
1221         usb_pipe_stop_isoc_polling.9f
1222         usb_print_descr_tree.9f
1223         usb_set_alt_if.9f
1224         usb_set_cfg.9f
1225         usb_unregister_hotplug_obs.9f
1226         va_copy.9f
1227         va_end.9f
1228         va_start.9f
1229         vcmn_err.9f
1230         vmem_destroy.9f
1231         vmem_free.9f
1232         vmem_size.9f
1233         vmem_xalloc.9f
1234         vmem_xcreate.9f
1235         vmem_xfree.9f
1236 #endif /* ! codereview */
1237         vsnprintf.9f
1238         vsprintf.9f
1239         vzcmn_err.9f
1240         wr.9f
1241         zcmn_err.9f

1243 assert.9f                := LINKSRC = ASSERT.9f
1245 intro.9f                 := LINKSRC = Intro.9f

```

```

1247 otherq.9f                := LINKSRC = OTHERQ.9f
1249 rd.9f                     := LINKSRC = RD.9f
1251 samestr.9f                := LINKSRC = SAMESTR.9f

1253 SIZEOF_PTR.9f             := LINKSRC = STRUCT_DECL.9f
1254 SIZEOF_STRUCT.9f         := LINKSRC = STRUCT_DECL.9f
1255 STRUCT_BUF.9f             := LINKSRC = STRUCT_DECL.9f
1256 STRUCT_FADDR.9f          := LINKSRC = STRUCT_DECL.9f
1257 STRUCT_FGET.9f           := LINKSRC = STRUCT_DECL.9f
1258 STRUCT_FGETP.9f          := LINKSRC = STRUCT_DECL.9f
1259 STRUCT_FSET.9f           := LINKSRC = STRUCT_DECL.9f
1260 STRUCT_FSETP.9f          := LINKSRC = STRUCT_DECL.9f
1261 STRUCT_HANDLE.9f         := LINKSRC = STRUCT_DECL.9f
1262 STRUCT_INIT.9f           := LINKSRC = STRUCT_DECL.9f
1263 STRUCT_SET_HANDLE.9f     := LINKSRC = STRUCT_DECL.9f
1264 STRUCT_SIZE.9f           := LINKSRC = STRUCT_DECL.9f

1266 wr.9f                     := LINKSRC = WR.9f

1268 atomic_add_16.9f          := LINKSRC = atomic_add.9f
1269 atomic_add_16_nv.9f      := LINKSRC = atomic_add.9f
1270 atomic_add_32.9f         := LINKSRC = atomic_add.9f
1271 atomic_add_32_nv.9f      := LINKSRC = atomic_add.9f
1272 atomic_add_64.9f         := LINKSRC = atomic_add.9f
1273 atomic_add_64_nv.9f      := LINKSRC = atomic_add.9f
1274 atomic_add_8.9f          := LINKSRC = atomic_add.9f
1275 atomic_add_8_nv.9f       := LINKSRC = atomic_add.9f
1276 atomic_add_char.9f       := LINKSRC = atomic_add.9f
1277 atomic_add_char_nv.9f    := LINKSRC = atomic_add.9f
1278 atomic_add_int.9f        := LINKSRC = atomic_add.9f
1279 atomic_add_int_nv.9f     := LINKSRC = atomic_add.9f
1280 atomic_add_long.9f       := LINKSRC = atomic_add.9f
1281 atomic_add_long_nv.9f    := LINKSRC = atomic_add.9f
1282 atomic_add_ptr.9f        := LINKSRC = atomic_add.9f
1283 atomic_add_ptr_nv.9f     := LINKSRC = atomic_add.9f
1284 atomic_add_short.9f      := LINKSRC = atomic_add.9f
1285 atomic_add_short_nv.9f   := LINKSRC = atomic_add.9f
1286 atomic_and_16.9f         := LINKSRC = atomic_and.9f
1287 atomic_and_16_nv.9f     := LINKSRC = atomic_and.9f
1288 atomic_and_32.9f         := LINKSRC = atomic_and.9f
1289 atomic_and_32_nv.9f     := LINKSRC = atomic_and.9f
1290 atomic_and_64.9f         := LINKSRC = atomic_and.9f
1291 atomic_and_64_nv.9f     := LINKSRC = atomic_and.9f
1292 atomic_and_8.9f          := LINKSRC = atomic_and.9f
1293 atomic_and_8_nv.9f       := LINKSRC = atomic_and.9f
1294 atomic_and_uchar.9f      := LINKSRC = atomic_and.9f
1295 atomic_and_uchar_nv.9f   := LINKSRC = atomic_and.9f
1296 atomic_and_uint.9f       := LINKSRC = atomic_and.9f
1297 atomic_and_uint_nv.9f    := LINKSRC = atomic_and.9f
1298 atomic_and_ulong.9f     := LINKSRC = atomic_and.9f
1299 atomic_and_ulong_nv.9f   := LINKSRC = atomic_and.9f
1300 atomic_and_ushort.9f     := LINKSRC = atomic_and.9f
1301 atomic_and_ushort_nv.9f  := LINKSRC = atomic_and.9f

1303 atomic_clear_long_excl.9f := LINKSRC = atomic_bits.9f
1304 atomic_set_long_excl.9f  := LINKSRC = atomic_bits.9f

1306 atomic_cas_16.9f         := LINKSRC = atomic_cas.9f
1307 atomic_cas_32.9f         := LINKSRC = atomic_cas.9f
1308 atomic_cas_64.9f         := LINKSRC = atomic_cas.9f
1309 atomic_cas_8.9f          := LINKSRC = atomic_cas.9f
1310 atomic_cas_ptr.9f        := LINKSRC = atomic_cas.9f
1311 atomic_cas_uchar.9f      := LINKSRC = atomic_cas.9f
1312 atomic_cas_uint.9f       := LINKSRC = atomic_cas.9f

```

```

1313 atomic_cas_ulong.9f      == LINKSRC = atomic_cas.9f
1314 atomic_cas_ushort.9f    == LINKSRC = atomic_cas.9f

1316 atomic_dec_16.9f        == LINKSRC = atomic_dec.9f
1317 atomic_dec_16_nv.9f     == LINKSRC = atomic_dec.9f
1318 atomic_dec_32.9f        == LINKSRC = atomic_dec.9f
1319 atomic_dec_32_nv.9f     == LINKSRC = atomic_dec.9f
1320 atomic_dec_64.9f        == LINKSRC = atomic_dec.9f
1321 atomic_dec_64_nv.9f     == LINKSRC = atomic_dec.9f
1322 atomic_dec_8.9f         == LINKSRC = atomic_dec.9f
1323 atomic_dec_8_nv.9f      == LINKSRC = atomic_dec.9f
1324 atomic_dec_ptr.9f       == LINKSRC = atomic_dec.9f
1325 atomic_dec_ptr_nv.9f    == LINKSRC = atomic_dec.9f
1326 atomic_dec_uchar.9f     == LINKSRC = atomic_dec.9f
1327 atomic_dec_uchar_nv.9f == LINKSRC = atomic_dec.9f
1328 atomic_dec_uint.9f      == LINKSRC = atomic_dec.9f
1329 atomic_dec_uint_nv.9f   == LINKSRC = atomic_dec.9f
1330 atomic_dec_ulong.9f     == LINKSRC = atomic_dec.9f
1331 atomic_dec_ulong_nv.9f  == LINKSRC = atomic_dec.9f
1332 atomic_dec_ushort.9f   == LINKSRC = atomic_dec.9f
1333 atomic_dec_ushort_nv.9f == LINKSRC = atomic_dec.9f

1335 atomic_inc_16.9f        == LINKSRC = atomic_inc.9f
1336 atomic_inc_16_nv.9f    == LINKSRC = atomic_inc.9f
1337 atomic_inc_32.9f        == LINKSRC = atomic_inc.9f
1338 atomic_inc_32_nv.9f    == LINKSRC = atomic_inc.9f
1339 atomic_inc_64.9f        == LINKSRC = atomic_inc.9f
1340 atomic_inc_64_nv.9f    == LINKSRC = atomic_inc.9f
1341 atomic_inc_8.9f         == LINKSRC = atomic_inc.9f
1342 atomic_inc_8_nv.9f     == LINKSRC = atomic_inc.9f
1343 atomic_inc_ptr.9f       == LINKSRC = atomic_inc.9f
1344 atomic_inc_ptr_nv.9f   == LINKSRC = atomic_inc.9f
1345 atomic_inc_uchar.9f    == LINKSRC = atomic_inc.9f
1346 atomic_inc_uchar_nv.9f == LINKSRC = atomic_inc.9f
1347 atomic_inc_uint.9f     == LINKSRC = atomic_inc.9f
1348 atomic_inc_uint_nv.9f  == LINKSRC = atomic_inc.9f
1349 atomic_inc_ulong.9f     == LINKSRC = atomic_inc.9f
1350 atomic_inc_ulong_nv.9f == LINKSRC = atomic_inc.9f
1351 atomic_inc_ushort.9f   == LINKSRC = atomic_inc.9f
1352 atomic_inc_ushort_nv.9f == LINKSRC = atomic_inc.9f

1354 atomic_or_16.9f        == LINKSRC = atomic_or.9f
1355 atomic_or_16_nv.9f    == LINKSRC = atomic_or.9f
1356 atomic_or_32.9f        == LINKSRC = atomic_or.9f
1357 atomic_or_32_nv.9f    == LINKSRC = atomic_or.9f
1358 atomic_or_64.9f        == LINKSRC = atomic_or.9f
1359 atomic_or_64_nv.9f    == LINKSRC = atomic_or.9f
1360 atomic_or_8.9f         == LINKSRC = atomic_or.9f
1361 atomic_or_8_nv.9f      == LINKSRC = atomic_or.9f
1362 atomic_or_uchar.9f     == LINKSRC = atomic_or.9f
1363 atomic_or_uchar_nv.9f  == LINKSRC = atomic_or.9f
1364 atomic_or_uint.9f      == LINKSRC = atomic_or.9f
1365 atomic_or_uint_nv.9f  == LINKSRC = atomic_or.9f
1366 atomic_or_ulong.9f     == LINKSRC = atomic_or.9f
1367 atomic_or_ulong_nv.9f == LINKSRC = atomic_or.9f
1368 atomic_or_ushort.9f   == LINKSRC = atomic_or.9f
1369 atomic_or_ushort_nv.9f == LINKSRC = atomic_or.9f

1371 atomic_swap_16.9f      == LINKSRC = atomic_swap.9f
1372 atomic_swap_32.9f      == LINKSRC = atomic_swap.9f
1373 atomic_swap_64.9f      == LINKSRC = atomic_swap.9f
1374 atomic_swap_8.9f       == LINKSRC = atomic_swap.9f
1375 atomic_swap_ptr.9f     == LINKSRC = atomic_swap.9f
1376 atomic_swap_uchar.9f  == LINKSRC = atomic_swap.9f
1377 atomic_swap_uint.9f    == LINKSRC = atomic_swap.9f
1378 atomic_swap_ulong.9f   == LINKSRC = atomic_swap.9f

```

```

1379 atomic_swap_ushort.9f  == LINKSRC = atomic_swap.9f

1381 avl_add.9f              == LINKSRC = avl.9f
1382 avl_create.9f          == LINKSRC = avl.9f
1383 avl_destroy.9f         == LINKSRC = avl.9f
1384 avl_destroy_nodes.9f   == LINKSRC = avl.9f
1385 avl_find.9f            == LINKSRC = avl.9f
1386 avl_first.9f           == LINKSRC = avl.9f
1387 avl_insert.9f          == LINKSRC = avl.9f
1388 avl_insert_here.9f     == LINKSRC = avl.9f
1389 avl_is_empty.9f       == LINKSRC = avl.9f
1390 avl_last.9f            == LINKSRC = avl.9f
1391 avl_nearest.9f         == LINKSRC = avl.9f
1392 avl_numnodes.9f        == LINKSRC = avl.9f
1393 avl_remove.9f          == LINKSRC = avl.9f
1394 avl_swap.9f            == LINKSRC = avl.9f
1395 AVL_NEXT.9f           == LINKSRC = avl.9f
1396 AVL_PREV.9f           == LINKSRC = avl.9f

1398 dev_err.9f             == LINKSRC = cmn_err.9f
1399 vcmn_err.9f            == LINKSRC = cmn_err.9f
1400 vzcmn_err.9f           == LINKSRC = cmn_err.9f
1401 zcmn_err.9f           == LINKSRC = cmn_err.9f

1403 cv_broadcast.9f        == LINKSRC = condvar.9f
1404 cv_destroy.9f          == LINKSRC = condvar.9f
1405 cv_init.9f             == LINKSRC = condvar.9f
1406 cv_reltimedwait.9f    == LINKSRC = condvar.9f
1407 cv_reltimedwait_sig.9f == LINKSRC = condvar.9f
1408 cv_signal.9f           == LINKSRC = condvar.9f
1409 cv_timedwait.9f       == LINKSRC = condvar.9f
1410 cv_timedwait_sig.9f   == LINKSRC = condvar.9f
1411 cv_wait.9f            == LINKSRC = condvar.9f
1412 cv_wait_sig.9f        == LINKSRC = condvar.9f

1414 csx_Get16.9f           == LINKSRC = csx_Get8.9f
1415 csx_Get32.9f          == LINKSRC = csx_Get8.9f
1416 csx_Get64.9f          == LINKSRC = csx_Get8.9f

1418 csx_GetNextClient.9f  == LINKSRC = csx_GetFirstClient.9f
1420 csx_GetNextTuple.9f  == LINKSRC = csx_GetFirstTuple.9f

1422 csx_RemoveDeviceNode.9f == LINKSRC = csx_MakeDeviceNode.9f

1424 csx_Parse_CISTPL_DEVICE_A.9f == LINKSRC = csx_Parse_CISTPL_DEVICE.9f
1425 csx_Parse_CISTPL_DEVICE_OA.9f == LINKSRC = csx_Parse_CISTPL_DEVICE.9f
1426 csx_Parse_CISTPL_DEVICE_OC.9f == LINKSRC = csx_Parse_CISTPL_DEVICE.9f

1428 csx_Parse_CISTPL_JEDEC_A.9f == LINKSRC = csx_Parse_CISTPL_JEDEC_C.9f

1430 csx_Parse_CISTPL_LOGLINK_C.9f == LINKSRC = csx_Parse_CISTPL_LOGLINK_A

1432 csx_Put16.9f          == LINKSRC = csx_Put8.9f
1433 csx_Put32.9f          == LINKSRC = csx_Put8.9f
1434 csx_Put64.9f          == LINKSRC = csx_Put8.9f

1436 csx_RepGet16.9f      == LINKSRC = csx_RepGet8.9f
1437 csx_RepGet32.9f      == LINKSRC = csx_RepGet8.9f
1438 csx_RepGet64.9f      == LINKSRC = csx_RepGet8.9f

1440 csx_RepPut16.9f      == LINKSRC = csx_RepPut8.9f
1441 csx_RepPut32.9f      == LINKSRC = csx_RepPut8.9f
1442 csx_RepPut64.9f      == LINKSRC = csx_RepPut8.9f

1444 csx_ReleaseIO.9f     == LINKSRC = csx_RequestIO.9f

```



```

1446 csx_ReleaseIRQ.9f           == LINKSRC = csx_RequestIRQ.9f
1448 csx_ReleaseSocketMask.9f   == LINKSRC = csx_RequestSocketMask.9f
1450 csx_ReleaseWindow.9f       == LINKSRC = csx_RequestWindow.9f
1452 csx_GetEventMask.9f        == LINKSRC = csx_SetEventMask.9f
1454 ddi_get_iblock_cookie.9f    == LINKSRC = ddi_add_intr.9f
1455 ddi_remove_intr.9f          == LINKSRC = ddi_add_intr.9f
1457 ddi_get_soft_iblock_cookie.9f == LINKSRC = ddi_add_softintr.9f
1458 ddi_remove_softintr.9f      == LINKSRC = ddi_add_softintr.9f
1459 ddi_trigger_softintr.9f     == LINKSRC = ddi_add_softintr.9f
1461 ddi_get_name.9f             == LINKSRC = ddi_binding_name.9f
1463 ddi_btopr.9f                 == LINKSRC = ddi_btop.9f
1464 ddi_ptob.9f                  == LINKSRC = ddi_btop.9f
1466 ddi_cb_unregister.9f        == LINKSRC = ddi_cb_register.9f
1468 ddi_check_dma_handle.9f     == LINKSRC = ddi_check_acc_handle.9f
1470 bcanputnext.9f             == LINKSRC = canputnext.9f
1472 crgetgid.9f                 == LINKSRC = ddi_cred.9f
1473 crgetgroups.9f              == LINKSRC = ddi_cred.9f
1474 crgetngroups.9f             == LINKSRC = ddi_cred.9f
1475 crgetrgid.9f                == LINKSRC = ddi_cred.9f
1476 crgetruid.9f                == LINKSRC = ddi_cred.9f
1477 crgetsgid.9f               == LINKSRC = ddi_cred.9f
1478 crgetsuid.9f               == LINKSRC = ddi_cred.9f
1479 crgetuid.9f                 == LINKSRC = ddi_cred.9f
1480 crgetzoneid.9f             == LINKSRC = ddi_cred.9f
1482 ddi_devid_free.9f           == LINKSRC = ddi_devid_compare.9f
1483 ddi_devid_get.9f            == LINKSRC = ddi_devid_compare.9f
1484 ddi_devid_init.9f           == LINKSRC = ddi_devid_compare.9f
1485 ddi_devid_register.9f       == LINKSRC = ddi_devid_compare.9f
1486 ddi_devid_sizeof.9f         == LINKSRC = ddi_devid_compare.9f
1487 ddi_devid_str_decode.9f     == LINKSRC = ddi_devid_compare.9f
1488 ddi_devid_str_encode.9f     == LINKSRC = ddi_devid_compare.9f
1489 ddi_devid_str_free.9f       == LINKSRC = ddi_devid_compare.9f
1490 ddi_devid_unregister.9f     == LINKSRC = ddi_devid_compare.9f
1491 ddi_devid_valid.9f          == LINKSRC = ddi_devid_compare.9f
1493 ddi_dmae_1stparty.9f        == LINKSRC = ddi_dmae.9f
1494 ddi_dmae_alloc.9f           == LINKSRC = ddi_dmae.9f
1495 ddi_dmae_disable.9f         == LINKSRC = ddi_dmae.9f
1496 ddi_dmae_enable.9f         == LINKSRC = ddi_dmae.9f
1497 ddi_dmae_getattr.9f        == LINKSRC = ddi_dmae.9f
1498 ddi_dmae_getcnt.9f          == LINKSRC = ddi_dmae.9f
1499 ddi_dmae_prog.9f            == LINKSRC = ddi_dmae.9f
1500 ddi_dmae_release.9f         == LINKSRC = ddi_dmae.9f
1501 ddi_dmae_stop.9f            == LINKSRC = ddi_dmae.9f
1503 ddi_exit_critical.9f        == LINKSRC = ddi_enter_critical.9f
1505 ddi_fls.9f                  == LINKSRC = ddi_ffs.9f
1507 ddi_fm_dma_err_clear.9f     == LINKSRC = ddi_fm_acc_err_clear.9f
1509 ddi_fm_dma_err_get.9f       == LINKSRC = ddi_fm_acc_err_get.9f

```

```

1511 ddi_fm_handler_unregister.9f == LINKSRC = ddi_fm_handler_register.9f
1513 ddi_fm_capable.9f           == LINKSRC = ddi_fm_init.9f
1514 ddi_fm_fini.9f              == LINKSRC = ddi_fm_init.9f
1516 ddi_get16.9f                == LINKSRC = ddi_get8.9f
1517 ddi_get32.9f                == LINKSRC = ddi_get8.9f
1518 ddi_get64.9f                == LINKSRC = ddi_get8.9f
1519 ddi_getb.9f                 == LINKSRC = ddi_get8.9f
1520 ddi_getl.9f                  == LINKSRC = ddi_get8.9f
1521 ddi_getll.9f                == LINKSRC = ddi_get8.9f
1522 ddi_getw.9f                 == LINKSRC = ddi_get8.9f
1524 ddi_set_driver_private.9f   == LINKSRC = ddi_get_driver_private.9f
1526 ddi_get_lbolt64.9f         == LINKSRC = ddi_get_lbolt.9f
1528 ddi_intr_remove_handler.9f  == LINKSRC = ddi_intr_add_handler.9f
1530 ddi_intr_get_softint_pri.9f == LINKSRC = ddi_intr_add_softint.9f
1531 ddi_intr_remove_softint.9f == LINKSRC = ddi_intr_add_softint.9f
1532 ddi_intr_set_softint_pri.9f == LINKSRC = ddi_intr_add_softint.9f
1533 ddi_intr_trigger_softint.9f == LINKSRC = ddi_intr_add_softint.9f
1535 ddi_intr_free.9f            == LINKSRC = ddi_intr_alloc.9f
1537 ddi_intr_block_disable.9f   == LINKSRC = ddi_intr_enable.9f
1538 ddi_intr_block_enable.9f    == LINKSRC = ddi_intr_enable.9f
1539 ddi_intr_disable.9f         == LINKSRC = ddi_intr_enable.9f
1541 ddi_intr_set_cap.9f          == LINKSRC = ddi_intr_get_cap.9f
1543 ddi_intr_get_navail.9f      == LINKSRC = ddi_intr_get_nintrs.9f
1545 ddi_intr_set_pri.9f         == LINKSRC = ddi_intr_get_pri.9f
1547 ddi_intr_clr_mask.9f        == LINKSRC = ddi_intr_set_mask.9f
1549 ddi_io_get16.9f             == LINKSRC = ddi_io_get8.9f
1550 ddi_io_get32.9f            == LINKSRC = ddi_io_get8.9f
1551 ddi_io_getb.9f              == LINKSRC = ddi_io_get8.9f
1552 ddi_io_getl.9f             == LINKSRC = ddi_io_get8.9f
1553 ddi_io_getw.9f             == LINKSRC = ddi_io_get8.9f
1555 ddi_io_put16.9f            == LINKSRC = ddi_io_put8.9f
1556 ddi_io_put32.9f            == LINKSRC = ddi_io_put8.9f
1557 ddi_io_putb.9f              == LINKSRC = ddi_io_put8.9f
1558 ddi_io_putl.9f             == LINKSRC = ddi_io_put8.9f
1559 ddi_io_putw.9f             == LINKSRC = ddi_io_put8.9f
1561 ddi_io_rep_get16.9f         == LINKSRC = ddi_io_rep_get8.9f
1562 ddi_io_rep_get32.9f        == LINKSRC = ddi_io_rep_get8.9f
1563 ddi_io_rep_getb.9f         == LINKSRC = ddi_io_rep_get8.9f
1564 ddi_io_rep_getl.9f         == LINKSRC = ddi_io_rep_get8.9f
1565 ddi_io_rep_getw.9f         == LINKSRC = ddi_io_rep_get8.9f
1567 ddi_io_rep_put16.9f        == LINKSRC = ddi_io_rep_put8.9f
1568 ddi_io_rep_put32.9f        == LINKSRC = ddi_io_rep_put8.9f
1569 ddi_io_rep_putb.9f         == LINKSRC = ddi_io_rep_put8.9f
1570 ddi_io_rep_putl.9f         == LINKSRC = ddi_io_rep_put8.9f
1571 ddi_io_rep_putw.9f         == LINKSRC = ddi_io_rep_put8.9f
1573 ddi_unmap_regs.9f          == LINKSRC = ddi_map_regs.9f
1575 ddi_mem_get16.9f            == LINKSRC = ddi_mem_get8.9f
1576 ddi_mem_get32.9f            == LINKSRC = ddi_mem_get8.9f

```

```

1577 ddi_mem_get64.9f      := LINKSRC = ddi_mem_get8.9f
1578 ddi_mem_getb.9f       := LINKSRC = ddi_mem_get8.9f
1579 ddi_mem_getl.9f        := LINKSRC = ddi_mem_get8.9f
1580 ddi_mem_getll.9f       := LINKSRC = ddi_mem_get8.9f
1581 ddi_mem_getw.9f        := LINKSRC = ddi_mem_get8.9f

1583 ddi_mem_put16.9f      := LINKSRC = ddi_mem_put8.9f
1584 ddi_mem_put32.9f      := LINKSRC = ddi_mem_put8.9f
1585 ddi_mem_put64.9f      := LINKSRC = ddi_mem_put8.9f
1586 ddi_mem_putb.9f       := LINKSRC = ddi_mem_put8.9f
1587 ddi_mem_putl.9f       := LINKSRC = ddi_mem_put8.9f
1588 ddi_mem_putll.9f      := LINKSRC = ddi_mem_put8.9f
1589 ddi_mem_putw.9f       := LINKSRC = ddi_mem_put8.9f

1591 ddi_mem_rep_get16.9f  := LINKSRC = ddi_mem_rep_get8.9f
1592 ddi_mem_rep_get32.9f := LINKSRC = ddi_mem_rep_get8.9f
1593 ddi_mem_rep_get64.9f := LINKSRC = ddi_mem_rep_get8.9f
1594 ddi_mem_rep_getb.9f   := LINKSRC = ddi_mem_rep_get8.9f
1595 ddi_mem_rep_getl.9f   := LINKSRC = ddi_mem_rep_get8.9f
1596 ddi_mem_rep_getll.9f  := LINKSRC = ddi_mem_rep_get8.9f
1597 ddi_mem_rep_getw.9f   := LINKSRC = ddi_mem_rep_get8.9f

1599 ddi_mem_rep_put16.9f  := LINKSRC = ddi_mem_rep_put8.9f
1600 ddi_mem_rep_put32.9f := LINKSRC = ddi_mem_rep_put8.9f
1601 ddi_mem_rep_put64.9f := LINKSRC = ddi_mem_rep_put8.9f
1602 ddi_mem_rep_putb.9f   := LINKSRC = ddi_mem_rep_put8.9f
1603 ddi_mem_rep_putl.9f   := LINKSRC = ddi_mem_rep_put8.9f
1604 ddi_mem_rep_putll.9f  := LINKSRC = ddi_mem_rep_put8.9f
1605 ddi_mem_rep_putw.9f   := LINKSRC = ddi_mem_rep_put8.9f

1607 ddi_modclose.9f      := LINKSRC = ddi_modopen.9f
1608 ddi_modsym.9f         := LINKSRC = ddi_modopen.9f

1610 ddi_peek16.9f         := LINKSRC = ddi_peek.9f
1611 ddi_peek32.9f        := LINKSRC = ddi_peek.9f
1612 ddi_peek64.9f        := LINKSRC = ddi_peek.9f
1613 ddi_peek8.9f          := LINKSRC = ddi_peek.9f
1614 ddi_peekc.9f         := LINKSRC = ddi_peek.9f
1615 ddi_peekd.9f         := LINKSRC = ddi_peek.9f
1616 ddi_peekl.9f         := LINKSRC = ddi_peek.9f
1617 ddi_peeks.9f         := LINKSRC = ddi_peek.9f

1619 ddi_poke16.9f         := LINKSRC = ddi_poke.9f
1620 ddi_poke32.9f        := LINKSRC = ddi_poke.9f
1621 ddi_poke64.9f        := LINKSRC = ddi_poke.9f
1622 ddi_poke8.9f         := LINKSRC = ddi_poke.9f
1623 ddi_pokec.9f         := LINKSRC = ddi_poke.9f
1624 ddi_poked.9f         := LINKSRC = ddi_poke.9f
1625 ddi_pokel.9f         := LINKSRC = ddi_poke.9f
1626 ddi_pokes.9f         := LINKSRC = ddi_poke.9f

1628 ddi_prop_modify.9f    := LINKSRC = ddi_prop_create.9f
1629 ddi_prop_remove.9f    := LINKSRC = ddi_prop_create.9f
1630 ddi_prop_remove_all.9f := LINKSRC = ddi_prop_create.9f
1631 ddi_prop_undefine.9f  := LINKSRC = ddi_prop_create.9f

1633 ddi_prop_get_int64.9f := LINKSRC = ddi_prop_get_int.9f

1635 ddi_prop_free.9f      := LINKSRC = ddi_prop_lookup.9f
1636 ddi_prop_lookup_byte_array.9f := LINKSRC = ddi_prop_lookup.9f
1637 ddi_prop_lookup_int64_array.9f := LINKSRC = ddi_prop_lookup.9f
1638 ddi_prop_lookup_int_array.9f := LINKSRC = ddi_prop_lookup.9f
1639 ddi_prop_lookup_string.9f := LINKSRC = ddi_prop_lookup.9f
1640 ddi_prop_lookup_string_array.9f := LINKSRC = ddi_prop_lookup.9f

1642 ddi_getlongprop.9f    := LINKSRC = ddi_prop_op.9f

```

```

1643 ddi_getlongprop_buf.9f := LINKSRC = ddi_prop_op.9f
1644 ddi_getprop.9f         := LINKSRC = ddi_prop_op.9f
1645 ddi_getpropflen.9f     := LINKSRC = ddi_prop_op.9f

1647 ddi_prop_update_byte_array.9f := LINKSRC = ddi_prop_update.9f
1648 ddi_prop_update_int.9f        := LINKSRC = ddi_prop_update.9f
1649 ddi_prop_update_int64.9f      := LINKSRC = ddi_prop_update.9f
1650 ddi_prop_update_int64_array.9f := LINKSRC = ddi_prop_update.9f
1651 ddi_prop_update_int_array.9f  := LINKSRC = ddi_prop_update.9f
1652 ddi_prop_update_string.9f     := LINKSRC = ddi_prop_update.9f
1653 ddi_prop_update_string_array.9f := LINKSRC = ddi_prop_update.9f

1655 ddi_put16.9f           := LINKSRC = ddi_put8.9f
1656 ddi_put32.9f           := LINKSRC = ddi_put8.9f
1657 ddi_put64.9f           := LINKSRC = ddi_put8.9f
1658 ddi_putb.9f            := LINKSRC = ddi_put8.9f
1659 ddi_putl.9f            := LINKSRC = ddi_put8.9f
1660 ddi_putll.9f           := LINKSRC = ddi_put8.9f
1661 ddi_putw.9f            := LINKSRC = ddi_put8.9f

1663 ddi_rep_get16.9f       := LINKSRC = ddi_rep_get8.9f
1664 ddi_rep_get32.9f      := LINKSRC = ddi_rep_get8.9f
1665 ddi_rep_get64.9f      := LINKSRC = ddi_rep_get8.9f
1666 ddi_rep_getb.9f       := LINKSRC = ddi_rep_get8.9f
1667 ddi_rep_getl.9f       := LINKSRC = ddi_rep_get8.9f
1668 ddi_rep_getll.9f      := LINKSRC = ddi_rep_get8.9f
1669 ddi_rep_getw.9f       := LINKSRC = ddi_rep_get8.9f

1671 ddi_rep_put16.9f      := LINKSRC = ddi_rep_put8.9f
1672 ddi_rep_put32.9f     := LINKSRC = ddi_rep_put8.9f
1673 ddi_rep_put64.9f     := LINKSRC = ddi_rep_put8.9f
1674 ddi_rep_putb.9f      := LINKSRC = ddi_rep_put8.9f
1675 ddi_rep_putl.9f      := LINKSRC = ddi_rep_put8.9f
1676 ddi_rep_putll.9f     := LINKSRC = ddi_rep_put8.9f
1677 ddi_rep_putw.9f      := LINKSRC = ddi_rep_put8.9f

1679 ddi_segmap_setup.9f   := LINKSRC = ddi_segmap.9f

1681 ddi_get_soft_state.9f := LINKSRC = ddi_soft_state.9f
1682 ddi_soft_state_fini.9f := LINKSRC = ddi_soft_state.9f
1683 ddi_soft_state_free.9f := LINKSRC = ddi_soft_state.9f
1684 ddi_soft_state_init.9f := LINKSRC = ddi_soft_state.9f
1685 ddi_soft_state_zalloc.9f := LINKSRC = ddi_soft_state.9f

1687 ddi_strtoul.9f        := LINKSRC = ddi_strtoll.9f

1689 ddi_umem_free.9f     := LINKSRC = ddi_umem_alloc.9f

1691 ddi_umem_unlock.9f   := LINKSRC = ddi_umem_lock.9f

1693 devmap_umem_setup.9f := LINKSRC = devmap_devmem_setup.9f

1695 ddi_devmap_segmap.9f := LINKSRC = devmap_setup.9f

1697 devmap_load.9f       := LINKSRC = devmap_unload.9f

1699 dlerrorack.9f        := LINKSRC = dlbindack.9f
1700 dlkack.9f            := LINKSRC = dlbindack.9f
1701 dlphysaddrack.9f     := LINKSRC = dlbindack.9f
1702 dluderrorind.9f      := LINKSRC = dlbindack.9f

1704 desballoc.9f         := LINKSRC = esballoc.9f

1706 unfreezestr.9f       := LINKSRC = freezestr.9f

1708 firmware_close.9f    := LINKSRC = firmload.9f

```

```

1709 firmware_free.9f      := LINKSRC = firmload.9f
1710 firmware_get_size.9f  := LINKSRC = firmload.9f
1711 firmware_malloc.9f    := LINKSRC = firmload.9f
1712 firmware_open.9f     := LINKSRC = firmload.9f
1713 firmware_read.9f     := LINKSRC = firmload.9f

1715 free_pktiopb.9f      := LINKSRC = get_pktiopb.9f

1717 gld_intr.9f           := LINKSRC = gld.9f
1718 gld_mac_alloc.9f     := LINKSRC = gld.9f
1719 gld_mac_free.9f     := LINKSRC = gld.9f
1720 gld_recv.9f          := LINKSRC = gld.9f
1721 gld_register.9f     := LINKSRC = gld.9f
1722 gld_sched.9f        := LINKSRC = gld.9f
1723 gld_unregister.9f   := LINKSRC = gld.9f

1725 id_space_create.9f   := LINKSRC = id_space.9f
1726 id_space_destroy.9f  := LINKSRC = id_space.9f
1727 id_space_extend.9f   := LINKSRC = id_space.9f
1728 id_alloc.9f          := LINKSRC = id_space.9f
1729 id_alloc_nosleep.9f  := LINKSRC = id_space.9f
1730 id_allocff.9f        := LINKSRC = id_space.9f
1731 id_allocff_nosleep.9f := LINKSRC = id_space.9f
1732 id_alloc_specific_nosleep.9f := LINKSRC = id_space.9f
1733 id_free.9f           := LINKSRC = id_space.9f

1735 id32_free.9f         := LINKSRC = id32_alloc.9f
1736 id32_lookup.9f     := LINKSRC = id32_alloc.9f

1738 inl.9f               := LINKSRC = inb.9f
1739 inw.9f               := LINKSRC = inb.9f
1740 repinsb.9f           := LINKSRC = inb.9f
1741 repinsd.9f          := LINKSRC = inb.9f
1742 repinsw.9f          := LINKSRC = inb.9f

1744 kmem_free.9f        := LINKSRC = kmem_alloc.9f
1745 kmem_zalloc.9f     := LINKSRC = kmem_alloc.9f

1747 kmem_cache_alloc.9f := LINKSRC = kmem_cache_create.9f
1748 kmem_cache_destroy.9f := LINKSRC = kmem_cache_create.9f
1749 kmem_cache_free.9f  := LINKSRC = kmem_cache_create.9f
1750 kmem_cache_set_move.9f := LINKSRC = kmem_cache_create.9f

1752 kstat_named_setstr.9f := LINKSRC = kstat_named_init.9f

1754 kstat_runq_back_to_waitq.9f := LINKSRC = kstat_queue.9f
1755 kstat_runq_enter.9f      := LINKSRC = kstat_queue.9f
1756 kstat_runq_exit.9f      := LINKSRC = kstat_queue.9f
1757 kstat_waitq_enter.9f    := LINKSRC = kstat_queue.9f
1758 kstat_waitq_exit.9f    := LINKSRC = kstat_queue.9f
1759 kstat_waitq_to_runq.9f  := LINKSRC = kstat_queue.9f

1761 ldi_awrite.9f        := LINKSRC = ldi_aread.9f

1763 ldi_get_devid.9f     := LINKSRC = ldi_get_dev.9f
1764 ldi_get_minor_name.9f := LINKSRC = ldi_get_dev.9f
1765 ldi_get_otyp.9f     := LINKSRC = ldi_get_dev.9f

1767 ldi_ident_from_dip.9f := LINKSRC = ldi_ident_from_dev.9f
1768 ldi_ident_from_stream.9f := LINKSRC = ldi_ident_from_dev.9f
1769 ldi_ident_release.9f := LINKSRC = ldi_ident_from_dev.9f

1771 ldi_close.9f        := LINKSRC = ldi_open_by_dev.9f
1772 ldi_open_by_devid.9f := LINKSRC = ldi_open_by_dev.9f
1773 ldi_open_by_name.9f := LINKSRC = ldi_open_by_dev.9f

```

```

1775 ldi_prop_get_int64.9f := LINKSRC = ldi_prop_get_int.9f

1777 ldi_prop_lookup_byte_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f
1778 ldi_prop_lookup_int64_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f
1779 ldi_prop_lookup_string.9f      := LINKSRC = ldi_prop_lookup_int_array.9f
1780 ldi_prop_lookup_string_array.9f := LINKSRC = ldi_prop_lookup_int_array.9f

1782 ldi_getmsg.9f           := LINKSRC = ldi_putmsg.9f

1784 ldi_write.9f          := LINKSRC = ldi_read.9f

1786 list_destroy.9f       := LINKSRC = list_create.9f
1787 list_head.9f         := LINKSRC = list_create.9f
1788 list_insert_after.9f  := LINKSRC = list_create.9f
1789 list_insert_before.9f := LINKSRC = list_create.9f
1790 list_insert_head.9f   := LINKSRC = list_create.9f
1791 list_insert_tail.9f   := LINKSRC = list_create.9f
1792 list_is_empty.9f     := LINKSRC = list_create.9f
1793 list_link_active.9f  := LINKSRC = list_create.9f
1794 list_link_init.9f    := LINKSRC = list_create.9f
1795 list_link_replace.9f := LINKSRC = list_create.9f
1796 list_move_tail.9f   := LINKSRC = list_create.9f
1797 list_next.9f        := LINKSRC = list_create.9f
1798 list_prev.9f        := LINKSRC = list_create.9f
1799 list_remove.9f      := LINKSRC = list_create.9f
1800 list_remove_head.9f := LINKSRC = list_create.9f
1801 list_remove_tail.9f := LINKSRC = list_create.9f
1802 list_tail.9f       := LINKSRC = list_create.9f

1804 mac_free.9f          := LINKSRC = mac_alloc.9f
1805 mac_hcksum_set.9f    := LINKSRC = mac_hcksum_get.9f
1806 mac_fini_ops.9f     := LINKSRC = mac_init_ops.9f
1807 mac_prop_info_set_default_link_flowctrl.9f := LINKSRC = mac_prop_info.9f
1808 mac_prop_info_set_default_str.9f          := LINKSRC = mac_prop_info.9f
1809 mac_prop_info_set_default_uint8.9f       := LINKSRC = mac_prop_info.9f
1810 mac_prop_info_set_default_uint32.9f      := LINKSRC = mac_prop_info.9f
1811 mac_prop_info_set_default_uint64.9f     := LINKSRC = mac_prop_info.9f
1812 mac_prop_info_set_perm.9f              := LINKSRC = mac_prop_info.9f
1813 mac_prop_info_set_range_uint32.9f      := LINKSRC = mac_prop_info.9f
1814 mac_unregister.9f                       := LINKSRC = mac_register.9f

1816 makecom_g0.9f        := LINKSRC = makecom.9f
1817 makecom_g0_s.9f     := LINKSRC = makecom.9f
1818 makecom_g1.9f       := LINKSRC = makecom.9f
1819 makecom_g5.9f       := LINKSRC = makecom.9f

1821 membar_consumer.9f  := LINKSRC = membar_ops.9f
1822 membar_enter.9f    := LINKSRC = membar_ops.9f
1823 membar_exit.9f     := LINKSRC = membar_ops.9f
1824 membar_producer.9f := LINKSRC = membar_ops.9f

1826 memcmp.9f          := LINKSRC = memchr.9f
1827 memcpy.9f         := LINKSRC = memchr.9f
1828 memmove.9f        := LINKSRC = memchr.9f
1829 memset.9f         := LINKSRC = memchr.9f

1831 mod_info.9f        := LINKSRC = mod_install.9f
1832 mod_modname.9f   := LINKSRC = mod_install.9f
1833 mod_remove.9f    := LINKSRC = mod_install.9f

1835 mutex_destroy.9f   := LINKSRC = mutex.9f
1836 mutex_enter.9f    := LINKSRC = mutex.9f
1837 mutex_exit.9f     := LINKSRC = mutex.9f
1838 mutex_init.9f     := LINKSRC = mutex.9f
1839 mutex_owed.9f     := LINKSRC = mutex.9f
1840 mutex_tryenter.9f := LINKSRC = mutex.9f

```

```

1842 net_event_notify_unregister.9f      == LINKSRC = net_event_notify_register.9
1844 net_instance_notify_unregister.9f    == LINKSRC = net_instance_notify_registe
1846 net_instance_protocol_unregister.9f  == LINKSRC = net_protocol_notify_registe

1848 nvlist_add_boolean_array.9f          == LINKSRC = nvlist_add_boolean.9f
1849 nvlist_add_boolean_value.9f         == LINKSRC = nvlist_add_boolean.9f
1850 nvlist_add_byte.9f                   == LINKSRC = nvlist_add_boolean.9f
1851 nvlist_add_byte_array.9f            == LINKSRC = nvlist_add_boolean.9f
1852 nvlist_add_int16.9f                  == LINKSRC = nvlist_add_boolean.9f
1853 nvlist_add_int16_array.9f            == LINKSRC = nvlist_add_boolean.9f
1854 nvlist_add_int32.9f                  == LINKSRC = nvlist_add_boolean.9f
1855 nvlist_add_int32_array.9f            == LINKSRC = nvlist_add_boolean.9f
1856 nvlist_add_int64.9f                  == LINKSRC = nvlist_add_boolean.9f
1857 nvlist_add_int64_array.9f            == LINKSRC = nvlist_add_boolean.9f
1858 nvlist_add_int8.9f                   == LINKSRC = nvlist_add_boolean.9f
1859 nvlist_add_int8_array.9f             == LINKSRC = nvlist_add_boolean.9f
1860 nvlist_add_nvlist.9f                  == LINKSRC = nvlist_add_boolean.9f
1861 nvlist_add_nvlist_array.9f            == LINKSRC = nvlist_add_boolean.9f
1862 nvlist_add_nvpair.9f                  == LINKSRC = nvlist_add_boolean.9f
1863 nvlist_add_string.9f                  == LINKSRC = nvlist_add_boolean.9f
1864 nvlist_add_string_array.9f            == LINKSRC = nvlist_add_boolean.9f
1865 nvlist_add_uint16.9f                  == LINKSRC = nvlist_add_boolean.9f
1866 nvlist_add_uint16_array.9f            == LINKSRC = nvlist_add_boolean.9f
1867 nvlist_add_uint32.9f                  == LINKSRC = nvlist_add_boolean.9f
1868 nvlist_add_uint32_array.9f            == LINKSRC = nvlist_add_boolean.9f
1869 nvlist_add_uint64.9f                  == LINKSRC = nvlist_add_boolean.9f
1870 nvlist_add_uint64_array.9f            == LINKSRC = nvlist_add_boolean.9f
1871 nvlist_add_uint8.9f                   == LINKSRC = nvlist_add_boolean.9f
1872 nvlist_add_uint8_array.9f             == LINKSRC = nvlist_add_boolean.9f
1873 nvlist_t.9f                           == LINKSRC = nvlist_add_boolean.9f

1875 nv_alloc_fini.9f                      == LINKSRC = nvlist_alloc.9f
1876 nv_alloc_init.9f                     == LINKSRC = nvlist_alloc.9f
1877 nvlist_dup.9f                          == LINKSRC = nvlist_alloc.9f
1878 nvlist_free.9f                         == LINKSRC = nvlist_alloc.9f
1879 nvlist_merge.9f                        == LINKSRC = nvlist_alloc.9f
1880 nvlist_pack.9f                         == LINKSRC = nvlist_alloc.9f
1881 nvlist_size.9f                         == LINKSRC = nvlist_alloc.9f
1882 nvlist_unpack.9f                       == LINKSRC = nvlist_alloc.9f
1883 nvlist_xalloc.9f                       == LINKSRC = nvlist_alloc.9f
1884 nvlist_xdup.9f                         == LINKSRC = nvlist_alloc.9f
1885 nvlist_xpack.9f                       == LINKSRC = nvlist_alloc.9f
1886 nvlist_xunpack.9f                     == LINKSRC = nvlist_alloc.9f

1888 nvlist_lookup_boolean_array.9f        == LINKSRC = nvlist_lookup_boolean.9f
1889 nvlist_lookup_boolean_value.9f        == LINKSRC = nvlist_lookup_boolean.9f
1890 nvlist_lookup_byte.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1891 nvlist_lookup_byte_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1892 nvlist_lookup_int16.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1893 nvlist_lookup_int16_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1894 nvlist_lookup_int32.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1895 nvlist_lookup_int32_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1896 nvlist_lookup_int64.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1897 nvlist_lookup_int64_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1898 nvlist_lookup_int8.9f                   == LINKSRC = nvlist_lookup_boolean.9f
1899 nvlist_lookup_int8_array.9f             == LINKSRC = nvlist_lookup_boolean.9f
1900 nvlist_lookup_nvlist.9f                == LINKSRC = nvlist_lookup_boolean.9f
1901 nvlist_lookup_nvlist_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1902 nvlist_lookup_pairs.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1903 nvlist_lookup_string.9f                 == LINKSRC = nvlist_lookup_boolean.9f
1904 nvlist_lookup_string_array.9f            == LINKSRC = nvlist_lookup_boolean.9f
1905 nvlist_lookup_uint16.9f                 == LINKSRC = nvlist_lookup_boolean.9f
1906 nvlist_lookup_uint16_array.9f            == LINKSRC = nvlist_lookup_boolean.9f

```

```

1907 nvlist_lookup_uint32.9f               == LINKSRC = nvlist_lookup_boolean.9f
1908 nvlist_lookup_uint32_array.9f           == LINKSRC = nvlist_lookup_boolean.9f
1909 nvlist_lookup_uint64.9f                 == LINKSRC = nvlist_lookup_boolean.9f
1910 nvlist_lookup_uint64_array.9f           == LINKSRC = nvlist_lookup_boolean.9f
1911 nvlist_lookup_uint8.9f                  == LINKSRC = nvlist_lookup_boolean.9f
1912 nvlist_lookup_uint8_array.9f            == LINKSRC = nvlist_lookup_boolean.9f

1914 nvlist_exists.9f                       == LINKSRC = nvlist_lookup_nvpair.9f

1916 nvpair_name.9f                         == LINKSRC = nvlist_next_nvpair.9f
1917 nvpair_type.9f                         == LINKSRC = nvlist_next_nvpair.9f

1919 nvlist_remove_all.9f                   == LINKSRC = nvlist_remove.9f

1921 nvpair_value_boolean_array.9f           == LINKSRC = nvpair_value_byte.9f
1922 nvpair_value_byte_array.9f             == LINKSRC = nvpair_value_byte.9f
1923 nvpair_value_int16.9f                   == LINKSRC = nvpair_value_byte.9f
1924 nvpair_value_int16_array.9f             == LINKSRC = nvpair_value_byte.9f
1925 nvpair_value_int32.9f                   == LINKSRC = nvpair_value_byte.9f
1926 nvpair_value_int32_array.9f            == LINKSRC = nvpair_value_byte.9f
1927 nvpair_value_int64.9f                   == LINKSRC = nvpair_value_byte.9f
1928 nvpair_value_int64_array.9f            == LINKSRC = nvpair_value_byte.9f
1929 nvpair_value_int8.9f                    == LINKSRC = nvpair_value_byte.9f
1930 nvpair_value_int8_array.9f              == LINKSRC = nvpair_value_byte.9f
1931 nvpair_value_nvlist.9f                  == LINKSRC = nvpair_value_byte.9f
1932 nvpair_value_nvlist_array.9f            == LINKSRC = nvpair_value_byte.9f
1933 nvpair_value_string.9f                  == LINKSRC = nvpair_value_byte.9f
1934 nvpair_value_string_array.9f            == LINKSRC = nvpair_value_byte.9f
1935 nvpair_value_uint16.9f                  == LINKSRC = nvpair_value_byte.9f
1936 nvpair_value_uint16_array.9f            == LINKSRC = nvpair_value_byte.9f
1937 nvpair_value_uint32.9f                  == LINKSRC = nvpair_value_byte.9f
1938 nvpair_value_uint32_array.9f            == LINKSRC = nvpair_value_byte.9f
1939 nvpair_value_uint64.9f                  == LINKSRC = nvpair_value_byte.9f
1940 nvpair_value_uint64_array.9f            == LINKSRC = nvpair_value_byte.9f
1941 nvpair_value_uint8.9f                   == LINKSRC = nvpair_value_byte.9f
1942 nvpair_value_uint8_array.9f             == LINKSRC = nvpair_value_byte.9f

1944 outl.9f                                 == LINKSRC = outb.9f
1945 outw.9f                                 == LINKSRC = outb.9f
1946 repoutsb.9f                             == LINKSRC = outb.9f
1947 repoutsd.9f                             == LINKSRC = outb.9f
1948 repoutsw.9f                             == LINKSRC = outb.9f

1950 pci_config_get16.9f                     == LINKSRC = pci_config_get8.9f
1951 pci_config_get32.9f                     == LINKSRC = pci_config_get8.9f
1952 pci_config_get64.9f                     == LINKSRC = pci_config_get8.9f
1953 pci_config_getb.9f                      == LINKSRC = pci_config_get8.9f
1954 pci_config_getl.9f                      == LINKSRC = pci_config_get8.9f
1955 pci_config_getll.9f                     == LINKSRC = pci_config_get8.9f
1956 pci_config_gettb.9f                     == LINKSRC = pci_config_get8.9f
1957 pci_config_put16.9f                     == LINKSRC = pci_config_get8.9f
1958 pci_config_put32.9f                     == LINKSRC = pci_config_get8.9f
1959 pci_config_put64.9f                     == LINKSRC = pci_config_get8.9f
1960 pci_config_put8.9f                       == LINKSRC = pci_config_get8.9f
1961 pci_config_putb.9f                       == LINKSRC = pci_config_get8.9f
1962 pci_config_putl.9f                       == LINKSRC = pci_config_get8.9f
1963 pci_config_putll.9f                      == LINKSRC = pci_config_get8.9f
1964 pci_config_putw.9f                       == LINKSRC = pci_config_get8.9f

1966 pci_config_teardown.9f                 == LINKSRC = pci_config_setup.9f

1968 pci_ereport_post.9f                     == LINKSRC = pci_ereport_setup.9f
1969 pci_ereport_teardown.9f                 == LINKSRC = pci_ereport_setup.9f

1971 pci_restore_config_regs.9f              == LINKSRC = pci_save_config_regs.9f

```

```

1973 minphys.9f           := LINKSRC = physio.9f
1975 pm_idle_component.9f := LINKSRC = pm_busy_component.9f
1977 pm_lower_power.9f    := LINKSRC = pm_raise_power.9f
1979 priv_policy_choice.9f := LINKSRC = priv_policy.9f
1980 priv_policy_only.9f   := LINKSRC = priv_policy.9f

1982 proc_ref.9f          := LINKSRC = proc_signal.9f
1983 proc_unref.9f        := LINKSRC = proc_signal.9f

1985 qprocsoff.9f         := LINKSRC = qprocson.9f

1987 qwait_sig.9f         := LINKSRC = qwait.9f

1989 rmallocmap_wait.9f    := LINKSRC = rmallocmap.9f
1990 rmfreemap.9f          := LINKSRC = rmallocmap.9f

1992 rw_destroy.9f         := LINKSRC = rwlock.9f
1993 rw_downgrade.9f      := LINKSRC = rwlock.9f
1994 rw_enter.9f           := LINKSRC = rwlock.9f
1995 rw_exit.9f            := LINKSRC = rwlock.9f
1996 rw_init.9f            := LINKSRC = rwlock.9f
1997 rw_read_locked.9f    := LINKSRC = rwlock.9f
1998 rw_tryenter.9f       := LINKSRC = rwlock.9f
1999 rw_tryupgrade.9f     := LINKSRC = rwlock.9f

2001 scsi_dname.9f         := LINKSRC = scsi_cname.9f
2002 scsi_mname.9f         := LINKSRC = scsi_cname.9f
2003 scsi_rname.9f        := LINKSRC = scsi_cname.9f
2004 scsi_sname.9f        := LINKSRC = scsi_cname.9f

2006 scsi_dmafree.9f      := LINKSRC = scsi_dmaget.9f

2008 scsi_sense_cmdspecific_uint64.9f := LINKSRC = scsi_ext_sense_fields.9f
2009 scsi_sense_info_uint64.9f := LINKSRC = scsi_ext_sense_fields.9f

2011 scsi_hba_detach.9f   := LINKSRC = scsi_hba_attach_setup.9f

2013 scsi_hba_fini.9f     := LINKSRC = scsi_hba_init.9f

2015 scsi_hba_pkt_free.9f := LINKSRC = scsi_hba_pkt_alloc.9f

2017 scsi_hba_tran_free.9f := LINKSRC = scsi_hba_tran_alloc.9f

2019 scsi_ifsetcap.9f     := LINKSRC = scsi_ifgetcap.9f

2021 scsi_pktfree.9f       := LINKSRC = scsi_pktalloc.9f
2022 scsi_realloc.9f      := LINKSRC = scsi_pktalloc.9f
2023 scsi_resfree.9f      := LINKSRC = scsi_pktalloc.9f

2025 scsi_sense_asc.9f     := LINKSRC = scsi_sense_key.9f
2026 scsi_sense_ascq.9f   := LINKSRC = scsi_sense_key.9f

2028 scsi_unslave.9f      := LINKSRC = scsi_unprobe.9f

2030 sema_destroy.9f       := LINKSRC = semaphore.9f
2031 sema_init.9f          := LINKSRC = semaphore.9f
2032 sema_p.9f             := LINKSRC = semaphore.9f
2033 sema_p_sig.9f         := LINKSRC = semaphore.9f
2034 sema_try.9f           := LINKSRC = semaphore.9f
2035 sema_v.9f             := LINKSRC = semaphore.9f

2037 snprintf.9f           := LINKSRC = sprintf.9f
2038 vsprintf.9f           := LINKSRC = sprintf.9f

```

```

2039 vsnprintf.9f         := LINKSRC = sprintf.9f

2041 numtos.9f            := LINKSRC = stoi.9f

2043 ddi_strdup.9f         := LINKSRC = string.9f
2044 strcasecmp.9f        := LINKSRC = string.9f
2045 strcat.9f            := LINKSRC = string.9f
2046 strchr.9f            := LINKSRC = string.9f
2047 strcmp.9f            := LINKSRC = string.9f
2048 strcpy.9f             := LINKSRC = string.9f
2049 strdup.9f            := LINKSRC = string.9f
2050 strfree.9f            := LINKSRC = string.9f
2051 strlcat.9f            := LINKSRC = string.9f
2052 strlcpy.9f            := LINKSRC = string.9f
2053 strlen.9f             := LINKSRC = string.9f
2054 strncasecmp.9f       := LINKSRC = string.9f
2055 strncat.9f           := LINKSRC = string.9f
2056 strncmp.9f           := LINKSRC = string.9f
2057 strncpy.9f           := LINKSRC = string.9f
2058 strnlen.9f           := LINKSRC = string.9f
2059 strchr.9f             := LINKSRC = string.9f
2060 strspn.9f             := LINKSRC = string.9f

2062 ddi_taskq_create.9f   := LINKSRC = taskq.9f
2063 ddi_taskq_destroy.9f := LINKSRC = taskq.9f
2064 ddi_taskq_dispatch.9f := LINKSRC = taskq.9f
2065 ddi_taskq_resume.9f  := LINKSRC = taskq.9f
2066 ddi_taskq_suspend.9f := LINKSRC = taskq.9f
2067 ddi_taskq_wait.9f    := LINKSRC = taskq.9f
2068 taskq_suspended.9f   := LINKSRC = taskq.9f

2070 uconv_u16tou8.9f      := LINKSRC = uconv_u16tou32.9f
2071 uconv_u32tou16.9f    := LINKSRC = uconv_u16tou32.9f
2072 uconv_u32tou8.9f     := LINKSRC = uconv_u16tou32.9f
2073 uconv_u8tou16.9f     := LINKSRC = uconv_u16tou32.9f
2074 uconv_u8tou32.9f     := LINKSRC = uconv_u16tou32.9f

2076 usb_alloc_bulk_req.9f := LINKSRC = usb_alloc_request.9f
2077 usb_alloc_ctrl_req.9f := LINKSRC = usb_alloc_request.9f
2078 usb_alloc_intr_req.9f := LINKSRC = usb_alloc_request.9f
2079 usb_alloc_isoc_req.9f := LINKSRC = usb_alloc_request.9f
2080 usb_free_bulk_req.9f  := LINKSRC = usb_alloc_request.9f
2081 usb_free_ctrl_req.9f := LINKSRC = usb_alloc_request.9f
2082 usb_free_intr_req.9f  := LINKSRC = usb_alloc_request.9f
2083 usb_free_isoc_req.9f := LINKSRC = usb_alloc_request.9f
2084 usb_client_detach.9f := LINKSRC = usb_client_attach.9f

2086 usb_get_if_number.9f  := LINKSRC = usb_get_alt_if.9f
2087 usb_owns_device.9f   := LINKSRC = usb_get_alt_if.9f
2088 usb_set_alt_if.9f    := LINKSRC = usb_get_alt_if.9f

2090 usb_set_cfg.9f        := LINKSRC = usb_get_cfg.9f

2092 usb_free_descr_tree.9f := LINKSRC = usb_get_dev_data.9f
2093 usb_free_dev_data.9f  := LINKSRC = usb_get_dev_data.9f
2094 usb_print_descr_tree.9f := LINKSRC = usb_get_dev_data.9f

2096 usb_pipe_ctrl_xfer_wait.9f := LINKSRC = usb_pipe_ctrl_xfer.9f
2097 usb_pipe_stop_intr_polling.9f := LINKSRC = usb_pipe_intr_xfer.9f
2098 usb_pipe_stop_isoc_polling.9f := LINKSRC = usb_pipe_isoc_xfer.9f

2100 usb_pipe_get_private.9f := LINKSRC = usb_pipe_set_private.9f

2102 usb_unregister_hotplug_cbs.9f := LINKSRC = usb_register_hotplug_cbs.9f

2104 va_copy.9f            := LINKSRC = va_arg.9f

```

```
2105 va_end.9f      := LINKSRC = va_arg.9f
2106 va_start.9f   := LINKSRC = va_arg.9f

2108 vmem_xalloc.9f := LINKSRC = vmem_alloc.9f
2109 vmem_free.9f  := LINKSRC = vmem_alloc.9f
2110 vmem_xfree.9f := LINKSRC = vmem_alloc.9f

2112 vmem_xcreate.9f := LINKSRC = vmem_create.9f
2113 vmem_destroy.9f := LINKSRC = vmem_create.9f

2115 vmem_size.9f   := LINKSRC = vmem_walk.9f

2117 #endif /* ! codereview */
2118 .KEEP_STATE:

2120 include      $(SRC)/man/Makefile.man

2122 install:    $(ROOTMANFILES) $(ROOTMANLINKS)
```

```
*****
```

```
1556 Thu Feb 2 13:24:15 2017
```

```
new/usr/src/man/man9f/vmem_add.9f
```

```
7831 want vmem manual pages
```

```
7832 big theory statements need a place in the manual
```

```
*****
```

```
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 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_ADD 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_add
19 .Nd add spans to a vmem arena
20 .Sh SYNOPSIS
21 .In sys/vmem.h
22 .Ft void *
23 .Fo vmem_add
24 .Fa "vmem_t *vmp"
25 .Fa "void *vaddr"
26 .Fa "size_t size"
27 .Fa "int vmflag"
28 .Fc
29 .Sh INTERFACE LEVEL
30 illumos DDI specific
31 .Sh PARAMETERS
32 .Bl -tag -width Ds
33 .It Fa vmp
34 The vmem arena to which the span should be added.
35 .It Fa vaddr
36 The base address of the span to add.
37 .It Fa size
38 The size of the span to add
39 .It Fa vmflag
40 Flags affecting the allocation of the span to add.
41 .El
42 .Sh DESCRIPTION
43 The
44 .Fn vmem_add
45 function adds
46 .Fa size
47 bytes starting at
48 .Fa vaddr
49 to a vmem arena from which future calls to
50 .Fn vmem_alloc
51 may allocate.
52 .Pp
53 .Dv VM_SLEEP
54 or
55 .Dv VM_NOSLEEP
56 must be specified, and indicate whether the addition may block.
57 .Sh CONTEXT
58 This function can be called from either user or kernel context.
59 If the
60 .Dv VM_NOSLEEP
```

```
61 flag is specified, it may also be called from interrupt context.
62 .Sh RETURN VALUES
63 Upon success
64 .Fn vmem_add
65 returns
66 .Fa vaddr .
67 On failure,
68 .Dv NULL
69 is returned.
70 .Sh SEE ALSO
71 .Xr vmem 9 ,
72 .Xr vmem_alloc 9F ,
73 .Xr vmem_create 9F
74 #endif /* ! codereview */
```

4062 Thu Feb 2 13:24:16 2017

new/usr/src/man/man9f/vmem_alloc.9f

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```

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 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_ALLOC 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_alloc ,
19 .Nm vmem_xalloc ,
20 .Nm vmem_free ,
21 .Nm vmem_xfree
22 .Nd allocate and free segments from a vmem arena
23 .Sh SYNOPSIS
24 .In sys/vmem.h
25 .Ft void *
26 .Fo vmem_alloc
27 .Fa "vmem_t *vmp"
28 .Fa "size_t size"
29 .Fa "int vmflag"
30 .Fc
31 .Ft void *
32 .Fo vmem_xalloc
33 .Fa "vmem_t *vmp"
34 .Fa "size_t size"
35 .Fa "size_t align_arg"
36 .Fa "size_t phase"
37 .Fa "size_t nocross"
38 .Fa "void *minaddr"
39 .Fa "void *maxaddr"
40 .Fa "int vmflag"
41 .Fc
42 .Ft void
43 .Fo vmem_free
44 .Fa "vmem_t *vmp"
45 .Fa "void *vaddr"
46 .Fa "size_t size"
47 .Fc
48 .Ft void
49 .Fo vmem_xfree
50 .Fa "vmem_t *vmp"
51 .Fa "void *vaddr"
52 .Fa "size_t size"
53 .Fc
54 .Sh INTERFACE LEVEL
55 illumos DDI specific
56 .Sh PARAMETERS
57 .Bl -tag -width Ds
58 .It Fa vmp
59 The vmem arena from which to allocate or free.
60 .It Fa size

```

```

61 The size of the segment to allocate or free.
62 .It Fa vmflag
63 A bitmask of flags controlling the behaviour of the allocation.
64 There are two meaningful groups of flags.
65 .Dv VM_SLEEP
66 or
67 .Dv VM_NOSLEEP
68 must be specified, and indicate whether the allocation may block. A
69 .Dv VM_SLEEP
70 allocation can never fail but may block indefinitely.
71 .Pp
72 The allocation policy may be specified by one of the following flags:
73 .Bl -tag -width Ds
74 .It Dv VM_BESTFIT
75 Take the segment from the smallest free segment that could satisfy this allocati
76 .It Dv VM_FIRSTFIT
77 Take the segment from the first free segment found that could satisfy this
78 allocation.
79 .It Dv VM_NEXTFIT
80 Take the segment from the segment after the one previously allocated. This
81 provides sequential behaviour useful when allocating identifiers from a
82 .Dv VMC_IDENTIFIER
83 arena.
84 .It Dv VM_ENDALLOC
85 May be specified in combination with
86 .Dv VM_BESTFIT ,
87 .Dv VM_FIRSTFIT
88 or the default policy to indicate that the higher addresses should be
89 preferred.
90 .El
91 .Pp
92 The default (un\-named) allocation policy is
93 .Dq instant fit
94 an approximation of
95 .Dv VM_BESTFIT
96 in guaranteed constant time.
97 .It Fa align_arg
98 The minimum alignment of the allocation. If
99 .Ql 0
100 the allocated segment will be aligned as the arena's quantum.
101 .It Fa phase
102 The allocated segment must be
103 .Fa phase
104 bytes from the alignment boundary.
105 .It Fa nocross
106 The allocated segment may not straddle a
107 .Fa nocross
108 alignment boundary.
109 .It Fa minaddr
110 The minimum address at which the segment may be allocated.
111 .It Fa maxaddr
112 The maximum address which may be included in the segment.
113 .It Fa vaddr
114 The address of the segment which
115 .Fn vmem_free
116 or
117 .Fn vmem_xfree
118 should free.
119 .El
120 .Sh DESCRIPTION
121 The
122 .Fn vmem_alloc
123 and
124 .Fn vmem_xalloc
125 functions allocate a segment of
126 .Fa size

```



```
127 length from the vmem arena
128 .Fa vmp .
129 .Pp
130 The
131 .Fa vmflag
132 argument controls the behaviour of the allocation. As described in
133 .Sx PARAMETERS
134 .Pp
135 For allocations with complex requirements, such as those used for DMA
136 .Fn vmem_xalloc
137 takes additional arguments allowing those requirements to be expressed.
138 .Pp
139 Segments allocated with
140 .Fn vmem_xalloc
141 must always be freed with
142 .Fn vmem_xfree ,
143 since these allocations are uncached.
144 .Sh CONTEXT
145 This function can be called from either user or kernel context.
146 If the
147 .Dv VM_NOSLEEP
148 flag is specified, it may also be called from interrupt context.
149 .Sh RETURN VALUES
150 Upon successful completion the
151 .Fn vmem_alloc
152 and
153 .Fn vmem_xalloc
154 functions return a pointer to the beginning of the allocated segment. In the
155 case of a
156 .Dv VMC_IDENTIFIER
157 arena, the address of this pointer is the meaningful component, not the value
158 to which it points.
159 .Pp
160 On failure,
161 .Dv NULL
162 is returned.
163 When the
164 .Dv VM_SLEEP
165 flag is specified, these functions can never fail (but may block forever).
166 .Sh SEE ALSO
167 .Xr vmem 9 ,
168 .Xr vmem_create 9F
169 #endif /* ! codereview */
```

1266 Thu Feb 2 13:24:16 2017

new/usr/src/man/man9f/vmem_contains.9f

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```
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 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_CONTAINS 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_contains ,
19 .Nd check for membership in a vmem arena
20 .Sh SYNOPSIS
21 .In sys/vmem.h
22 .Ft int
23 .Fo vmem_contains
24 .Fa "vmem_t *vmp"
25 .Fa "void *vaddr"
26 .Fa "size_t *size"
27 .Fc
28 .Sh INTERFACE LEVEL
29 illumos DDI specific
30 .Sh PARAMETERS
31 .Bl -tag -width Ds
32 .It Fa vmp
33 The vmem arena
34 .It Fa vaddr
35 address of the segment to query.
36 .It Fa size
37 size of the segment to query
38 .El
39 .Sh DESCRIPTION
40 The
41 .Fn vmem_contains
42 function checks whether a segment of
43 .Fa size
44 bytes at
45 .Fa vaddr
46 exists within the vmem arena
47 .Fa vmp .
48 .Sh CONTEXT
49 This function may be called from user or kernel context.
50 .Sh RETURN VALUES
51 .Fn vmem_contains
52 returns non-0 if the segment exists, and 0 otherwise.
53 .Sh SEE ALSO
54 .Xr vmem 9 ,
55 .Xr vmem_add 9F ,
56 .Xr vmem_alloc 9F ,
57 .Xr vmem_create 9F
58 #endif /* !codereview */
```

3536 Thu Feb 2 13:24:17 2017

new/usr/src/man/man9f/vmem_create.9f

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```

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 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_CREATE 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_create ,
19 .Nm vmem_xcreate ,
20 .Nm vmem_destroy
21 .Nd create and destroy vmem arenas
22 .Sh SYNOPSIS
23 .In sys/vmem.h
24 .Vt "typedef struct vmem vmem_t;"
25 .Vt "typedef void *(vmem_alloc_t)(vmem_t *, size_t, int);"
26 .Vt "typedef void (vmem_free_t)(vmem_t *, void *, size_t);"
27 .Vt "typedef void *(vmem_ximport_t)(vmem_t *, size_t *, size_t, int);"
28 .Ft vmem_t *
29 .Fo vmem_create
30 .Fa "const char *name"
31 .Fa "void *base"
32 .Fa "size_t size"
33 .Fa "size_t quantum"
34 .Fa "vmem_alloc_t *afunc"
35 .Fa "vmem_free_t *ffunc"
36 .Fa "vmem_t *source"
37 .Fa "size_t qcache_max"
38 .Fa "int vmflag"
39 .Fc
40 .Ft vmem_t *
41 .Fo vmem_xcreate
42 .Fa "const char *name"
43 .Fa "void *base"
44 .Fa "size_t size"
45 .Fa "size_t quantum"
46 .Fa "vmem_ximport_t *afunc"
47 .Fa "vmem_free_t *ffunc"
48 .Fa "vmem_t *source"
49 .Fa "size_t qcache_max"
50 .Fa "int vmflag"
51 .Fc
52 .Ft void
53 .Fo vmem_destroy
54 .Fa "vmem_t *vmp"
55 .Fc
56 .Sh INTERFACE LEVEL
57 illumos DDI specific
58 .Sh PARAMETERS
59 .Bl -tag -width Ds
60 .It Fa name

```

```

61 A character string giving a name to the vmem
62 arena to be created.
63 .It Fa base
64 An address indicating the lowest possible value in the arena.
65 .It Fa size
66 The size of the arena to create
67 .It Fa quantum
68 The arena's
69 .Dq quantum .
70 The granularity of the arena. The amount allocated at minimum by each request.
71 .It Fa afunc
72 A function which is called to import new spans from
73 .Fa source .
74 Which may be
75 .Dv NULL
76 if this arena does not import from another.
77 .It Fa ffunc
78 A function which is called to return spans to
79 .Fa source .
80 Which may be
81 .Dv NULL
82 if this arena does not import from another.
83 .It Fa source
84 An arena from which this arena will import.
85 Which may be
86 .Dv NULL
87 if this arena does not import from another.
88 .It Fa qcache_max
89 Each arena offers caching of integer multiples of
90 .Fa quantum
91 up to
92 .Fa qcache_max .
93 .It Fa vmflag
94 A bitmask of flags indicating the characteristics of this arena.
95 .Bl -tag -width Ds
96 .It Dv VMC_IDENTIFIER
97 The arena represents arbitrary integer identifiers, rather than virtual
98 memory.
99 .El
100 .It Fa vmp
101 A pointer to the vmem arena to be destroyed.
102 .El
103 .Sh DESCRIPTION
104 A
105 .Em vmem arena
106 is a section of an arbitrary address space (a range of integer addresses).
107 This commonly represents virtual memory, but can in fact be an arbitrary set
108 of integers. The
109 .Dv VMC_IDENTIFIER
110 flag set at arena creation time differentiates between these two cases.
111 .Pp
112 The
113 .Fa afunc ,
114 .Fa ffunc , and
115 .Fa source
116 arguments combine to support a hierarchical structure of arenas, each
117 importing from a single parent (the
118 .Fa source ) .
119 The
120 .Fn vmem_create
121 and
122 .Fn vmem_xcreate
123 functions differ in that the latter provides an interface for
124 .Fa afunc
125 to alter the size of the span imported from
126 .Fa source .

```

```
127 It is only legal to
128 .Em increase
129 thise size.
130 .Sh CONTEXT
131 These functions can be called from user or kernel context.
132 .Sh RETURN VALUES
133 Upon successful completion the
134 .Fn vmem_create and
135 .Fn vmem_xcreate
136 functions return a pointer to a vmem arena. Otherwise,
137 .Dv NULL
138 is returned to indicate the arena could not be created.
139 .Sh SEE ALSO
140 .Xr vmem 9 ,
141 .Xr vmem_add 9F ,
142 .Xr vmem_alloc 9F
143 #endif /* ! codereview */
```

1823 Thu Feb 2 13:24:17 2017

new/usr/src/man/man9f/vmem_walk.9f

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```

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 .\" Copyright 2017, Richard Lowe.
13 .\"
14 .Dd Jan 18, 2017
15 .Dt VMEM_WALK 9F
16 .Os
17 .Sh NAME
18 .Nm vmem_walk ,
19 .Nm vmem_size
20 .Nd walk a (sub\-)set of the segments in a vmem arena
21 .Sh SYNOPSIS
22 .In sys/vmem.h
23 .Ft void
24 .Fo vmem_walk
25 .Fa "vmem_t *vmp"
26 .Fa "int typemask"
27 .Fa "void (*func)(void *, void *, size_t)"
28 .Fa "void *arg"
29 .Fc
30 .Ft size_t
31 .Fo vmem_size
32 .Fa "vmem_t *vmp"
33 .Fa "int typemask"
34 .Fc
35 .Sh INTERFACE LEVEL
36 illumos DDI specific
37 .Sh PARAMETERS
38 .Bl -tag -width Ds
39 .It Fa vmp
40 The vmem arena to walk
41 .It Fa typemask
42 A bitmask indicating the types of segment to operate on
43 .Bl -tag -width Ds
44 .It Dv VMEM_ALLOC
45 Allocated segments
46 .It Dv VMEM_FREE
47 Free segments
48 .El
49 .It Fa func
50 The function to apply to each segment matching
51 .Fa typemask .
52 .Fa func
53 should be of 3 values and return void:
54 .Bl -tag -width Ds
55 .It Fa "void *arg"
56 The
57 .Fa arg
58 passed to
59 .Fn vmem_walk
60 .It Fa "void *vaddr"

```

```

61 The base address of the segment
62 .It Fa "size_t size"
63 The size of the segment
64 .El
65 .It Fa arg
66 An arbitrary argument passed to each call to
67 .Fn func
68 .El
69 .Sh DESCRIPTION
70 .Fn vmem_walk
71 walks each segment in the arena
72 .Fa vmp
73 and applies
74 .Fa func
75 to each which matches
76 .Fa typemask .
77 .Pp
78 .Fn vmem_size
79 walks each segment in the arena
80 .Fa vmp
81 and totals the size of each matching
82 .Fa typemask .
83 .Sh CONTEXT
84 This function may be called from user or kernel context.
85 .Sh SEE ALSO
86 .Xr vmem 9 ,
87 .Xr vmem_alloc 9F ,
88 .Xr vmem_create 9F
89 #endif /* ! codereview */

```

new/usr/src/pkg/manifests/system-kernel.man9.inc

1

623 Thu Feb 2 13:24:19 2017

new/usr/src/pkg/manifests/system-kernel.man9.inc

7831 want vmem manual pages

7832 big theory statements need a place in the manual

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

9 # at <http://www.illumos.org/license/CDDL>.

10 #

12 #

13 # **Copyright 2017, Richard Lowe**

13 # *Copyright 2011, Richard Lowe*

14 # Copyright 2012 Nexenta Systems, Inc. All rights reserved.

15 #

16 **file path=usr/share/man/man9/Intro.9**

17 **file path=usr/share/man/man9/vmem.9**

18 **link path=usr/share/man/man9/intro.9 target=Intro.9**

17 *link path=usr/share/man/man9/Intro.9 target=../man9e/Intro.9e*

18 *link path=usr/share/man/man9/intro.9 target=../man9e/Intro.9e*

new/usr/src/pkg/manifests/system-kernel.man9f.inc

1

73025 Thu Feb 2 13:24:20 2017

new/usr/src/pkg/manifests/system-kernel.man9f.inc

7831 want vmem manual pages

7832 big theory statements need a place in the manual

```
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
9 # at http://www.illumos.org/license/CDDL.
10 #
```

```
12 #
13 # Copyright 2017, Richard Lowe
14 # Copyright 2011, Richard Lowe
15 # Copyright 2014 Garrett D'Amore <garrett@damore.org>
16 # Copyright 2016 Nexenta Systems, Inc.
17 # Copyright 2016 Hans Rosenfeld <rosenfeld@grumpf.hope-2000.org>
```

```
19 file path=usr/share/man/man9f/ASSERT.9f
20 file path=usr/share/man/man9f/Intro.9f
21 file path=usr/share/man/man9f/OTHERQ.9f
22 file path=usr/share/man/man9f/RD.9f
23 file path=usr/share/man/man9f/SAMESTR.9f
24 file path=usr/share/man/man9f/STRUCT_DECL.9f
25 file path=usr/share/man/man9f/WR.9f
26 file path=usr/share/man/man9f/adjmsg.9f
27 file path=usr/share/man/man9f/allpcb.9f
28 file path=usr/share/man/man9f/atomic_add.9f
29 file path=usr/share/man/man9f/atomic_and.9f
30 file path=usr/share/man/man9f/atomic_bits.9f
31 file path=usr/share/man/man9f/atomic_cas.9f
32 file path=usr/share/man/man9f/atomic_dec.9f
33 file path=usr/share/man/man9f/atomic_inc.9f
34 file path=usr/share/man/man9f/atomic_ops.9f
35 file path=usr/share/man/man9f/atomic_or.9f
36 file path=usr/share/man/man9f/atomic_swap.9f
37 file path=usr/share/man/man9f/avl.9f
38 file path=usr/share/man/man9f/backq.9f
39 file path=usr/share/man/man9f/bcanput.9f
40 file path=usr/share/man/man9f/bcmp.9f
41 file path=usr/share/man/man9f/bcopy.9f
42 file path=usr/share/man/man9f/bioclone.9f
43 file path=usr/share/man/man9f/biodone.9f
44 file path=usr/share/man/man9f/bioerror.9f
45 file path=usr/share/man/man9f/biofini.9f
46 file path=usr/share/man/man9f/bioinit.9f
47 file path=usr/share/man/man9f/biomodified.9f
48 file path=usr/share/man/man9f/bioreset.9f
49 file path=usr/share/man/man9f/biosize.9f
50 file path=usr/share/man/man9f/biowait.9f
51 file path=usr/share/man/man9f/bp_copyin.9f
52 file path=usr/share/man/man9f/bp_copyout.9f
53 file path=usr/share/man/man9f/bp_mapin.9f
54 file path=usr/share/man/man9f/bp_mapout.9f
55 file path=usr/share/man/man9f/btop.9f
56 file path=usr/share/man/man9f/btopr.9f
57 file path=usr/share/man/man9f/bufcall.9f
58 file path=usr/share/man/man9f/bzero.9f
59 file path=usr/share/man/man9f/canput.9f
```

new/usr/src/pkg/manifests/system-kernel.man9f.inc

2

```
60 file path=usr/share/man/man9f/canputnext.9f
61 file path=usr/share/man/man9f/clrbuf.9f
62 file path=usr/share/man/man9f/cmn_err.9f
63 file path=usr/share/man/man9f/condvar.9f
64 file path=usr/share/man/man9f/copyb.9f
65 file path=usr/share/man/man9f/copyin.9f
66 file path=usr/share/man/man9f/copymsg.9f
67 file path=usr/share/man/man9f/copyout.9f
68 file path=usr/share/man/man9f/csx_AccessConfigurationRegister.9f
69 file path=usr/share/man/man9f/csx_CS_DDI_Info.9f
70 file path=usr/share/man/man9f/csx_ConvertSize.9f
71 file path=usr/share/man/man9f/csx_ConvertSpeed.9f
72 file path=usr/share/man/man9f/csx_DeregisterClient.9f
73 file path=usr/share/man/man9f/csx_DupHandle.9f
74 file path=usr/share/man/man9f/csx_Error2Text.9f
75 file path=usr/share/man/man9f/csx_Event2Text.9f
76 file path=usr/share/man/man9f/csx_FreeHandle.9f
77 file path=usr/share/man/man9f/csx_Get8.9f
78 file path=usr/share/man/man9f/csx_GetFirstClient.9f
79 file path=usr/share/man/man9f/csx_GetFirstTuple.9f
80 file path=usr/share/man/man9f/csx_GetHandleOffset.9f
81 file path=usr/share/man/man9f/csx_GetMappedAddr.9f
82 file path=usr/share/man/man9f/csx_GetStatus.9f
83 file path=usr/share/man/man9f/csx_GetTupleData.9f
84 file path=usr/share/man/man9f/csx_MakeDeviceNode.9f
85 file path=usr/share/man/man9f/csx_MapLogSocket.9f
86 file path=usr/share/man/man9f/csx_MapMemPage.9f
87 file path=usr/share/man/man9f/csx_ModifyConfiguration.9f
88 file path=usr/share/man/man9f/csx_ModifyWindow.9f
89 file path=usr/share/man/man9f/csx_ParseTuple.9f
90 file path=usr/share/man/man9f/csx_Parse_CISTPL_BATTERY.9f
91 file path=usr/share/man/man9f/csx_Parse_CISTPL_BYTEORDER.9f
92 file path=usr/share/man/man9f/csx_Parse_CISTPL_CFTABLE_ENTRY.9f
93 file path=usr/share/man/man9f/csx_Parse_CISTPL_CONFIG.9f
94 file path=usr/share/man/man9f/csx_Parse_CISTPL_DATE.9f
95 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE.9f
96 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICEGEO.9f
97 file path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICEGEO_A.9f
98 file path=usr/share/man/man9f/csx_Parse_CISTPL_FORMAT.9f
99 file path=usr/share/man/man9f/csx_Parse_CISTPL_FUNC.9f
100 file path=usr/share/man/man9f/csx_Parse_CISTPL_FUNCID.9f
101 file path=usr/share/man/man9f/csx_Parse_CISTPL_GEOMETRY.9f
102 file path=usr/share/man/man9f/csx_Parse_CISTPL_JEDEC_C.9f
103 file path=usr/share/man/man9f/csx_Parse_CISTPL_LINKTARGET.9f
104 file path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_A.9f
105 file path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_MFC.9f
106 file path=usr/share/man/man9f/csx_Parse_CISTPL_MANFID.9f
107 file path=usr/share/man/man9f/csx_Parse_CISTPL_ORG.9f
108 file path=usr/share/man/man9f/csx_Parse_CISTPL_SPCL.9f
109 file path=usr/share/man/man9f/csx_Parse_CISTPL_SWIL.9f
110 file path=usr/share/man/man9f/csx_Parse_CISTPL_VERS_1.9f
111 file path=usr/share/man/man9f/csx_Parse_CISTPL_VERS_2.9f
112 file path=usr/share/man/man9f/csx_Put8.9f
113 file path=usr/share/man/man9f/csx_RegisterClient.9f
114 file path=usr/share/man/man9f/csx_ReleaseConfiguration.9f
115 file path=usr/share/man/man9f/csx_RepGet8.9f
116 file path=usr/share/man/man9f/csx_RepPut8.9f
117 file path=usr/share/man/man9f/csx_RequestConfiguration.9f
118 file path=usr/share/man/man9f/csx_RequestIO.9f
119 file path=usr/share/man/man9f/csx_RequestIRQ.9f
120 file path=usr/share/man/man9f/csx_RequestSocketMask.9f
121 file path=usr/share/man/man9f/csx_RequestWindow.9f
122 file path=usr/share/man/man9f/csx_ResetFunction.9f
123 file path=usr/share/man/man9f/csx_SetEventMask.9f
124 file path=usr/share/man/man9f/csx_SetHandleOffset.9f
125 file path=usr/share/man/man9f/csx_ValidateCIS.9f
```

```

126 file path=usr/share/man/man9f/datams9.9f
127 file path=usr/share/man/man9f/ddi_add_event_handler.9f
128 file path=usr/share/man/man9f/ddi_add_intr.9f
129 file path=usr/share/man/man9f/ddi_add_softintr.9f
130 file path=usr/share/man/man9f/ddi_binding_name.9f
131 file path=usr/share/man/man9f/ddi_btop.9f
132 file path=usr/share/man/man9f/ddi_can_receive_sig.9f
133 file path=usr/share/man/man9f/ddi_cb_register.9f
134 file path=usr/share/man/man9f/ddi_check_acc_handle.9f
135 file path=usr/share/man/man9f/ddi_copyin.9f
136 file path=usr/share/man/man9f/ddi_copyout.9f
137 file path=usr/share/man/man9f/ddi_create_minor_node.9f
138 file path=usr/share/man/man9f/ddi_cred.9f
139 file path=usr/share/man/man9f/ddi_dev_is_needed.9f
140 file path=usr/share/man/man9f/ddi_dev_is_sid.9f
141 file path=usr/share/man/man9f/ddi_dev_nintrs.9f
142 file path=usr/share/man/man9f/ddi_dev_nregs.9f
143 file path=usr/share/man/man9f/ddi_dev_regsize.9f
144 file path=usr/share/man/man9f/ddi_dev_report_fault.9f
145 file path=usr/share/man/man9f/ddi_device_copy.9f
146 file path=usr/share/man/man9f/ddi_device_zero.9f
147 file path=usr/share/man/man9f/ddi_devid_compare.9f
148 file path=usr/share/man/man9f/ddi_dma_addr_bind_handle.9f
149 file path=usr/share/man/man9f/ddi_dma_alloc_handle.9f
150 file path=usr/share/man/man9f/ddi_dma_buf_bind_handle.9f
151 file path=usr/share/man/man9f/ddi_dma_burstsizes.9f
152 file path=usr/share/man/man9f/ddi_dma_free_handle.9f
153 file path=usr/share/man/man9f/ddi_dma_getwin.9f
154 file path=usr/share/man/man9f/ddi_dma_mem_alloc.9f
155 file path=usr/share/man/man9f/ddi_dma_mem_free.9f
156 file path=usr/share/man/man9f/ddi_dma_nextcookie.9f
157 file path=usr/share/man/man9f/ddi_dma_numwin.9f
158 file path=usr/share/man/man9f/ddi_dma_set_sbust64.9f
159 file path=usr/share/man/man9f/ddi_dma_sync.9f
160 file path=usr/share/man/man9f/ddi_dma_unbind_handle.9f
161 file path=usr/share/man/man9f/ddi_dmae.9f
162 file path=usr/share/man/man9f/ddi_driver_major.9f
163 file path=usr/share/man/man9f/ddi_driver_name.9f
164 file path=usr/share/man/man9f/ddi_enter_critical.9f
165 file path=usr/share/man/man9f/ddi_ffs.9f
166 file path=usr/share/man/man9f/ddi_fm_acc_err_clear.9f
167 file path=usr/share/man/man9f/ddi_fm_acc_err_get.9f
168 file path=usr/share/man/man9f/ddi_fm_ereport_post.9f
169 file path=usr/share/man/man9f/ddi_fm_handler_register.9f
170 file path=usr/share/man/man9f/ddi_fm_init.9f
171 file path=usr/share/man/man9f/ddi_fm_service_impact.9f
172 file path=usr/share/man/man9f/ddi_get8.9f
173 file path=usr/share/man/man9f/ddi_get_cred.9f
174 file path=usr/share/man/man9f/ddi_get_devstate.9f
175 file path=usr/share/man/man9f/ddi_get_driver_private.9f
176 file path=usr/share/man/man9f/ddi_get_eventcookie.9f
177 file path=usr/share/man/man9f/ddi_get_instance.9f
178 file path=usr/share/man/man9f/ddi_get_kt_did.9f
179 file path=usr/share/man/man9f/ddi_get_lbolt.9f
180 file path=usr/share/man/man9f/ddi_get_parent.9f
181 file path=usr/share/man/man9f/ddi_get_pid.9f
182 file path=usr/share/man/man9f/ddi_get_time.9f
183 file path=usr/share/man/man9f/ddi_getiminor.9f
184 file path=usr/share/man/man9f/ddi_in_panic.9f
185 file path=usr/share/man/man9f/ddi_intr_add_handler.9f
186 file path=usr/share/man/man9f/ddi_intr_add_softint.9f
187 file path=usr/share/man/man9f/ddi_intr_alloc.9f
188 file path=usr/share/man/man9f/ddi_intr_dup_handler.9f
189 file path=usr/share/man/man9f/ddi_intr_enable.9f
190 file path=usr/share/man/man9f/ddi_intr_get_cap.9f
191 file path=usr/share/man/man9f/ddi_intr_get_hilevel_pri.9f

```

```

192 file path=usr/share/man/man9f/ddi_intr_get_nintrs.9f
193 file path=usr/share/man/man9f/ddi_intr_get_pending.9f
194 file path=usr/share/man/man9f/ddi_intr_get_pri.9f
195 file path=usr/share/man/man9f/ddi_intr_get_supported_types.9f
196 file path=usr/share/man/man9f/ddi_intr_hilevel.9f
197 file path=usr/share/man/man9f/ddi_intr_set_mask.9f
198 file path=usr/share/man/man9f/ddi_intr_set_nreq.9f
199 file path=usr/share/man/man9f/ddi_io_get8.9f
200 file path=usr/share/man/man9f/ddi_io_put8.9f
201 file path=usr/share/man/man9f/ddi_io_rep_get8.9f
202 file path=usr/share/man/man9f/ddi_io_rep_put8.9f
203 file path=usr/share/man/man9f/ddi_log_sysevent.9f
204 file path=usr/share/man/man9f/ddi_map_regs.9f
205 file path=usr/share/man/man9f/ddi_mem_get8.9f
206 file path=usr/share/man/man9f/ddi_mem_put8.9f
207 file path=usr/share/man/man9f/ddi_mem_rep_get8.9f
208 file path=usr/share/man/man9f/ddi_mem_rep_put8.9f
209 file path=usr/share/man/man9f/ddi_mmap_get_model.9f
210 file path=usr/share/man/man9f/ddi_model_convert_from.9f
211 file path=usr/share/man/man9f/ddi_modopen.9f
212 file path=usr/share/man/man9f/ddi_no_info.9f
213 file path=usr/share/man/man9f/ddi_node_name.9f
214 file path=usr/share/man/man9f/ddi_peek.9f
215 file path=usr/share/man/man9f/ddi_periodic_add.9f
216 file path=usr/share/man/man9f/ddi_periodic_delete.9f
217 file path=usr/share/man/man9f/ddi_poke.9f
218 file path=usr/share/man/man9f/ddi_prop_create.9f
219 file path=usr/share/man/man9f/ddi_prop_exists.9f
220 file path=usr/share/man/man9f/ddi_prop_get_int.9f
221 file path=usr/share/man/man9f/ddi_prop_lookup.9f
222 file path=usr/share/man/man9f/ddi_prop_op.9f
223 file path=usr/share/man/man9f/ddi_prop_update.9f
224 file path=usr/share/man/man9f/ddi_put8.9f
225 file path=usr/share/man/man9f/ddi_regs_map_free.9f
226 file path=usr/share/man/man9f/ddi_regs_map_setup.9f
227 file path=usr/share/man/man9f/ddi_remove_event_handler.9f
228 file path=usr/share/man/man9f/ddi_remove_minor_node.9f
229 file path=usr/share/man/man9f/ddi_removing_power.9f
230 file path=usr/share/man/man9f/ddi_rep_get8.9f
231 file path=usr/share/man/man9f/ddi_rep_put8.9f
232 file path=usr/share/man/man9f/ddi_report_dev.9f
233 file path=usr/share/man/man9f/ddi_root_node.9f
234 file path=usr/share/man/man9f/ddi_segmap.9f
235 file path=usr/share/man/man9f/ddi_slaveonly.9f
236 file path=usr/share/man/man9f/ddi_soft_state.9f
237 file path=usr/share/man/man9f/ddi_strtol.9f
238 file path=usr/share/man/man9f/ddi_strtoll.9f
239 file path=usr/share/man/man9f/ddi_strtoul.9f
240 file path=usr/share/man/man9f/ddi_umem_alloc.9f
241 file path=usr/share/man/man9f/ddi_umem_iosetup.9f
242 file path=usr/share/man/man9f/ddi_umem_lock.9f
243 file path=usr/share/man/man9f/delay.9f
244 file path=usr/share/man/man9f/devfs_clean.9f
245 file path=usr/share/man/man9f/devmap_default_access.9f
246 file path=usr/share/man/man9f/devmap_devmem_setup.9f
247 file path=usr/share/man/man9f/devmap_do_ctxmgt.9f
248 file path=usr/share/man/man9f/devmap_set_ctx_timeout.9f
249 file path=usr/share/man/man9f/devmap_setup.9f
250 file path=usr/share/man/man9f/devmap_unload.9f
251 file path=usr/share/man/man9f/disksort.9f
252 file path=usr/share/man/man9f/dlbindack.9f
253 file path=usr/share/man/man9f/drv_getparm.9f
254 file path=usr/share/man/man9f/drv_hztousec.9f
255 file path=usr/share/man/man9f/drv_priv.9f
256 file path=usr/share/man/man9f/drv_usectohz.9f
257 file path=usr/share/man/man9f/drv_usecwait.9f

```


258 file path=usr/share/man/man9f/dupb.9f
259 file path=usr/share/man/man9f/dupmsg.9f
260 file path=usr/share/man/man9f/enableok.9f
261 file path=usr/share/man/man9f/esballoc.9f
262 file path=usr/share/man/man9f/esbcall.9f
263 file path=usr/share/man/man9f/firmload.9f
264 file path=usr/share/man/man9f/flushband.9f
265 file path=usr/share/man/man9f/flushq.9f
266 file path=usr/share/man/man9f/freeb.9f
267 file path=usr/share/man/man9f/freemsg.9f
268 file path=usr/share/man/man9f/freerbuf.9f
269 file path=usr/share/man/man9f/freezestr.9f
270 file path=usr/share/man/man9f/get_pktiopb.9f
271 file path=usr/share/man/man9f/geterror.9f
272 file path=usr/share/man/man9f/gethrtime.9f
273 file path=usr/share/man/man9f/getmajor.9f
274 file path=usr/share/man/man9f/getminor.9f
275 file path=usr/share/man/man9f/getq.9f
276 file path=usr/share/man/man9f/getrbuf.9f
277 file path=usr/share/man/man9f/gld.9f
278 file path=usr/share/man/man9f/hook_alloc.9f
279 file path=usr/share/man/man9f/hook_free.9f
280 file path=usr/share/man/man9f/id32_alloc.9f
281 file path=usr/share/man/man9f/id_space.9f
282 file path=usr/share/man/man9f/inb.9f
283 file path=usr/share/man/man9f/insq.9f
284 file path=usr/share/man/man9f/kiconv.9f
285 file path=usr/share/man/man9f/kiconv_close.9f
286 file path=usr/share/man/man9f/kiconv_open.9f
287 file path=usr/share/man/man9f/kiconvstr.9f
288 file path=usr/share/man/man9f/kmem_alloc.9f
289 file path=usr/share/man/man9f/kmem_cache_create.9f
290 file path=usr/share/man/man9f/kstat_create.9f
291 file path=usr/share/man/man9f/kstat_delete.9f
292 file path=usr/share/man/man9f/kstat_install.9f
293 file path=usr/share/man/man9f/kstat_named_init.9f
294 file path=usr/share/man/man9f/kstat_queue.9f
295 file path=usr/share/man/man9f/ldi_add_event_handler.9f
296 file path=usr/share/man/man9f/ldi_aread.9f
297 file path=usr/share/man/man9f/ldi_devmap.9f
298 file path=usr/share/man/man9f/ldi_dump.9f
299 file path=usr/share/man/man9f/ldi_ev_finalize.9f
300 file path=usr/share/man/man9f/ldi_ev_get_cookie.9f
301 file path=usr/share/man/man9f/ldi_ev_get_type.9f
302 file path=usr/share/man/man9f/ldi_ev_notify.9f
303 file path=usr/share/man/man9f/ldi_ev_register_callbacks.9f
304 file path=usr/share/man/man9f/ldi_ev_remove_callbacks.9f
305 file path=usr/share/man/man9f/ldi_get_dev.9f
306 file path=usr/share/man/man9f/ldi_get_eventcookie.9f
307 file path=usr/share/man/man9f/ldi_get_size.9f
308 file path=usr/share/man/man9f/ldi_ident_from_dev.9f
309 file path=usr/share/man/man9f/ldi_ioctl.9f
310 file path=usr/share/man/man9f/ldi_open_by_dev.9f
311 file path=usr/share/man/man9f/ldi_poll.9f
312 file path=usr/share/man/man9f/ldi_prop_exists.9f
313 file path=usr/share/man/man9f/ldi_prop_get_int.9f
314 file path=usr/share/man/man9f/ldi_prop_lookup_int_array.9f
315 file path=usr/share/man/man9f/ldi_putmsg.9f
316 file path=usr/share/man/man9f/ldi_read.9f
317 file path=usr/share/man/man9f/ldi_remove_event_handler.9f
318 file path=usr/share/man/man9f/ldi_strategy.9f
319 file path=usr/share/man/man9f/linkb.9f
320 file path=usr/share/man/man9f/list_create.9f
321 file path=usr/share/man/man9f/mac_alloc.9f
322 file path=usr/share/man/man9f/mac_hcksum_get.9f
323 file path=usr/share/man/man9f/mac_init_ops.9f

324 file path=usr/share/man/man9f/mac_link_update.9f
325 file path=usr/share/man/man9f/mac_lso_get.9f
326 file path=usr/share/man/man9f/mac_maxsdu_update.9f
327 file path=usr/share/man/man9f/mac_prop_info.9f
328 file path=usr/share/man/man9f/mac_register.9f
329 file path=usr/share/man/man9f/mac_rx.9f
330 file path=usr/share/man/man9f/mac_tx_update.9f
331 file path=usr/share/man/man9f/makecom.9f
332 file path=usr/share/man/man9f/makedevice.9f
333 file path=usr/share/man/man9f/max.9f
334 file path=usr/share/man/man9f/mcopyin.9f
335 file path=usr/share/man/man9f/mcopymsg.9f
336 file path=usr/share/man/man9f/mcopyout.9f
337 file path=usr/share/man/man9f/membar_ops.9f
338 file path=usr/share/man/man9f/memchr.9f
339 file path=usr/share/man/man9f/merror.9f
340 file path=usr/share/man/man9f/mexchange.9f
341 file path=usr/share/man/man9f/min.9f
342 file path=usr/share/man/man9f/mioc2ack.9f
343 file path=usr/share/man/man9f/miocack.9f
344 file path=usr/share/man/man9f/miocnak.9f
345 file path=usr/share/man/man9f/miocpullup.9f
346 file path=usr/share/man/man9f/mkiocb.9f
347 file path=usr/share/man/man9f/mod_install.9f
348 file path=usr/share/man/man9f/msgdsize.9f
349 file path=usr/share/man/man9f/msgpullup.9f
350 file path=usr/share/man/man9f/msgsize.9f
351 file path=usr/share/man/man9f/mt-streams.9f
352 file path=usr/share/man/man9f/mutex.9f
353 file path=usr/share/man/man9f/net_event_notify_register.9f
354 file path=usr/share/man/man9f/net_getifname.9f
355 file path=usr/share/man/man9f/net_getlifaddr.9f
356 file path=usr/share/man/man9f/net_getmtu.9f
357 file path=usr/share/man/man9f/net_getnetid.9f
358 file path=usr/share/man/man9f/net_getpmtuenabled.9f
359 file path=usr/share/man/man9f/net_hook_register.9f
360 file path=usr/share/man/man9f/net_hook_unregister.9f
361 file path=usr/share/man/man9f/net_inject.9f
362 file path=usr/share/man/man9f/net_inject_alloc.9f
363 file path=usr/share/man/man9f/net_inject_free.9f
364 file path=usr/share/man/man9f/net_instance_alloc.9f
365 file path=usr/share/man/man9f/net_instance_free.9f
366 file path=usr/share/man/man9f/net_instance_notify_register.9f
367 file path=usr/share/man/man9f/net_instance_register.9f
368 file path=usr/share/man/man9f/net_instance_unregister.9f
369 file path=usr/share/man/man9f/net_ispartialchecksum.9f
370 file path=usr/share/man/man9f/net_isvalidchecksum.9f
371 file path=usr/share/man/man9f/net_kstat_create.9f
372 file path=usr/share/man/man9f/net_kstat_delete.9f
373 file path=usr/share/man/man9f/net_lifgetnext.9f
374 file path=usr/share/man/man9f/net_netidtozonid.9f
375 file path=usr/share/man/man9f/net_phygetnext.9f
376 file path=usr/share/man/man9f/net_phylookup.9f
377 file path=usr/share/man/man9f/net_protocol_lookup.9f
378 file path=usr/share/man/man9f/net_protocol_notify_register.9f
379 file path=usr/share/man/man9f/net_protocol_release.9f
380 file path=usr/share/man/man9f/net_protocol_walk.9f
381 file path=usr/share/man/man9f/net_routeto.9f
382 file path=usr/share/man/man9f/net_zoneidtonetid.9f
383 file path=usr/share/man/man9f/netinfo.9f
384 file path=usr/share/man/man9f/nochpoll.9f
385 file path=usr/share/man/man9f/nodev.9f
386 file path=usr/share/man/man9f/noenable.9f
387 file path=usr/share/man/man9f/nulldev.9f
388 file path=usr/share/man/man9f/nvlist_add_boolean.9f
389 file path=usr/share/man/man9f/nvlist_alloc.9f

```

390 file path=usr/share/man/man9f/nvlist_lookup_boolean.9f
391 file path=usr/share/man/man9f/nvlist_lookup_nvpair.9f
392 file path=usr/share/man/man9f/nvlist_next_nvpair.9f
393 file path=usr/share/man/man9f/nvlist_remove.9f
394 file path=usr/share/man/man9f/nvpair_value_byte.9f
395 file path=usr/share/man/man9f/outb.9f
396 file path=usr/share/man/man9f/pci_config_get8.9f
397 file path=usr/share/man/man9f/pci_config_setup.9f
398 file path=usr/share/man/man9f/pci_ereport_setup.9f
399 file path=usr/share/man/man9f/pci_report_pmcaps.9f
400 file path=usr/share/man/man9f/pci_save_config_regs.9f
401 file path=usr/share/man/man9f/physio.9f
402 file path=usr/share/man/man9f/pm_busy_component.9f
403 file path=usr/share/man/man9f/pm_power_has_changed.9f
404 file path=usr/share/man/man9f/pm_raise_power.9f
405 file path=usr/share/man/man9f/pm_trans_check.9f
406 file path=usr/share/man/man9f/pollhead_clean.9f
407 file path=usr/share/man/man9f/pollwakeups.9f
408 file path=usr/share/man/man9f/priv_getbyname.9f
409 file path=usr/share/man/man9f/priv_policy.9f
410 file path=usr/share/man/man9f/proc_signal.9f
411 file path=usr/share/man/man9f/ptob.9f
412 file path=usr/share/man/man9f/pullupmsg.9f
413 file path=usr/share/man/man9f/put.9f
414 file path=usr/share/man/man9f/putbq.9f
415 file path=usr/share/man/man9f/putctl.9f
416 file path=usr/share/man/man9f/putctl1.9f
417 file path=usr/share/man/man9f/putnext.9f
418 file path=usr/share/man/man9f/putnextctl.9f
419 file path=usr/share/man/man9f/putnextctl1.9f
420 file path=usr/share/man/man9f/putq.9f
421 file path=usr/share/man/man9f/qassociate.9f
422 file path=usr/share/man/man9f/qbufcall.9f
423 file path=usr/share/man/man9f/qenable.9f
424 file path=usr/share/man/man9f/qprocson.9f
425 file path=usr/share/man/man9f/qreply.9f
426 file path=usr/share/man/man9f/qsize.9f
427 file path=usr/share/man/man9f/qtimeout.9f
428 file path=usr/share/man/man9f/qunbufcall.9f
429 file path=usr/share/man/man9f/qtimeout.9f
430 file path=usr/share/man/man9f/qwait.9f
431 file path=usr/share/man/man9f/qwriter.9f
432 file path=usr/share/man/man9f/rmalloc.9f
433 file path=usr/share/man/man9f/rmalloc_wait.9f
434 file path=usr/share/man/man9f/rmallocmap.9f
435 file path=usr/share/man/man9f/rmfree.9f
436 file path=usr/share/man/man9f/rmvb.9f
437 file path=usr/share/man/man9f/rmvq.9f
438 file path=usr/share/man/man9f/rwlock.9f
439 file path=usr/share/man/man9f/scsi_abort.9f
440 file path=usr/share/man/man9f/scsi_alloc_consistent_buf.9f
441 file path=usr/share/man/man9f/scsi_cname.9f
442 file path=usr/share/man/man9f/scsi_destroy_pkt.9f
443 file path=usr/share/man/man9f/scsi_dmagent.9f
444 file path=usr/share/man/man9f/scsi_errmsg.9f
445 file path=usr/share/man/man9f/scsi_ext_sense_fields.9f
446 file path=usr/share/man/man9f/scsi_find_sense_descr.9f
447 file path=usr/share/man/man9f/scsi_free_consistent_buf.9f
448 file path=usr/share/man/man9f/scsi_get_device_type_scsi_options.9f
449 file path=usr/share/man/man9f/scsi_get_device_type_string.9f
450 file path=usr/share/man/man9f/scsi_hba_attach_setup.9f
451 file path=usr/share/man/man9f/scsi_hba_init.9f
452 file path=usr/share/man/man9f/scsi_hba_lookup_capstr.9f
453 file path=usr/share/man/man9f/scsi_hba_pkt_alloc.9f
454 file path=usr/share/man/man9f/scsi_hba_pkt_comp.9f
455 file path=usr/share/man/man9f/scsi_hba_probe.9f

```

```

456 file path=usr/share/man/man9f/scsi_hba_tran_alloc.9f
457 file path=usr/share/man/man9f/scsi_ifgetcap.9f
458 file path=usr/share/man/man9f/scsi_init_pkt.9f
459 file path=usr/share/man/man9f/scsi_log.9f
460 file path=usr/share/man/man9f/scsi_pktalloc.9f
461 file path=usr/share/man/man9f/scsi_poll.9f
462 file path=usr/share/man/man9f/scsi_probe.9f
463 file path=usr/share/man/man9f/scsi_reset.9f
464 file path=usr/share/man/man9f/scsi_reset_notify.9f
465 file path=usr/share/man/man9f/scsi_sense_key.9f
466 file path=usr/share/man/man9f/scsi_setup_cdb.9f
467 file path=usr/share/man/man9f/scsi_slave.9f
468 file path=usr/share/man/man9f/scsi_sync_pkt.9f
469 file path=usr/share/man/man9f/scsi_transport.9f
470 file path=usr/share/man/man9f/scsi_unprobe.9f
471 file path=usr/share/man/man9f/scsi_validate_sense.9f
472 file path=usr/share/man/man9f/scsi_vu_errmsg.9f
473 file path=usr/share/man/man9f/semaphore.9f
474 file path=usr/share/man/man9f/sprintf.9f
475 file path=usr/share/man/man9f/stoi.9f
476 file path=usr/share/man/man9f/string.9f
477 file path=usr/share/man/man9f/strlog.9f
478 file path=usr/share/man/man9f/strqget.9f
479 file path=usr/share/man/man9f/strqset.9f
480 file path=usr/share/man/man9f/swab.9f
481 file path=usr/share/man/man9f/taskq.9f
482 file path=usr/share/man/man9f/testb.9f
483 file path=usr/share/man/man9f/timeout.9f
484 file path=usr/share/man/man9f/u8_strcmp.9f
485 file path=usr/share/man/man9f/u8_textprep_str.9f
486 file path=usr/share/man/man9f/u8_validate.9f
487 file path=usr/share/man/man9f/uconv_ul6tou32.9f
488 file path=usr/share/man/man9f/uiomove.9f
489 file path=usr/share/man/man9f/unbufcall.9f
490 file path=usr/share/man/man9f/unlinkb.9f
491 file path=usr/share/man/man9f/untimeout.9f
492 file path=usr/share/man/man9f/ureadc.9f
493 file path=usr/share/man/man9f/uwritec.9f
494 file path=usr/share/man/man9f/va_arg.9f
495 file path=usr/share/man/man9f/vmem_add.9f
496 file path=usr/share/man/man9f/vmem_alloc.9f
497 file path=usr/share/man/man9f/vmem_contains.9f
498 file path=usr/share/man/man9f/vmem_create.9f
499 file path=usr/share/man/man9f/vmem_walk.9f
500 #endif /* ! codereview */
501 link path=usr/share/man/man9f/AVL_NEXT.9f target=avl.9f
502 link path=usr/share/man/man9f/AVL_PREV.9f target=avl.9f
503 link path=usr/share/man/man9f/SIZEOF_PTR.9f target=STRUCT_DECL.9f
504 link path=usr/share/man/man9f/SIZEOF_STRUCT.9f target=STRUCT_DECL.9f
505 link path=usr/share/man/man9f/STRUCT_BUF.9f target=STRUCT_DECL.9f
506 link path=usr/share/man/man9f/STRUCT_FADDR.9f target=STRUCT_DECL.9f
507 link path=usr/share/man/man9f/STRUCT_FGET.9f target=STRUCT_DECL.9f
508 link path=usr/share/man/man9f/STRUCT_FGETP.9f target=STRUCT_DECL.9f
509 link path=usr/share/man/man9f/STRUCT_FSET.9f target=STRUCT_DECL.9f
510 link path=usr/share/man/man9f/STRUCT_FSETP.9f target=STRUCT_DECL.9f
511 link path=usr/share/man/man9f/STRUCT_HANDLE.9f target=STRUCT_DECL.9f
512 link path=usr/share/man/man9f/STRUCT_INIT.9f target=STRUCT_DECL.9f
513 link path=usr/share/man/man9f/STRUCT_SET_HANDLE.9f target=STRUCT_DECL.9f
514 link path=usr/share/man/man9f/STRUCT_SIZE.9f target=STRUCT_DECL.9f
515 link path=usr/share/man/man9f/assert.9f target=ASSERT.9f
516 link path=usr/share/man/man9f/atomic_add_16.9f target=atomic_add.9f
517 link path=usr/share/man/man9f/atomic_add_16_nv.9f target=atomic_add.9f
518 link path=usr/share/man/man9f/atomic_add_32.9f target=atomic_add.9f
519 link path=usr/share/man/man9f/atomic_add_32_nv.9f target=atomic_add.9f
520 link path=usr/share/man/man9f/atomic_add_64.9f target=atomic_add.9f
521 link path=usr/share/man/man9f/atomic_add_64_nv.9f target=atomic_add.9f

```

```

522 link path=usr/share/man/man9f/atomic_add_8.9f target=atomic_add.9f
523 link path=usr/share/man/man9f/atomic_add_8_nv.9f target=atomic_add.9f
524 link path=usr/share/man/man9f/atomic_add_char.9f target=atomic_add.9f
525 link path=usr/share/man/man9f/atomic_add_char_nv.9f target=atomic_add.9f
526 link path=usr/share/man/man9f/atomic_add_int.9f target=atomic_add.9f
527 link path=usr/share/man/man9f/atomic_add_int_nv.9f target=atomic_add.9f
528 link path=usr/share/man/man9f/atomic_add_long.9f target=atomic_add.9f
529 link path=usr/share/man/man9f/atomic_add_long_nv.9f target=atomic_add.9f
530 link path=usr/share/man/man9f/atomic_add_ptr.9f target=atomic_add.9f
531 link path=usr/share/man/man9f/atomic_add_ptr_nv.9f target=atomic_add.9f
532 link path=usr/share/man/man9f/atomic_add_short.9f target=atomic_add.9f
533 link path=usr/share/man/man9f/atomic_add_short_nv.9f target=atomic_add.9f
534 link path=usr/share/man/man9f/atomic_and_16.9f target=atomic_and.9f
535 link path=usr/share/man/man9f/atomic_and_16_nv.9f target=atomic_and.9f
536 link path=usr/share/man/man9f/atomic_and_32.9f target=atomic_and.9f
537 link path=usr/share/man/man9f/atomic_and_32_nv.9f target=atomic_and.9f
538 link path=usr/share/man/man9f/atomic_and_64.9f target=atomic_and.9f
539 link path=usr/share/man/man9f/atomic_and_64_nv.9f target=atomic_and.9f
540 link path=usr/share/man/man9f/atomic_and_8.9f target=atomic_and.9f
541 link path=usr/share/man/man9f/atomic_and_8_nv.9f target=atomic_and.9f
542 link path=usr/share/man/man9f/atomic_and_uchar.9f target=atomic_and.9f
543 link path=usr/share/man/man9f/atomic_and_uchar_nv.9f target=atomic_and.9f
544 link path=usr/share/man/man9f/atomic_and_uint.9f target=atomic_and.9f
545 link path=usr/share/man/man9f/atomic_and_uint_nv.9f target=atomic_and.9f
546 link path=usr/share/man/man9f/atomic_and_ulong.9f target=atomic_and.9f
547 link path=usr/share/man/man9f/atomic_and_ulong_nv.9f target=atomic_and.9f
548 link path=usr/share/man/man9f/atomic_and_ushort.9f target=atomic_and.9f
549 link path=usr/share/man/man9f/atomic_and_ushort_nv.9f target=atomic_and.9f
550 link path=usr/share/man/man9f/atomic_cas_16.9f target=atomic_cas.9f
551 link path=usr/share/man/man9f/atomic_cas_32.9f target=atomic_cas.9f
552 link path=usr/share/man/man9f/atomic_cas_64.9f target=atomic_cas.9f
553 link path=usr/share/man/man9f/atomic_cas_8.9f target=atomic_cas.9f
554 link path=usr/share/man/man9f/atomic_cas_ptr.9f target=atomic_cas.9f
555 link path=usr/share/man/man9f/atomic_cas_uchar.9f target=atomic_cas.9f
556 link path=usr/share/man/man9f/atomic_cas_uint.9f target=atomic_cas.9f
557 link path=usr/share/man/man9f/atomic_cas_ulong.9f target=atomic_cas.9f
558 link path=usr/share/man/man9f/atomic_cas_ushort.9f target=atomic_cas.9f
559 link path=usr/share/man/man9f/atomic_clear_long_excl.9f target=atomic_bits.9f
560 link path=usr/share/man/man9f/atomic_dec_16.9f target=atomic_dec.9f
561 link path=usr/share/man/man9f/atomic_dec_16_nv.9f target=atomic_dec.9f
562 link path=usr/share/man/man9f/atomic_dec_32.9f target=atomic_dec.9f
563 link path=usr/share/man/man9f/atomic_dec_32_nv.9f target=atomic_dec.9f
564 link path=usr/share/man/man9f/atomic_dec_64.9f target=atomic_dec.9f
565 link path=usr/share/man/man9f/atomic_dec_64_nv.9f target=atomic_dec.9f
566 link path=usr/share/man/man9f/atomic_dec_8.9f target=atomic_dec.9f
567 link path=usr/share/man/man9f/atomic_dec_8_nv.9f target=atomic_dec.9f
568 link path=usr/share/man/man9f/atomic_dec_ptr.9f target=atomic_dec.9f
569 link path=usr/share/man/man9f/atomic_dec_ptr_nv.9f target=atomic_dec.9f
570 link path=usr/share/man/man9f/atomic_dec_uchar.9f target=atomic_dec.9f
571 link path=usr/share/man/man9f/atomic_dec_uchar_nv.9f target=atomic_dec.9f
572 link path=usr/share/man/man9f/atomic_dec_uint.9f target=atomic_dec.9f
573 link path=usr/share/man/man9f/atomic_dec_uint_nv.9f target=atomic_dec.9f
574 link path=usr/share/man/man9f/atomic_dec_ulong.9f target=atomic_dec.9f
575 link path=usr/share/man/man9f/atomic_dec_ulong_nv.9f target=atomic_dec.9f
576 link path=usr/share/man/man9f/atomic_dec_ushort.9f target=atomic_dec.9f
577 link path=usr/share/man/man9f/atomic_dec_ushort_nv.9f target=atomic_dec.9f
578 link path=usr/share/man/man9f/atomic_inc_16.9f target=atomic_inc.9f
579 link path=usr/share/man/man9f/atomic_inc_16_nv.9f target=atomic_inc.9f
580 link path=usr/share/man/man9f/atomic_inc_32.9f target=atomic_inc.9f
581 link path=usr/share/man/man9f/atomic_inc_32_nv.9f target=atomic_inc.9f
582 link path=usr/share/man/man9f/atomic_inc_64.9f target=atomic_inc.9f
583 link path=usr/share/man/man9f/atomic_inc_64_nv.9f target=atomic_inc.9f
584 link path=usr/share/man/man9f/atomic_inc_8.9f target=atomic_inc.9f
585 link path=usr/share/man/man9f/atomic_inc_8_nv.9f target=atomic_inc.9f
586 link path=usr/share/man/man9f/atomic_inc_ptr.9f target=atomic_inc.9f
587 link path=usr/share/man/man9f/atomic_inc_ptr_nv.9f target=atomic_inc.9f

```

```

588 link path=usr/share/man/man9f/atomic_inc_uchar.9f target=atomic_inc.9f
589 link path=usr/share/man/man9f/atomic_inc_uchar_nv.9f target=atomic_inc.9f
590 link path=usr/share/man/man9f/atomic_inc_uint.9f target=atomic_inc.9f
591 link path=usr/share/man/man9f/atomic_inc_uint_nv.9f target=atomic_inc.9f
592 link path=usr/share/man/man9f/atomic_inc_ulong.9f target=atomic_inc.9f
593 link path=usr/share/man/man9f/atomic_inc_ulong_nv.9f target=atomic_inc.9f
594 link path=usr/share/man/man9f/atomic_inc_ushort.9f target=atomic_inc.9f
595 link path=usr/share/man/man9f/atomic_inc_ushort_nv.9f target=atomic_inc.9f
596 link path=usr/share/man/man9f/atomic_or_16.9f target=atomic_or.9f
597 link path=usr/share/man/man9f/atomic_or_16_nv.9f target=atomic_or.9f
598 link path=usr/share/man/man9f/atomic_or_32.9f target=atomic_or.9f
599 link path=usr/share/man/man9f/atomic_or_32_nv.9f target=atomic_or.9f
600 link path=usr/share/man/man9f/atomic_or_64.9f target=atomic_or.9f
601 link path=usr/share/man/man9f/atomic_or_64_nv.9f target=atomic_or.9f
602 link path=usr/share/man/man9f/atomic_or_8.9f target=atomic_or.9f
603 link path=usr/share/man/man9f/atomic_or_8_nv.9f target=atomic_or.9f
604 link path=usr/share/man/man9f/atomic_or_uchar.9f target=atomic_or.9f
605 link path=usr/share/man/man9f/atomic_or_uchar_nv.9f target=atomic_or.9f
606 link path=usr/share/man/man9f/atomic_or_uint.9f target=atomic_or.9f
607 link path=usr/share/man/man9f/atomic_or_uint_nv.9f target=atomic_or.9f
608 link path=usr/share/man/man9f/atomic_or_ulong.9f target=atomic_or.9f
609 link path=usr/share/man/man9f/atomic_or_ulong_nv.9f target=atomic_or.9f
610 link path=usr/share/man/man9f/atomic_or_ushort.9f target=atomic_or.9f
611 link path=usr/share/man/man9f/atomic_or_ushort_nv.9f target=atomic_or.9f
612 link path=usr/share/man/man9f/atomic_set_long_excl.9f target=atomic_bits.9f
613 link path=usr/share/man/man9f/atomic_swap_16.9f target=atomic_swap.9f
614 link path=usr/share/man/man9f/atomic_swap_32.9f target=atomic_swap.9f
615 link path=usr/share/man/man9f/atomic_swap_64.9f target=atomic_swap.9f
616 link path=usr/share/man/man9f/atomic_swap_8.9f target=atomic_swap.9f
617 link path=usr/share/man/man9f/atomic_swap_ptr.9f target=atomic_swap.9f
618 link path=usr/share/man/man9f/atomic_swap_uchar.9f target=atomic_swap.9f
619 link path=usr/share/man/man9f/atomic_swap_uint.9f target=atomic_swap.9f
620 link path=usr/share/man/man9f/atomic_swap_ulong.9f target=atomic_swap.9f
621 link path=usr/share/man/man9f/atomic_swap_ushort.9f target=atomic_swap.9f
622 link path=usr/share/man/man9f/avl_add.9f target=avl.9f
623 link path=usr/share/man/man9f/avl_create.9f target=avl.9f
624 link path=usr/share/man/man9f/avl_destroy.9f target=avl.9f
625 link path=usr/share/man/man9f/avl_destroy_nodes.9f target=avl.9f
626 link path=usr/share/man/man9f/avl_find.9f target=avl.9f
627 link path=usr/share/man/man9f/avl_first.9f target=avl.9f
628 link path=usr/share/man/man9f/avl_insert.9f target=avl.9f
629 link path=usr/share/man/man9f/avl_insert_here.9f target=avl.9f
630 link path=usr/share/man/man9f/avl_is_empty.9f target=avl.9f
631 link path=usr/share/man/man9f/avl_last.9f target=avl.9f
632 link path=usr/share/man/man9f/avl_nearest.9f target=avl.9f
633 link path=usr/share/man/man9f/avl_numnodes.9f target=avl.9f
634 link path=usr/share/man/man9f/avl_remove.9f target=avl.9f
635 link path=usr/share/man/man9f/avl_swap.9f target=avl.9f
636 link path=usr/share/man/man9f/bcanputnext.9f target=canputnext.9f
637 link path=usr/share/man/man9f/crgetgid.9f target=ddi_cred.9f
638 link path=usr/share/man/man9f/crgetgroups.9f target=ddi_cred.9f
639 link path=usr/share/man/man9f/crgetngroups.9f target=ddi_cred.9f
640 link path=usr/share/man/man9f/crgetrgid.9f target=ddi_cred.9f
641 link path=usr/share/man/man9f/crgetruid.9f target=ddi_cred.9f
642 link path=usr/share/man/man9f/crgetsgid.9f target=ddi_cred.9f
643 link path=usr/share/man/man9f/crgetsuid.9f target=ddi_cred.9f
644 link path=usr/share/man/man9f/crgetuid.9f target=ddi_cred.9f
645 link path=usr/share/man/man9f/crgetzoneid.9f target=ddi_cred.9f
646 link path=usr/share/man/man9f/csx_Get16.9f target=csx_Get8.9f
647 link path=usr/share/man/man9f/csx_Get32.9f target=csx_Get8.9f
648 link path=usr/share/man/man9f/csx_Get64.9f target=csx_Get8.9f
649 link path=usr/share/man/man9f/csx_GetEventMask.9f target=csx_SetEventMask.9f
650 link path=usr/share/man/man9f/csx_GetNextClient.9f \
651     target=csx_GetFirstClient.9f
652 link path=usr/share/man/man9f/csx_GetNextTuple.9f target=csx_GetFirstTuple.9f
653 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_A.9f \

```

```

654 target=csx_Parse_CISTPL_DEVICE.9f
655 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_OA.9f \
656 target=csx_Parse_CISTPL_DEVICE.9f
657 link path=usr/share/man/man9f/csx_Parse_CISTPL_DEVICE_OC.9f \
658 target=csx_Parse_CISTPL_DEVICE.9f
659 link path=usr/share/man/man9f/csx_Parse_CISTPL_JEDEC_A.9f \
660 target=csx_Parse_CISTPL_JEDEC_C.9f
661 link path=usr/share/man/man9f/csx_Parse_CISTPL_LONGLINK_C.9f \
662 target=csx_Parse_CISTPL_LONGLINK_A.9f
663 link path=usr/share/man/man9f/csx_Put16.9f target=csx_Put8.9f
664 link path=usr/share/man/man9f/csx_Put32.9f target=csx_Put8.9f
665 link path=usr/share/man/man9f/csx_Put64.9f target=csx_Put8.9f
666 link path=usr/share/man/man9f/csx_ReleaseIO.9f target=csx_RequestIO.9f
667 link path=usr/share/man/man9f/csx_ReleaseIRQ.9f target=csx_RequestIRQ.9f
668 link path=usr/share/man/man9f/csx_ReleaseSocketMask.9f \
669 target=csx_RequestSocketMask.9f
670 link path=usr/share/man/man9f/csx_ReleaseWindow.9f target=csx_RequestWindow.9f
671 link path=usr/share/man/man9f/csx_RemoveDeviceNode.9f \
672 target=csx_MakeDeviceNode.9f
673 link path=usr/share/man/man9f/csx_RepGet16.9f target=csx_RepGet8.9f
674 link path=usr/share/man/man9f/csx_RepGet32.9f target=csx_RepGet8.9f
675 link path=usr/share/man/man9f/csx_RepGet64.9f target=csx_RepGet8.9f
676 link path=usr/share/man/man9f/csx_RepPut16.9f target=csx_RepPut8.9f
677 link path=usr/share/man/man9f/csx_RepPut32.9f target=csx_RepPut8.9f
678 link path=usr/share/man/man9f/csx_RepPut64.9f target=csx_RepPut8.9f
679 link path=usr/share/man/man9f/cv_broadcast.9f target=condvar.9f
680 link path=usr/share/man/man9f/cv_destroy.9f target=condvar.9f
681 link path=usr/share/man/man9f/cv_init.9f target=condvar.9f
682 link path=usr/share/man/man9f/cv_reltimedwait.9f target=condvar.9f
683 link path=usr/share/man/man9f/cv_reltimedwait_sig.9f target=condvar.9f
684 link path=usr/share/man/man9f/cv_signal.9f target=condvar.9f
685 link path=usr/share/man/man9f/cv_timedwait.9f target=condvar.9f
686 link path=usr/share/man/man9f/cv_timedwait_sig.9f target=condvar.9f
687 link path=usr/share/man/man9f/cv_wait.9f target=condvar.9f
688 link path=usr/share/man/man9f/cv_wait_sig.9f target=condvar.9f
689 link path=usr/share/man/man9f/ddi_btopr.9f target=ddi_btop.9f
690 link path=usr/share/man/man9f/ddi_cb_unregister.9f target=ddi_cb_register.9f
691 link path=usr/share/man/man9f/ddi_check_dma_handle.9f \
692 target=ddi_check_acc_handle.9f
693 link path=usr/share/man/man9f/ddi_devid_free.9f target=ddi_devid_compare.9f
694 link path=usr/share/man/man9f/ddi_devid_get.9f target=ddi_devid_compare.9f
695 link path=usr/share/man/man9f/ddi_devid_init.9f target=ddi_devid_compare.9f
696 link path=usr/share/man/man9f/ddi_devid_register.9f \
697 target=ddi_devid_compare.9f
698 link path=usr/share/man/man9f/ddi_devid_sizeof.9f target=ddi_devid_compare.9f
699 link path=usr/share/man/man9f/ddi_devid_str_decode.9f \
700 target=ddi_devid_compare.9f
701 link path=usr/share/man/man9f/ddi_devid_str_encode.9f \
702 target=ddi_devid_compare.9f
703 link path=usr/share/man/man9f/ddi_devid_str_free.9f \
704 target=ddi_devid_compare.9f
705 link path=usr/share/man/man9f/ddi_devid_unregister.9f \
706 target=ddi_devid_compare.9f
707 link path=usr/share/man/man9f/ddi_devid_valid.9f target=ddi_devid_compare.9f
708 link path=usr/share/man/man9f/ddi_devmap_segmap.9f target=devmap_setup.9f
709 link path=usr/share/man/man9f/ddi_dmae_1stparty.9f target=ddi_dmae.9f
710 link path=usr/share/man/man9f/ddi_dmae_alloc.9f target=ddi_dmae.9f
711 link path=usr/share/man/man9f/ddi_dmae_disable.9f target=ddi_dmae.9f
712 link path=usr/share/man/man9f/ddi_dmae_enable.9f target=ddi_dmae.9f
713 link path=usr/share/man/man9f/ddi_dmae_getattr.9f target=ddi_dmae.9f
714 link path=usr/share/man/man9f/ddi_dmae_getcnt.9f target=ddi_dmae.9f
715 link path=usr/share/man/man9f/ddi_dmae_prog.9f target=ddi_dmae.9f
716 link path=usr/share/man/man9f/ddi_dmae_release.9f target=ddi_dmae.9f
717 link path=usr/share/man/man9f/ddi_dmae_stop.9f target=ddi_dmae.9f
718 link path=usr/share/man/man9f/ddi_exit_critical.9f \
719 target=ddi_enter_critical.9f

```

```

720 link path=usr/share/man/man9f/ddi_fls.9f target=ddi_ffs.9f
721 link path=usr/share/man/man9f/ddi_fm_capable.9f target=ddi_fm_init.9f
722 link path=usr/share/man/man9f/ddi_fm_dma_err_clear.9f \
723 target=ddi_fm_acc_err_clear.9f
724 link path=usr/share/man/man9f/ddi_fm_dma_err_get.9f \
725 target=ddi_fm_acc_err_get.9f
726 link path=usr/share/man/man9f/ddi_fm_fini.9f target=ddi_fm_init.9f
727 link path=usr/share/man/man9f/ddi_fm_handler_unregister.9f \
728 target=ddi_fm_handler_register.9f
729 link path=usr/share/man/man9f/ddi_get16.9f target=ddi_get8.9f
730 link path=usr/share/man/man9f/ddi_get32.9f target=ddi_get8.9f
731 link path=usr/share/man/man9f/ddi_get64.9f target=ddi_get8.9f
732 link path=usr/share/man/man9f/ddi_get_iblock_cookie.9f target=ddi_add_intr.9f
733 link path=usr/share/man/man9f/ddi_get_lbolt64.9f target=ddi_get_lbolt.9f
734 link path=usr/share/man/man9f/ddi_get_name.9f target=ddi_binding_name.9f
735 link path=usr/share/man/man9f/ddi_get_soft_iblock_cookie.9f \
736 target=ddi_add_softintr.9f
737 link path=usr/share/man/man9f/ddi_get_soft_state.9f target=ddi_soft_state.9f
738 link path=usr/share/man/man9f/ddi_getb.9f target=ddi_get8.9f
739 link path=usr/share/man/man9f/ddi_getl.9f target=ddi_get8.9f
740 link path=usr/share/man/man9f/ddi_getll.9f target=ddi_get8.9f
741 link path=usr/share/man/man9f/ddi_getlongprop.9f target=ddi_prop_op.9f
742 link path=usr/share/man/man9f/ddi_getlongprop_buf.9f target=ddi_prop_op.9f
743 link path=usr/share/man/man9f/ddi_getprop.9f target=ddi_prop_op.9f
744 link path=usr/share/man/man9f/ddi_getpropflen.9f target=ddi_prop_op.9f
745 link path=usr/share/man/man9f/ddi_getw.9f target=ddi_get8.9f
746 link path=usr/share/man/man9f/ddi_intr_block_disable.9f \
747 target=ddi_intr_enable.9f
748 link path=usr/share/man/man9f/ddi_intr_block_enable.9f \
749 target=ddi_intr_enable.9f
750 link path=usr/share/man/man9f/ddi_intr_clr_mask.9f target=ddi_intr_set_mask.9f
751 link path=usr/share/man/man9f/ddi_intr_disable.9f target=ddi_intr_enable.9f
752 link path=usr/share/man/man9f/ddi_intr_free.9f target=ddi_intr_alloc.9f
753 link path=usr/share/man/man9f/ddi_intr_get_navail.9f \
754 target=ddi_intr_get_nintrs.9f
755 link path=usr/share/man/man9f/ddi_intr_get_softint_pri.9f \
756 target=ddi_intr_add_softint.9f
757 link path=usr/share/man/man9f/ddi_intr_remove_handler.9f \
758 target=ddi_intr_add_handler.9f
759 link path=usr/share/man/man9f/ddi_intr_remove_softint.9f \
760 target=ddi_intr_add_softint.9f
761 link path=usr/share/man/man9f/ddi_intr_set_cap.9f target=ddi_intr_get_cap.9f
762 link path=usr/share/man/man9f/ddi_intr_set_pri.9f target=ddi_intr_get_pri.9f
763 link path=usr/share/man/man9f/ddi_intr_set_softint_pri.9f \
764 target=ddi_intr_add_softint.9f
765 link path=usr/share/man/man9f/ddi_intr_trigger_softint.9f \
766 target=ddi_intr_add_softint.9f
767 link path=usr/share/man/man9f/ddi_io_get16.9f target=ddi_io_get8.9f
768 link path=usr/share/man/man9f/ddi_io_get32.9f target=ddi_io_get8.9f
769 link path=usr/share/man/man9f/ddi_io_getb.9f target=ddi_io_get8.9f
770 link path=usr/share/man/man9f/ddi_io_getl.9f target=ddi_io_get8.9f
771 link path=usr/share/man/man9f/ddi_io_getw.9f target=ddi_io_get8.9f
772 link path=usr/share/man/man9f/ddi_io_put16.9f target=ddi_io_put8.9f
773 link path=usr/share/man/man9f/ddi_io_put32.9f target=ddi_io_put8.9f
774 link path=usr/share/man/man9f/ddi_io_putb.9f target=ddi_io_put8.9f
775 link path=usr/share/man/man9f/ddi_io_putl.9f target=ddi_io_put8.9f
776 link path=usr/share/man/man9f/ddi_io_putw.9f target=ddi_io_put8.9f
777 link path=usr/share/man/man9f/ddi_io_rep_get16.9f target=ddi_io_rep_get8.9f
778 link path=usr/share/man/man9f/ddi_io_rep_get32.9f target=ddi_io_rep_get8.9f
779 link path=usr/share/man/man9f/ddi_io_rep_getb.9f target=ddi_io_rep_get8.9f
780 link path=usr/share/man/man9f/ddi_io_rep_getl.9f target=ddi_io_rep_get8.9f
781 link path=usr/share/man/man9f/ddi_io_rep_getw.9f target=ddi_io_rep_get8.9f
782 link path=usr/share/man/man9f/ddi_io_rep_put16.9f target=ddi_io_rep_put8.9f
783 link path=usr/share/man/man9f/ddi_io_rep_put32.9f target=ddi_io_rep_put8.9f
784 link path=usr/share/man/man9f/ddi_io_rep_putb.9f target=ddi_io_rep_put8.9f
785 link path=usr/share/man/man9f/ddi_io_rep_putl.9f target=ddi_io_rep_put8.9f

```

```

786 link path=usr/share/man/man9f/ddi_io_rep_putw.9f target=ddi_io_rep_put8.9f
787 link path=usr/share/man/man9f/ddi_mem_get16.9f target=ddi_mem_get8.9f
788 link path=usr/share/man/man9f/ddi_mem_get32.9f target=ddi_mem_get8.9f
789 link path=usr/share/man/man9f/ddi_mem_get64.9f target=ddi_mem_get8.9f
790 link path=usr/share/man/man9f/ddi_mem_getb.9f target=ddi_mem_get8.9f
791 link path=usr/share/man/man9f/ddi_mem_getl.9f target=ddi_mem_get8.9f
792 link path=usr/share/man/man9f/ddi_mem_getll.9f target=ddi_mem_get8.9f
793 link path=usr/share/man/man9f/ddi_mem_getw.9f target=ddi_mem_get8.9f
794 link path=usr/share/man/man9f/ddi_mem_put16.9f target=ddi_mem_put8.9f
795 link path=usr/share/man/man9f/ddi_mem_put32.9f target=ddi_mem_put8.9f
796 link path=usr/share/man/man9f/ddi_mem_put64.9f target=ddi_mem_put8.9f
797 link path=usr/share/man/man9f/ddi_mem_putb.9f target=ddi_mem_put8.9f
798 link path=usr/share/man/man9f/ddi_mem_putl.9f target=ddi_mem_put8.9f
799 link path=usr/share/man/man9f/ddi_mem_putll.9f target=ddi_mem_put8.9f
800 link path=usr/share/man/man9f/ddi_mem_putw.9f target=ddi_mem_put8.9f
801 link path=usr/share/man/man9f/ddi_mem_rep_get16.9f target=ddi_mem_rep_get8.9f
802 link path=usr/share/man/man9f/ddi_mem_rep_get32.9f target=ddi_mem_rep_get8.9f
803 link path=usr/share/man/man9f/ddi_mem_rep_get64.9f target=ddi_mem_rep_get8.9f
804 link path=usr/share/man/man9f/ddi_mem_rep_getb.9f target=ddi_mem_rep_get8.9f
805 link path=usr/share/man/man9f/ddi_mem_rep_getl.9f target=ddi_mem_rep_get8.9f
806 link path=usr/share/man/man9f/ddi_mem_rep_getll.9f target=ddi_mem_rep_get8.9f
807 link path=usr/share/man/man9f/ddi_mem_rep_getw.9f target=ddi_mem_rep_get8.9f
808 link path=usr/share/man/man9f/ddi_mem_rep_put16.9f target=ddi_mem_rep_put8.9f
809 link path=usr/share/man/man9f/ddi_mem_rep_put32.9f target=ddi_mem_rep_put8.9f
810 link path=usr/share/man/man9f/ddi_mem_rep_put64.9f target=ddi_mem_rep_put8.9f
811 link path=usr/share/man/man9f/ddi_mem_rep_putb.9f target=ddi_mem_rep_put8.9f
812 link path=usr/share/man/man9f/ddi_mem_rep_putl.9f target=ddi_mem_rep_put8.9f
813 link path=usr/share/man/man9f/ddi_mem_rep_putll.9f target=ddi_mem_rep_put8.9f
814 link path=usr/share/man/man9f/ddi_mem_rep_putw.9f target=ddi_mem_rep_put8.9f
815 link path=usr/share/man/man9f/ddi_modclose.9f target=ddi_modopen.9f
816 link path=usr/share/man/man9f/ddi_modsym.9f target=ddi_modopen.9f
817 link path=usr/share/man/man9f/ddi_peek16.9f target=ddi_peek.9f
818 link path=usr/share/man/man9f/ddi_peek32.9f target=ddi_peek.9f
819 link path=usr/share/man/man9f/ddi_peek64.9f target=ddi_peek.9f
820 link path=usr/share/man/man9f/ddi_peek8.9f target=ddi_peek.9f
821 link path=usr/share/man/man9f/ddi_peekc.9f target=ddi_peek.9f
822 link path=usr/share/man/man9f/ddi_peekd.9f target=ddi_peek.9f
823 link path=usr/share/man/man9f/ddi_peekl.9f target=ddi_peek.9f
824 link path=usr/share/man/man9f/ddi_peeks.9f target=ddi_peek.9f
825 link path=usr/share/man/man9f/ddi_poke16.9f target=ddi_poke.9f
826 link path=usr/share/man/man9f/ddi_poke32.9f target=ddi_poke.9f
827 link path=usr/share/man/man9f/ddi_poke64.9f target=ddi_poke.9f
828 link path=usr/share/man/man9f/ddi_poke8.9f target=ddi_poke.9f
829 link path=usr/share/man/man9f/ddi_pokec.9f target=ddi_poke.9f
830 link path=usr/share/man/man9f/ddi_poked.9f target=ddi_poke.9f
831 link path=usr/share/man/man9f/ddi_pokel.9f target=ddi_poke.9f
832 link path=usr/share/man/man9f/ddi_pokes.9f target=ddi_poke.9f
833 link path=usr/share/man/man9f/ddi_prop_free.9f target=ddi_prop_lookup.9f
834 link path=usr/share/man/man9f/ddi_prop_get_int64.9f target=ddi_prop_get_int.9f
835 link path=usr/share/man/man9f/ddi_prop_lookup_byte_array.9f \
836 target=ddi_prop_lookup.9f
837 link path=usr/share/man/man9f/ddi_prop_lookup_int64_array.9f \
838 target=ddi_prop_lookup.9f
839 link path=usr/share/man/man9f/ddi_prop_lookup_int_array.9f \
840 target=ddi_prop_lookup.9f
841 link path=usr/share/man/man9f/ddi_prop_lookup_string.9f \
842 target=ddi_prop_lookup.9f
843 link path=usr/share/man/man9f/ddi_prop_lookup_string_array.9f \
844 target=ddi_prop_lookup.9f
845 link path=usr/share/man/man9f/ddi_prop_modify.9f target=ddi_prop_create.9f
846 link path=usr/share/man/man9f/ddi_prop_remove.9f target=ddi_prop_create.9f
847 link path=usr/share/man/man9f/ddi_prop_remove_all.9f target=ddi_prop_create.9f
848 link path=usr/share/man/man9f/ddi_prop_undefine.9f target=ddi_prop_create.9f
849 link path=usr/share/man/man9f/ddi_prop_update_byte_array.9f \
850 target=ddi_prop_update.9f
851 link path=usr/share/man/man9f/ddi_prop_update_int.9f target=ddi_prop_update.9f

```

```

852 link path=usr/share/man/man9f/ddi_prop_update_int64.9f \
853 target=ddi_prop_update.9f
854 link path=usr/share/man/man9f/ddi_prop_update_int64_array.9f \
855 target=ddi_prop_update.9f
856 link path=usr/share/man/man9f/ddi_prop_update_int_array.9f \
857 target=ddi_prop_update.9f
858 link path=usr/share/man/man9f/ddi_prop_update_string.9f \
859 target=ddi_prop_update.9f
860 link path=usr/share/man/man9f/ddi_prop_update_string_array.9f \
861 target=ddi_prop_update.9f
862 link path=usr/share/man/man9f/ddi_ptob.9f target=ddi_btop.9f
863 link path=usr/share/man/man9f/ddi_put16.9f target=ddi_put8.9f
864 link path=usr/share/man/man9f/ddi_put32.9f target=ddi_put8.9f
865 link path=usr/share/man/man9f/ddi_put64.9f target=ddi_put8.9f
866 link path=usr/share/man/man9f/ddi_putb.9f target=ddi_put8.9f
867 link path=usr/share/man/man9f/ddi_putl.9f target=ddi_put8.9f
868 link path=usr/share/man/man9f/ddi_putll.9f target=ddi_put8.9f
869 link path=usr/share/man/man9f/ddi_putw.9f target=ddi_put8.9f
870 link path=usr/share/man/man9f/ddi_remove_intr.9f target=ddi_add_intr.9f
871 link path=usr/share/man/man9f/ddi_remove_softintr.9f \
872 target=ddi_add_softintr.9f
873 link path=usr/share/man/man9f/ddi_rep_get16.9f target=ddi_rep_get8.9f
874 link path=usr/share/man/man9f/ddi_rep_get32.9f target=ddi_rep_get8.9f
875 link path=usr/share/man/man9f/ddi_rep_get64.9f target=ddi_rep_get8.9f
876 link path=usr/share/man/man9f/ddi_rep_getb.9f target=ddi_rep_get8.9f
877 link path=usr/share/man/man9f/ddi_rep_getl.9f target=ddi_rep_get8.9f
878 link path=usr/share/man/man9f/ddi_rep_getll.9f target=ddi_rep_get8.9f
879 link path=usr/share/man/man9f/ddi_rep_getw.9f target=ddi_rep_get8.9f
880 link path=usr/share/man/man9f/ddi_rep_put16.9f target=ddi_rep_put8.9f
881 link path=usr/share/man/man9f/ddi_rep_put32.9f target=ddi_rep_put8.9f
882 link path=usr/share/man/man9f/ddi_rep_put64.9f target=ddi_rep_put8.9f
883 link path=usr/share/man/man9f/ddi_rep_putb.9f target=ddi_rep_put8.9f
884 link path=usr/share/man/man9f/ddi_rep_putl.9f target=ddi_rep_put8.9f
885 link path=usr/share/man/man9f/ddi_rep_putll.9f target=ddi_rep_put8.9f
886 link path=usr/share/man/man9f/ddi_rep_putw.9f target=ddi_rep_put8.9f
887 link path=usr/share/man/man9f/ddi_segmap_setup.9f target=ddi_segmap.9f
888 link path=usr/share/man/man9f/ddi_set_driver_private.9f \
889 target=ddi_get_driver_private.9f
890 link path=usr/share/man/man9f/ddi_soft_state_fini.9f target=ddi_soft_state.9f
891 link path=usr/share/man/man9f/ddi_soft_state_free.9f target=ddi_soft_state.9f
892 link path=usr/share/man/man9f/ddi_soft_state_init.9f target=ddi_soft_state.9f
893 link path=usr/share/man/man9f/ddi_soft_state_zalloc.9f \
894 target=ddi_soft_state.9f
895 link path=usr/share/man/man9f/ddi_strdup.9f target=string.9f
896 link path=usr/share/man/man9f/ddi_strtoll.9f target=ddi_strtoll.9f
897 link path=usr/share/man/man9f/ddi_taskq_create.9f target=taskq.9f
898 link path=usr/share/man/man9f/ddi_taskq_destroy.9f target=taskq.9f
899 link path=usr/share/man/man9f/ddi_taskq_dispatch.9f target=taskq.9f
900 link path=usr/share/man/man9f/ddi_taskq_resume.9f target=taskq.9f
901 link path=usr/share/man/man9f/ddi_taskq_suspend.9f target=taskq.9f
902 link path=usr/share/man/man9f/ddi_taskq_wait.9f target=taskq.9f
903 link path=usr/share/man/man9f/ddi_trigger_softintr.9f \
904 target=ddi_add_softintr.9f
905 link path=usr/share/man/man9f/ddi_umem_free.9f target=ddi_umem_alloc.9f
906 link path=usr/share/man/man9f/ddi_umem_unlock.9f target=ddi_umem_lock.9f
907 link path=usr/share/man/man9f/ddi_unmap_regs.9f target=ddi_map_regs.9f
908 link path=usr/share/man/man9f/desballoc.9f target=esballoc.9f
909 link path=usr/share/man/man9f/dev_err.9f target=cmm_err.9f
910 link path=usr/share/man/man9f/devmap_load.9f target=devmap_unload.9f
911 link path=usr/share/man/man9f/devmap_umem_setup.9f \
912 target=devmap_devmem_setup.9f
913 link path=usr/share/man/man9f/dlerrorack.9f target=dlbindack.9f
914 link path=usr/share/man/man9f/dlokack.9f target=dlbindack.9f
915 link path=usr/share/man/man9f/dlphysaddrack.9f target=dlbindack.9f
916 link path=usr/share/man/man9f/dluderrorind.9f target=dlbindack.9f
917 link path=usr/share/man/man9f/firmware_close.9f target=firmlod.9f

```

```
918 link path=usr/share/man/man9f/firmware_free.9f target=firmlload.9f
919 link path=usr/share/man/man9f/firmware_get_size.9f target=firmlload.9f
920 link path=usr/share/man/man9f/firmware_malloc.9f target=firmlload.9f
921 link path=usr/share/man/man9f/firmware_open.9f target=firmlload.9f
922 link path=usr/share/man/man9f/firmware_read.9f target=firmlload.9f
923 link path=usr/share/man/man9f/free_pktiopb.9f target=get_pktiopb.9f
924 link path=usr/share/man/man9f/gld_intr.9f target=gld.9f
925 link path=usr/share/man/man9f/gld_mac_alloc.9f target=gld.9f
926 link path=usr/share/man/man9f/gld_mac_free.9f target=gld.9f
927 link path=usr/share/man/man9f/gld_rcv.9f target=gld.9f
928 link path=usr/share/man/man9f/gld_register.9f target=gld.9f
929 link path=usr/share/man/man9f/gld_sched.9f target=gld.9f
930 link path=usr/share/man/man9f/gld_unregister.9f target=gld.9f
931 link path=usr/share/man/man9f/id32_free.9f target=id32_alloc.9f
932 link path=usr/share/man/man9f/id32_lookup.9f target=id32_alloc.9f
933 link path=usr/share/man/man9f/id_alloc.9f target=id_space.9f
934 link path=usr/share/man/man9f/id_alloc_nosleep.9f target=id_space.9f
935 link path=usr/share/man/man9f/id_alloc_specific_nosleep.9f target=id_space.9f
936 link path=usr/share/man/man9f/id_alloff.9f target=id_space.9f
937 link path=usr/share/man/man9f/id_alloff_nosleep.9f target=id_space.9f
938 link path=usr/share/man/man9f/id_free.9f target=id_space.9f
939 link path=usr/share/man/man9f/id_space_create.9f target=id_space.9f
940 link path=usr/share/man/man9f/id_space_destroy.9f target=id_space.9f
941 link path=usr/share/man/man9f/id_space_extend.9f target=id_space.9f
942 link path=usr/share/man/man9f/inl.9f target=inb.9f
943 link path=usr/share/man/man9f/intro.9f target=intro.9f
944 link path=usr/share/man/man9f/inw.9f target=inb.9f
945 link path=usr/share/man/man9f/kmem_cache_alloc.9f target=kmem_cache_create.9f
946 link path=usr/share/man/man9f/kmem_cache_destroy.9f \
947 target=kmem_cache_create.9f
948 link path=usr/share/man/man9f/kmem_cache_free.9f target=kmem_cache_create.9f
949 link path=usr/share/man/man9f/kmem_cache_set_move.9f \
950 target=kmem_cache_create.9f
951 link path=usr/share/man/man9f/kmem_free.9f target=kmem_alloc.9f
952 link path=usr/share/man/man9f/kmem_zalloc.9f target=kmem_alloc.9f
953 link path=usr/share/man/man9f/kstat_named_setstr.9f target=kstat_named_init.9f
954 link path=usr/share/man/man9f/kstat_runq_back_to_waitq.9f \
955 target=kstat_queue.9f
956 link path=usr/share/man/man9f/kstat_runq_enter.9f target=kstat_queue.9f
957 link path=usr/share/man/man9f/kstat_runq_exit.9f target=kstat_queue.9f
958 link path=usr/share/man/man9f/kstat_waitq_enter.9f target=kstat_queue.9f
959 link path=usr/share/man/man9f/kstat_waitq_exit.9f target=kstat_queue.9f
960 link path=usr/share/man/man9f/kstat_waitq_to_runq.9f target=kstat_queue.9f
961 link path=usr/share/man/man9f/ldi_awrite.9f target=ldi_aread.9f
962 link path=usr/share/man/man9f/ldi_close.9f target=ldi_open_by_dev.9f
963 link path=usr/share/man/man9f/ldi_get_devid.9f target=ldi_get_dev.9f
964 link path=usr/share/man/man9f/ldi_get_minor_name.9f target=ldi_get_dev.9f
965 link path=usr/share/man/man9f/ldi_get_otyp.9f target=ldi_get_dev.9f
966 link path=usr/share/man/man9f/ldi_getmsg.9f target=ldi_putmsg.9f
967 link path=usr/share/man/man9f/ldi_ident_from_dip.9f \
968 target=ldi_ident_from_dev.9f
969 link path=usr/share/man/man9f/ldi_ident_from_stream.9f \
970 target=ldi_ident_from_dev.9f
971 link path=usr/share/man/man9f/ldi_ident_release.9f \
972 target=ldi_ident_from_dev.9f
973 link path=usr/share/man/man9f/ldi_open_by_devid.9f target=ldi_open_by_dev.9f
974 link path=usr/share/man/man9f/ldi_open_by_name.9f target=ldi_open_by_dev.9f
975 link path=usr/share/man/man9f/ldi_prop_get_int64.9f target=ldi_prop_get_int.9f
976 link path=usr/share/man/man9f/ldi_prop_lookup_byte_array.9f \
977 target=ldi_prop_lookup_int_array.9f
978 link path=usr/share/man/man9f/ldi_prop_lookup_int64_array.9f \
979 target=ldi_prop_lookup_int_array.9f
980 link path=usr/share/man/man9f/ldi_prop_lookup_string.9f \
981 target=ldi_prop_lookup_int_array.9f
982 link path=usr/share/man/man9f/ldi_prop_lookup_string_array.9f \
983 target=ldi_prop_lookup_int_array.9f
```

```
984 link path=usr/share/man/man9f/ldi_write.9f target=ldi_read.9f
985 link path=usr/share/man/man9f/list_destroy.9f target=list_create.9f
986 link path=usr/share/man/man9f/list_head.9f target=list_create.9f
987 link path=usr/share/man/man9f/list_insert_after.9f target=list_create.9f
988 link path=usr/share/man/man9f/list_insert_before.9f target=list_create.9f
989 link path=usr/share/man/man9f/list_insert_head.9f target=list_create.9f
990 link path=usr/share/man/man9f/list_insert_tail.9f target=list_create.9f
991 link path=usr/share/man/man9f/list_is_empty.9f target=list_create.9f
992 link path=usr/share/man/man9f/list_link_active.9f target=list_create.9f
993 link path=usr/share/man/man9f/list_link_init.9f target=list_create.9f
994 link path=usr/share/man/man9f/list_link_replace.9f target=list_create.9f
995 link path=usr/share/man/man9f/list_move_tail.9f target=list_create.9f
996 link path=usr/share/man/man9f/list_next.9f target=list_create.9f
997 link path=usr/share/man/man9f/list_prev.9f target=list_create.9f
998 link path=usr/share/man/man9f/list_remove.9f target=list_create.9f
999 link path=usr/share/man/man9f/list_remove_head.9f target=list_create.9f
1000 link path=usr/share/man/man9f/list_remove_tail.9f target=list_create.9f
1001 link path=usr/share/man/man9f/list_tail.9f target=list_create.9f
1002 link path=usr/share/man/man9f/mac_fini_ops.9f target=mac_init_ops.9f
1003 link path=usr/share/man/man9f/mac_free.9f target=mac_alloc.9f
1004 link path=usr/share/man/man9f/mac_hcksum_set.9f target=mac_hcksum_get.9f
1005 link path=usr/share/man/man9f/mac_prop_info_set_default_link_flowctrl.9f \
1006 target=mac_prop_info.9f
1007 link path=usr/share/man/man9f/mac_prop_info_set_default_str.9f \
1008 target=mac_prop_info.9f
1009 link path=usr/share/man/man9f/mac_prop_info_set_default_uint32.9f \
1010 target=mac_prop_info.9f
1011 link path=usr/share/man/man9f/mac_prop_info_set_default_uint64.9f \
1012 target=mac_prop_info.9f
1013 link path=usr/share/man/man9f/mac_prop_info_set_default_uint8.9f \
1014 target=mac_prop_info.9f
1015 link path=usr/share/man/man9f/mac_prop_info_set_perm.9f \
1016 target=mac_prop_info.9f
1017 link path=usr/share/man/man9f/mac_prop_info_set_range_uint32.9f \
1018 target=mac_prop_info.9f
1019 link path=usr/share/man/man9f/mac_unregister.9f target=mac_register.9f
1020 link path=usr/share/man/man9f/makecom_g0.9f target=makecom.9f
1021 link path=usr/share/man/man9f/makecom_g0_s.9f target=makecom.9f
1022 link path=usr/share/man/man9f/makecom_g1.9f target=makecom.9f
1023 link path=usr/share/man/man9f/makecom_g5.9f target=makecom.9f
1024 link path=usr/share/man/man9f/membar_consumer.9f target=membar_ops.9f
1025 link path=usr/share/man/man9f/membar_enter.9f target=membar_ops.9f
1026 link path=usr/share/man/man9f/membar_exit.9f target=membar_ops.9f
1027 link path=usr/share/man/man9f/membar_producer.9f target=membar_ops.9f
1028 link path=usr/share/man/man9f/memcmp.9f target=memchr.9f
1029 link path=usr/share/man/man9f/memcpy.9f target=memchr.9f
1030 link path=usr/share/man/man9f/memmove.9f target=memchr.9f
1031 link path=usr/share/man/man9f/memset.9f target=memchr.9f
1032 link path=usr/share/man/man9f/minphys.9f target=physio.9f
1033 link path=usr/share/man/man9f/mod_info.9f target=mod_install.9f
1034 link path=usr/share/man/man9f/mod_modname.9f target=mod_install.9f
1035 link path=usr/share/man/man9f/mod_remove.9f target=mod_install.9f
1036 link path=usr/share/man/man9f/mutex_destroy.9f target=mutex.9f
1037 link path=usr/share/man/man9f/mutex_enter.9f target=mutex.9f
1038 link path=usr/share/man/man9f/mutex_exit.9f target=mutex.9f
1039 link path=usr/share/man/man9f/mutex_init.9f target=mutex.9f
1040 link path=usr/share/man/man9f/mutex_owed.9f target=mutex.9f
1041 link path=usr/share/man/man9f/mutex_tryenter.9f target=mutex.9f
1042 link path=usr/share/man/man9f/net_event_notify_unregister.9f \
1043 target=net_event_notify_register.9f
1044 link path=usr/share/man/man9f/net_instance_notify_unregister.9f \
1045 target=net_instance_notify_register.9f
1046 link path=usr/share/man/man9f/net_instance_protocol_unregister.9f \
1047 target=net_protocol_notify_register.9f
1048 link path=usr/share/man/man9f/numtos.9f target=stoi.9f
1049 link path=usr/share/man/man9f/nv_alloc_fini.9f target=nvlist_alloc.9f
```

```
1050 link path=usr/share/man/man9f/nv_alloc_init.9f target=nvlist_alloc.9f
1051 link path=usr/share/man/man9f/nvlist_add_boolean_array.9f \
1052 target=nvlist_add_boolean.9f
1053 link path=usr/share/man/man9f/nvlist_add_boolean_value.9f \
1054 target=nvlist_add_boolean.9f
1055 link path=usr/share/man/man9f/nvlist_add_byte.9f target=nvlist_add_boolean.9f
1056 link path=usr/share/man/man9f/nvlist_add_byte_array.9f \
1057 target=nvlist_add_boolean.9f
1058 link path=usr/share/man/man9f/nvlist_add_int16.9f target=nvlist_add_boolean.9f
1059 link path=usr/share/man/man9f/nvlist_add_int16_array.9f \
1060 target=nvlist_add_boolean.9f
1061 link path=usr/share/man/man9f/nvlist_add_int32.9f target=nvlist_add_boolean.9f
1062 link path=usr/share/man/man9f/nvlist_add_int32_array.9f \
1063 target=nvlist_add_boolean.9f
1064 link path=usr/share/man/man9f/nvlist_add_int64.9f target=nvlist_add_boolean.9f
1065 link path=usr/share/man/man9f/nvlist_add_int64_array.9f \
1066 target=nvlist_add_boolean.9f
1067 link path=usr/share/man/man9f/nvlist_add_int8.9f target=nvlist_add_boolean.9f
1068 link path=usr/share/man/man9f/nvlist_add_int8_array.9f \
1069 target=nvlist_add_boolean.9f
1070 link path=usr/share/man/man9f/nvlist_add_nvlist.9f \
1071 target=nvlist_add_boolean.9f
1072 link path=usr/share/man/man9f/nvlist_add_nvlist_array.9f \
1073 target=nvlist_add_boolean.9f
1074 link path=usr/share/man/man9f/nvlist_add_nvpair.9f \
1075 target=nvlist_add_boolean.9f
1076 link path=usr/share/man/man9f/nvlist_add_string.9f \
1077 target=nvlist_add_boolean.9f
1078 link path=usr/share/man/man9f/nvlist_add_string_array.9f \
1079 target=nvlist_add_boolean.9f
1080 link path=usr/share/man/man9f/nvlist_add_uint16.9f \
1081 target=nvlist_add_boolean.9f
1082 link path=usr/share/man/man9f/nvlist_add_uint16_array.9f \
1083 target=nvlist_add_boolean.9f
1084 link path=usr/share/man/man9f/nvlist_add_uint32.9f \
1085 target=nvlist_add_boolean.9f
1086 link path=usr/share/man/man9f/nvlist_add_uint32_array.9f \
1087 target=nvlist_add_boolean.9f
1088 link path=usr/share/man/man9f/nvlist_add_uint64.9f \
1089 target=nvlist_add_boolean.9f
1090 link path=usr/share/man/man9f/nvlist_add_uint64_array.9f \
1091 target=nvlist_add_boolean.9f
1092 link path=usr/share/man/man9f/nvlist_add_uint8.9f target=nvlist_add_boolean.9f
1093 link path=usr/share/man/man9f/nvlist_add_uint8_array.9f \
1094 target=nvlist_add_boolean.9f
1095 link path=usr/share/man/man9f/nvlist_dup.9f target=nvlist_alloc.9f
1096 link path=usr/share/man/man9f/nvlist_exists.9f target=nvlist_lookup_nvpair.9f
1097 link path=usr/share/man/man9f/nvlist_free.9f target=nvlist_alloc.9f
1098 link path=usr/share/man/man9f/nvlist_lookup_boolean_array.9f \
1099 target=nvlist_lookup_boolean.9f
1100 link path=usr/share/man/man9f/nvlist_lookup_boolean_value.9f \
1101 target=nvlist_lookup_boolean.9f
1102 link path=usr/share/man/man9f/nvlist_lookup_byte.9f \
1103 target=nvlist_lookup_boolean.9f
1104 link path=usr/share/man/man9f/nvlist_lookup_byte_array.9f \
1105 target=nvlist_lookup_boolean.9f
1106 link path=usr/share/man/man9f/nvlist_lookup_int16.9f \
1107 target=nvlist_lookup_boolean.9f
1108 link path=usr/share/man/man9f/nvlist_lookup_int16_array.9f \
1109 target=nvlist_lookup_boolean.9f
1110 link path=usr/share/man/man9f/nvlist_lookup_int32.9f \
1111 target=nvlist_lookup_boolean.9f
1112 link path=usr/share/man/man9f/nvlist_lookup_int32_array.9f \
1113 target=nvlist_lookup_boolean.9f
1114 link path=usr/share/man/man9f/nvlist_lookup_int64.9f \
1115 target=nvlist_lookup_boolean.9f
```

```
1116 link path=usr/share/man/man9f/nvlist_lookup_int64_array.9f \
1117 target=nvlist_lookup_boolean.9f
1118 link path=usr/share/man/man9f/nvlist_lookup_int8.9f \
1119 target=nvlist_lookup_boolean.9f
1120 link path=usr/share/man/man9f/nvlist_lookup_int8_array.9f \
1121 target=nvlist_lookup_boolean.9f
1122 link path=usr/share/man/man9f/nvlist_lookup_nvlist.9f \
1123 target=nvlist_lookup_boolean.9f
1124 link path=usr/share/man/man9f/nvlist_lookup_nvlist_array.9f \
1125 target=nvlist_lookup_boolean.9f
1126 link path=usr/share/man/man9f/nvlist_lookup_pairs.9f \
1127 target=nvlist_lookup_boolean.9f
1128 link path=usr/share/man/man9f/nvlist_lookup_string.9f \
1129 target=nvlist_lookup_boolean.9f
1130 link path=usr/share/man/man9f/nvlist_lookup_string_array.9f \
1131 target=nvlist_lookup_boolean.9f
1132 link path=usr/share/man/man9f/nvlist_lookup_uint16.9f \
1133 target=nvlist_lookup_boolean.9f
1134 link path=usr/share/man/man9f/nvlist_lookup_uint16_array.9f \
1135 target=nvlist_lookup_boolean.9f
1136 link path=usr/share/man/man9f/nvlist_lookup_uint32.9f \
1137 target=nvlist_lookup_boolean.9f
1138 link path=usr/share/man/man9f/nvlist_lookup_uint32_array.9f \
1139 target=nvlist_lookup_boolean.9f
1140 link path=usr/share/man/man9f/nvlist_lookup_uint64.9f \
1141 target=nvlist_lookup_boolean.9f
1142 link path=usr/share/man/man9f/nvlist_lookup_uint64_array.9f \
1143 target=nvlist_lookup_boolean.9f
1144 link path=usr/share/man/man9f/nvlist_lookup_uint8.9f \
1145 target=nvlist_lookup_boolean.9f
1146 link path=usr/share/man/man9f/nvlist_lookup_uint8_array.9f \
1147 target=nvlist_lookup_boolean.9f
1148 link path=usr/share/man/man9f/nvlist_merge.9f target=nvlist_alloc.9f
1149 link path=usr/share/man/man9f/nvlist_pack.9f target=nvlist_alloc.9f
1150 link path=usr/share/man/man9f/nvlist_remove.9f target=nvlist_remove.9f
1151 link path=usr/share/man/man9f/nvlist_size.9f target=nvlist_alloc.9f
1152 link path=usr/share/man/man9f/nvlist_t.9f target=nvlist_add_boolean.9f
1153 link path=usr/share/man/man9f/nvlist_unpack.9f target=nvlist_alloc.9f
1154 link path=usr/share/man/man9f/nvlist_xalloc.9f target=nvlist_alloc.9f
1155 link path=usr/share/man/man9f/nvlist_xdup.9f target=nvlist_alloc.9f
1156 link path=usr/share/man/man9f/nvlist_xpack.9f target=nvlist_alloc.9f
1157 link path=usr/share/man/man9f/nvlist_xunpack.9f target=nvlist_alloc.9f
1158 link path=usr/share/man/man9f/nvpair_name.9f target=nvlist_next_nvpair.9f
1159 link path=usr/share/man/man9f/nvpair_type.9f target=nvlist_next_nvpair.9f
1160 link path=usr/share/man/man9f/nvpair_value_boolean_array.9f \
1161 target=nvpair_value_byte.9f
1162 link path=usr/share/man/man9f/nvpair_value_byte_array.9f \
1163 target=nvpair_value_byte.9f
1164 link path=usr/share/man/man9f/nvpair_value_int16.9f \
1165 target=nvpair_value_byte.9f
1166 link path=usr/share/man/man9f/nvpair_value_int16_array.9f \
1167 target=nvpair_value_byte.9f
1168 link path=usr/share/man/man9f/nvpair_value_int32.9f \
1169 target=nvpair_value_byte.9f
1170 link path=usr/share/man/man9f/nvpair_value_int32_array.9f \
1171 target=nvpair_value_byte.9f
1172 link path=usr/share/man/man9f/nvpair_value_int64.9f \
1173 target=nvpair_value_byte.9f
1174 link path=usr/share/man/man9f/nvpair_value_int64_array.9f \
1175 target=nvpair_value_byte.9f
1176 link path=usr/share/man/man9f/nvpair_value_int8.9f target=nvpair_value_byte.9f
1177 link path=usr/share/man/man9f/nvpair_value_int8_array.9f \
1178 target=nvpair_value_byte.9f
1179 link path=usr/share/man/man9f/nvpair_value_nvlist.9f \
1180 target=nvpair_value_byte.9f
1181 link path=usr/share/man/man9f/nvpair_value_nvlist_array.9f \
```

```
1182 target=nvpair_value_byte.9f
1183 link path=usr/share/man/man9f/nvpair_value_string.9f \
1184 target=nvpair_value_byte.9f
1185 link path=usr/share/man/man9f/nvpair_value_string_array.9f \
1186 target=nvpair_value_byte.9f
1187 link path=usr/share/man/man9f/nvpair_value_uint16.9f \
1188 target=nvpair_value_byte.9f
1189 link path=usr/share/man/man9f/nvpair_value_uint16_array.9f \
1190 target=nvpair_value_byte.9f
1191 link path=usr/share/man/man9f/nvpair_value_uint32.9f \
1192 target=nvpair_value_byte.9f
1193 link path=usr/share/man/man9f/nvpair_value_uint32_array.9f \
1194 target=nvpair_value_byte.9f
1195 link path=usr/share/man/man9f/nvpair_value_uint64.9f \
1196 target=nvpair_value_byte.9f
1197 link path=usr/share/man/man9f/nvpair_value_uint64_array.9f \
1198 target=nvpair_value_byte.9f
1199 link path=usr/share/man/man9f/nvpair_value_uint8.9f \
1200 target=nvpair_value_byte.9f
1201 link path=usr/share/man/man9f/nvpair_value_uint8_array.9f \
1202 target=nvpair_value_byte.9f
1203 link path=usr/share/man/man9f/otherq.9f target=OTHERQ.9f
1204 link path=usr/share/man/man9f/outl.9f target=outb.9f
1205 link path=usr/share/man/man9f/outw.9f target=outb.9f
1206 link path=usr/share/man/man9f/pci_config_get16.9f target=pci_config_get8.9f
1207 link path=usr/share/man/man9f/pci_config_get32.9f target=pci_config_get8.9f
1208 link path=usr/share/man/man9f/pci_config_get64.9f target=pci_config_get8.9f
1209 link path=usr/share/man/man9f/pci_config_getb.9f target=pci_config_get8.9f
1210 link path=usr/share/man/man9f/pci_config_getl.9f target=pci_config_get8.9f
1211 link path=usr/share/man/man9f/pci_config_getll.9f target=pci_config_get8.9f
1212 link path=usr/share/man/man9f/pci_config_getw.9f target=pci_config_get8.9f
1213 link path=usr/share/man/man9f/pci_config_put16.9f target=pci_config_get8.9f
1214 link path=usr/share/man/man9f/pci_config_put32.9f target=pci_config_get8.9f
1215 link path=usr/share/man/man9f/pci_config_put64.9f target=pci_config_get8.9f
1216 link path=usr/share/man/man9f/pci_config_put8.9f target=pci_config_get8.9f
1217 link path=usr/share/man/man9f/pci_config_putb.9f target=pci_config_get8.9f
1218 link path=usr/share/man/man9f/pci_config_putl.9f target=pci_config_get8.9f
1219 link path=usr/share/man/man9f/pci_config_putll.9f target=pci_config_get8.9f
1220 link path=usr/share/man/man9f/pci_config_putw.9f target=pci_config_get8.9f
1221 link path=usr/share/man/man9f/pci_config_teardown.9f \
1222 target=pci_config_setup.9f
1223 link path=usr/share/man/man9f/pci_ereport_post.9f target=pci_ereport_setup.9f
1224 link path=usr/share/man/man9f/pci_ereport_teardown.9f \
1225 target=pci_ereport_setup.9f
1226 link path=usr/share/man/man9f/pci_restore_config_regs.9f \
1227 target=pci_save_config_regs.9f
1228 link path=usr/share/man/man9f/pm_idle_component.9f target=pm_busy_component.9f
1229 link path=usr/share/man/man9f/pm_lower_power.9f target=pm_raise_power.9f
1230 link path=usr/share/man/man9f/priv_policy_choice.9f target=priv_policy.9f
1231 link path=usr/share/man/man9f/priv_policy_only.9f target=priv_policy.9f
1232 link path=usr/share/man/man9f/proc_ref.9f target=proc_signal.9f
1233 link path=usr/share/man/man9f/proc_unref.9f target=proc_signal.9f
1234 link path=usr/share/man/man9f/qprocsoff.9f target=qprocson.9f
1235 link path=usr/share/man/man9f/qwait_sig.9f target=qwait.9f
1236 link path=usr/share/man/man9f/rd.9f target=RD.9f
1237 link path=usr/share/man/man9f/repinsb.9f target=inb.9f
1238 link path=usr/share/man/man9f/repinsd.9f target=inb.9f
1239 link path=usr/share/man/man9f/repinsw.9f target=inb.9f
1240 link path=usr/share/man/man9f/repoutsb.9f target=outb.9f
1241 link path=usr/share/man/man9f/repoutsd.9f target=outb.9f
1242 link path=usr/share/man/man9f/repoutsw.9f target=outb.9f
1243 link path=usr/share/man/man9f/rmallocmap_wait.9f target=rmallocmap.9f
1244 link path=usr/share/man/man9f/rmfreemap.9f target=rmallocmap.9f
1245 link path=usr/share/man/man9f/rw_destroy.9f target=rwlock.9f
1246 link path=usr/share/man/man9f/rw_downgrade.9f target=rwlock.9f
1247 link path=usr/share/man/man9f/rw_enter.9f target=rwlock.9f
```

```
1248 link path=usr/share/man/man9f/rw_exit.9f target=rwlock.9f
1249 link path=usr/share/man/man9f/rw_init.9f target=rwlock.9f
1250 link path=usr/share/man/man9f/rw_read_locked.9f target=rwlock.9f
1251 link path=usr/share/man/man9f/rw_tryenter.9f target=rwlock.9f
1252 link path=usr/share/man/man9f/rw_tryupgrade.9f target=rwlock.9f
1253 link path=usr/share/man/man9f/samestr.9f target=SAMESTR.9f
1254 link path=usr/share/man/man9f/scsi_dmafree.9f target=scsi_dmaget.9f
1255 link path=usr/share/man/man9f/scsi_dname.9f target=scsi_cname.9f
1256 link path=usr/share/man/man9f/scsi_hba_detach.9f \
1257 target=scsi_hba_attach_setup.9f
1258 link path=usr/share/man/man9f/scsi_hba_fini.9f target=scsi_hba_init.9f
1259 link path=usr/share/man/man9f/scsi_hba_pkt_free.9f \
1260 target=scsi_hba_pkt_alloc.9f
1261 link path=usr/share/man/man9f/scsi_hba_tran_free.9f \
1262 target=scsi_hba_tran_alloc.9f
1263 link path=usr/share/man/man9f/scsi_ifsetcap.9f target=scsi_ifgetcap.9f
1264 link path=usr/share/man/man9f/scsi_mname.9f target=scsi_cname.9f
1265 link path=usr/share/man/man9f/scsi_pktfree.9f target=scsi_pktalloc.9f
1266 link path=usr/share/man/man9f/scsi_realloc.9f target=scsi_pktalloc.9f
1267 link path=usr/share/man/man9f/scsi_resfree.9f target=scsi_pktalloc.9f
1268 link path=usr/share/man/man9f/scsi_rname.9f target=scsi_cname.9f
1269 link path=usr/share/man/man9f/scsi_sense_asc.9f target=scsi_sense_key.9f
1270 link path=usr/share/man/man9f/scsi_sense_ascq.9f target=scsi_sense_key.9f
1271 link path=usr/share/man/man9f/scsi_sense_cmdspecific_uint64.9f \
1272 target=scsi_ext_sense_fields.9f
1273 link path=usr/share/man/man9f/scsi_sense_info_uint64.9f \
1274 target=scsi_ext_sense_fields.9f
1275 link path=usr/share/man/man9f/scsi_sname.9f target=scsi_cname.9f
1276 link path=usr/share/man/man9f/scsi_unslave.9f target=scsi_unprobe.9f
1277 link path=usr/share/man/man9f/sem_destroy.9f target=semaphore.9f
1278 link path=usr/share/man/man9f/sem_init.9f target=semaphore.9f
1279 link path=usr/share/man/man9f/sem_p.9f target=semaphore.9f
1280 link path=usr/share/man/man9f/sem_p_sig.9f target=semaphore.9f
1281 link path=usr/share/man/man9f/sem_try.9f target=semaphore.9f
1282 link path=usr/share/man/man9f/sem_v.9f target=semaphore.9f
1283 link path=usr/share/man/man9f/snprintf.9f target=sprintf.9f
1284 link path=usr/share/man/man9f/strcasemp.9f target=string.9f
1285 link path=usr/share/man/man9f/strcat.9f target=string.9f
1286 link path=usr/share/man/man9f/strchr.9f target=string.9f
1287 link path=usr/share/man/man9f/strcmp.9f target=string.9f
1288 link path=usr/share/man/man9f/strcpy.9f target=string.9f
1289 link path=usr/share/man/man9f/strdup.9f target=string.9f
1290 link path=usr/share/man/man9f/strfree.9f target=string.9f
1291 link path=usr/share/man/man9f/strlcat.9f target=string.9f
1292 link path=usr/share/man/man9f/strncpy.9f target=string.9f
1293 link path=usr/share/man/man9f/strlen.9f target=string.9f
1294 link path=usr/share/man/man9f/strncasemp.9f target=string.9f
1295 link path=usr/share/man/man9f/strncat.9f target=string.9f
1296 link path=usr/share/man/man9f/strncmp.9f target=string.9f
1297 link path=usr/share/man/man9f/strncpy.9f target=string.9f
1298 link path=usr/share/man/man9f/strnlen.9f target=string.9f
1299 link path=usr/share/man/man9f/strrchr.9f target=string.9f
1300 link path=usr/share/man/man9f/strspn.9f target=string.9f
1301 link path=usr/share/man/man9f/taskq_suspended.9f target=taskq.9f
1302 link path=usr/share/man/man9f/uconv_ul6tou8.9f target=uconv_ul6tou32.9f
1303 link path=usr/share/man/man9f/uconv_u32tou16.9f target=uconv_ul6tou32.9f
1304 link path=usr/share/man/man9f/uconv_u32tou8.9f target=uconv_ul6tou32.9f
1305 link path=usr/share/man/man9f/uconv_u8tou16.9f target=uconv_ul6tou32.9f
1306 link path=usr/share/man/man9f/uconv_u8tou32.9f target=uconv_ul6tou32.9f
1307 link path=usr/share/man/man9f/unfreezestr.9f target=freezestr.9f
1308 link path=usr/share/man/man9f/va_copy.9f target=va_arg.9f
1309 link path=usr/share/man/man9f/va_end.9f target=va_arg.9f
1310 link path=usr/share/man/man9f/va_start.9f target=va_arg.9f
1311 link path=usr/share/man/man9f/vcmm_err.9f target=cmn_err.9f
1312 link path=usr/share/man/man9f/vmem_xalloc.9f target=vmem_alloc.9f
1313 link path=usr/share/man/man9f/vmem_free.9f target=vmem_alloc.9f
```



```
1314 link path=usr/share/man/man9f/vmem_xfree.9f target=vmem_alloc.9f
1315 link path=usr/share/man/man9f/vmem_xcreate.9f target=vmem_create.9f
1316 link path=usr/share/man/man9f/vmem_destroy.9f target=vmem_create.9f
1317 link path=usr/share/man/man9f/vmem_size.9f target=vmem_walk.9f
1318 #endif /* ! codereview */
1319 link path=usr/share/man/man9f/vsnprintf.9f target=sprintf.9f
1320 link path=usr/share/man/man9f/vsprintf.9f target=sprintf.9f
1321 link path=usr/share/man/man9f/vzcmn_err.9f target=cmn_err.9f
1322 link path=usr/share/man/man9f/wr.9f target=WR.9f
1323 link path=usr/share/man/man9f/zcmn_err.9f target=cmn_err.9f
```

```

*****
54917 Thu Feb  2 13:24:21 2017
new/usr/src/uts/common/os/vmem.c
7831 want vmem manual pages
7832 big theory statements need a place in the manual
*****
1 /*
2  * CDDL HEADER START
3  *
4  * The contents of this file are subject to the terms of the
5  * Common Development and Distribution License (the "License").
6  * You may not use this file except in compliance with the License.
7  *
8  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
9  * or http://www.opensolaris.org/os/licensing.
10 * See the License for the specific language governing permissions
11 * and limitations under the License.
12 *
13 * When distributing Covered Code, include this CDDL HEADER in each
14 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
15 * If applicable, add the following below this CDDL HEADER, with the
16 * fields enclosed by brackets "[ ]" replaced with your own identifying
17 * information: Portions Copyright [yyyy] [name of copyright owner]
18 *
19 * CDDL HEADER END
20 */
21 /*
22  * Copyright 2010 Sun Microsystems, Inc. All rights reserved.
23  * Use is subject to license terms.
24 */
25
26 /*
27  * Copyright (c) 2012, 2015 by Delphix. All rights reserved.
28  * Copyright (c) 2012, Joyent, Inc. All rights reserved.
29 */
30
31 /*
32  * Big Theory Statement for the virtual memory allocator.
33  *
34  * For a more complete description of the main ideas, see:
35  *
36  *     Jeff Bonwick and Jonathan Adams,
37  *
38  *     Magazines and vmem: Extending the Slab Allocator to Many CPUs and
39  *     Arbitrary Resources.
40  *
41  *     Proceedings of the 2001 Usenix Conference.
42  *     Available as http://www.usenix.org/event/usenix01/bonwick.html
43  *
44  * Section 1, below, is also the primary contents of vmem(9).  If for some
45  * reason you are updating this comment, you will also wish to update the
46  * manual.
47 #endif /* ! codereview */
48 *
49  * 1. General Concepts
50  * -----
51 *
52  * 1.1 Overview
53  * -----
54  * We divide the kernel address space into a number of logically distinct
55  * pieces, or *arenas*: text, data, heap, stack, and so on.  Within these
56  * arenas we often subdivide further; for example, we use heap addresses
57  * not only for the kernel heap (kmem_alloc() space), but also for DVMA,
58  * bp_mapin(), /dev/kmem, and even some device mappings like the TOD chip.
59  * The kernel address space, therefore, is most accurately described as
60  * a tree of arenas in which each node of the tree *imports* some subset

```

```

61  * of its parent.  The virtual memory allocator manages these arenas and
62  * supports their natural hierarchical structure.
63  *
64  * 1.2 Arenas
65  * -----
66  * An arena is nothing more than a set of integers.  These integers most
67  * commonly represent virtual addresses, but in fact they can represent
68  * anything at all.  For example, we could use an arena containing the
69  * integers minpid through maxpid to allocate process IDs.  vmem_create()
70  * and vmem_destroy() create and destroy vmem arenas.  In order to
71  * differentiate between arenas used for addresses and arenas used for
72  * identifiers, the VMC_IDENTIFIER flag is passed to vmem_create().  This
73  * prevents identifier exhaustion from being diagnosed as general memory
74  * failure.
75  *
76  * 1.3 Spans
77  * -----
78  * We represent the integers in an arena as a collection of *spans*, or
79  * contiguous ranges of integers.  For example, the kernel heap consists
80  * of just one span: [kernelheap, ekernelheap).  Spans can be added to an
81  * arena in two ways: explicitly, by vmem_add(), or implicitly, by
82  * importing, as described in Section 1.5 below.
83  *
84  * 1.4 Segments
85  * -----
86  * Spans are subdivided into *segments*, each of which is either allocated
87  * or free.  A segment, like a span, is a contiguous range of integers.
88  * Each allocated segment [addr, addr + size) represents exactly one
89  * vmem_alloc(size) that returned addr.  Free segments represent the space
90  * between allocated segments.  If two free segments are adjacent, we
91  * coalesce them into one larger segment; that is, if segments [a, b) and
92  * [b, c) are both free, we merge them into a single segment [a, c).
93  * The segments within a span are linked together in increasing-address order
94  * so we can easily determine whether coalescing is possible.
95  *
96  * Segments never cross span boundaries.  When all segments within
97  * an imported span become free, we return the span to its source.
98  *
99  * 1.5 Imported Memory
100 * -----
101 * As mentioned in the overview, some arenas are logical subsets of
102 * other arenas.  For example, kmem_va_arena (a virtual address cache
103 * that satisfies most kmem_slab_create() requests) is just a subset
104 * of heap_arena (the kernel heap) that provides caching for the most
105 * common slab sizes.  When kmem_va_arena runs out of virtual memory,
106 * it *imports* more from the heap; we say that heap_arena is the
107 * *vmem source* for kmem_va_arena.  vmem_create() allows you to
108 * specify any existing vmem arena as the source for your new arena.
109 * Topologically, since every arena is a child of at most one source,
110 * the set of all arenas forms a collection of trees.
111 *
112 * 1.6 Constrained Allocations
113 * -----
114 * Some vmem clients are quite picky about the kind of address they want.
115 * For example, the DVMA code may need an address that is at a particular
116 * phase with respect to some alignment (to get good cache coloring), or
117 * that lies within certain limits (the addressable range of a device),
118 * or that doesn't cross some boundary (a DMA counter restriction) --
119 * or all of the above.  vmem_xalloc() allows the client to specify any
120 * or all of these constraints.
121 *
122 * 1.7 The Vmem Quantum
123 * -----
124 * Every arena has a notion of 'quantum', specified at vmem_create() time,
125 * that defines the arena's minimum unit of currency.  Most commonly the
126 * quantum is either 1 or PAGE_SIZE, but any power of 2 is legal.

```

```

127 * All vmem allocations are guaranteed to be quantum-aligned.
128 *
129 * 1.8 Quantum Caching
130 * -----
131 * A vmem arena may be so hot (frequently used) that the scalability of vmem
132 * allocation is a significant concern. We address this by allowing the most
133 * common allocation sizes to be serviced by the kernel memory allocator,
134 * which provides low-latency per-cpu caching. The qcache_max argument to
135 * vmem_create() specifies the largest allocation size to cache.
136 *
137 * 1.9 Relationship to Kernel Memory Allocator
138 * -----
139 * Every kmem cache has a vmem arena as its slab supplier. The kernel memory
140 * allocator uses vmem_alloc() and vmem_free() to create and destroy slabs.
141 *
142 *
143 * 2. Implementation
144 * -----
145 *
146 * 2.1 Segment lists and markers
147 * -----
148 * The segment structure (vmem_seg_t) contains two doubly-linked lists.
149 *
150 * The arena list (vs_anext/vs_aprev) links all segments in the arena.
151 * In addition to the allocated and free segments, the arena contains
152 * special marker segments at span boundaries. Span markers simplify
153 * coalescing and importing logic by making it easy to tell both when
154 * we're at a span boundary (so we don't coalesce across it), and when
155 * a span is completely free (its neighbors will both be span markers).
156 *
157 * Imported spans will have vs_import set.
158 *
159 * The next-of-kin list (vs_knext/vs_kprev) links segments of the same type:
160 * (1) for allocated segments, vs_knext is the hash chain linkage;
161 * (2) for free segments, vs_knext is the freelist linkage;
162 * (3) for span marker segments, vs_knext is the next span marker.
163 *
164 * 2.2 Allocation hashing
165 * -----
166 * We maintain a hash table of all allocated segments, hashed by address.
167 * This allows vmem_free() to discover the target segment in constant time.
168 * vmem_update() periodically resizes hash tables to keep hash chains short.
169 *
170 * 2.3 Freelist management
171 * -----
172 * We maintain power-of-2 freelists for free segments, i.e. free segments
173 * of size >= 2^n reside in vmp->vm_freelist[n]. To ensure constant-time
174 * allocation, vmem_xalloc() looks not in the first freelist that *might*
175 * satisfy the allocation, but in the first freelist that *definitely*
176 * satisfies the allocation (unless VM_BESTFIT is specified, or all larger
177 * freelists are empty). For example, a 1000-byte allocation will be
178 * satisfied not from the 512..1023-byte freelist, whose members *might*
179 * contain a 1000-byte segment, but from a 1024-byte or larger freelist,
180 * the first member of which will *definitely* satisfy the allocation.
181 * This ensures that vmem_xalloc() works in constant time.
182 *
183 * We maintain a bit map to determine quickly which freelists are non-empty.
184 * vmp->vm_freemap & (1 << n) is non-zero iff vmp->vm_freelist[n] is non-empty.
185 *
186 * The different freelists are linked together into one large freelist,
187 * with the freelist heads serving as markers. Freelist markers simplify
188 * the maintenance of vm_freemap by making it easy to tell when we're taking
189 * the last member of a freelist (both of its neighbors will be markers).
190 *
191 * 2.4 Vmem Locking
192 * -----

```

```

193 * For simplicity, all arena state is protected by a per-arena lock.
194 * For very hot arenas, use quantum caching for scalability.
195 *
196 * 2.5 Vmem Population
197 * -----
198 * Any internal vmem routine that might need to allocate new segment
199 * structures must prepare in advance by calling vmem_populate(), which
200 * will preallocate enough vmem_seg_t's to get is through the entire
201 * operation without dropping the arena lock.
202 *
203 * 2.6 Auditing
204 * -----
205 * If KMF_AUDIT is set in kmem_flags, we audit vmem allocations as well.
206 * Since virtual addresses cannot be scribbled on, there is no equivalent
207 * in vmem to redzone checking, deadbeef, or other kmem debugging features.
208 * Moreover, we do not audit frees because segment coalescing destroys the
209 * association between an address and its segment structure. Auditing is
210 * thus intended primarily to keep track of who's consuming the arena.
211 * Debugging support could certainly be extended in the future if it proves
212 * necessary, but we do so much live checking via the allocation hash table
213 * that even non-DEBUG systems get quite a bit of sanity checking already.
214 */

216 #include <sys/vmem_impl.h>
217 #include <sys/kmem.h>
218 #include <sys/kstat.h>
219 #include <sys/param.h>
220 #include <sys/system.h>
221 #include <sys/atomic.h>
222 #include <sys/bitmap.h>
223 #include <sys/sysmacros.h>
224 #include <sys/cmn_err.h>
225 #include <sys/debug.h>
226 #include <sys/panic.h>

228 #define VMEM_INITIAL          10      /* early vmem arenas */
229 #define VMEM_SEG_INITIAL     200     /* early segments */

231 /*
232 * Adding a new span to an arena requires two segment structures: one to
233 * represent the span, and one to represent the free segment it contains.
234 */
235 #define VMEM_SEGS_PER_SPAN_CREATE    2

237 /*
238 * Allocating a piece of an existing segment requires 0-2 segment structures
239 * depending on how much of the segment we're allocating.
240 *
241 * To allocate the entire segment, no new segment structures are needed; we
242 * simply move the existing segment structure from the freelist to the
243 * allocation hash table.
244 *
245 * To allocate a piece from the left or right end of the segment, we must
246 * split the segment into two pieces (allocated part and remainder), so we
247 * need one new segment structure to represent the remainder.
248 *
249 * To allocate from the middle of a segment, we need two new segment structures
250 * to represent the remainders on either side of the allocated part.
251 */
252 #define VMEM_SEGS_PER_EXACT_ALLOC    0
253 #define VMEM_SEGS_PER_LEFT_ALLOC    1
254 #define VMEM_SEGS_PER_RIGHT_ALLOC   1
255 #define VMEM_SEGS_PER_MIDDLE_ALLOC  2

257 /*
258 * vmem_populate() preallocates segment structures for vmem to do its work.

```

```

259 * It must preallocate enough for the worst case, which is when we must import
260 * a new span and then allocate from the middle of it.
261 */
262 #define VMEM_SEGS_PER_ALLOC_MAX      \
263     (VMEM_SEGS_PER_SPAN_CREATE + VMEM_SEGS_PER_MIDDLE_ALLOC)

265 /*
266 * The segment structures themselves are allocated from vmem_seg_arena, so
267 * we have a recursion problem when vmem_seg_arena needs to populate itself.
268 * We address this by working out the maximum number of segment structures
269 * this act will require, and multiplying by the maximum number of threads
270 * that we'll allow to do it simultaneously.
271 *
272 * The worst-case segment consumption to populate vmem_seg_arena is as
273 * follows (depicted as a stack trace to indicate why events are occurring):
274 *
275 * (In order to lower the fragmentation in the heap_arena, we specify a
276 * minimum import size for the vmem_metadata_arena which is the same size
277 * as the kmem_va quantum cache allocations. This causes the worst-case
278 * allocation from the vmem_metadata_arena to be 3 segments.)
279 *
280 * vmem_alloc(vmem_seg_arena)      -> 2 segs (span create + exact alloc)
281 * segkmem_alloc(vmem_metadata_arena)
282 * vmem_alloc(vmem_metadata_arena) -> 3 segs (span create + left alloc)
283 * vmem_alloc(heap_arena)         -> 1 seg (left alloc)
284 * page_create()
285 * hat_memload()
286 * kmem_cache_alloc()
287 * kmem_slab_create()
288 * vmem_alloc(hat_memload_arena)   -> 2 segs (span create + exact alloc)
289 * segkmem_alloc(heap_arena)
290 * vmem_alloc(heap_arena)         -> 1 seg (left alloc)
291 * page_create()
292 * hat_memload()                 -> (hat layer won't recurse further)
293 *
294 * The worst-case consumption for each arena is 3 segment structures.
295 * Of course, a 3-seg reserve could easily be blown by multiple threads.
296 * Therefore, we serialize all allocations from vmem_seg_arena (which is OK
297 * because they're rare). We cannot allow a non-blocking allocation to get
298 * tied up behind a blocking allocation, however, so we use separate locks
299 * for VM_SLEEP and VM_NOSLEEP allocations. Similarly, VM_PUSHPAGE allocations
300 * must not block behind ordinary VM_SLEEPS. In addition, if the system is
301 * panicking then we must keep enough resources for panic_thread to do its
302 * work. Thus we have at most four threads trying to allocate from
303 * vmem_seg_arena, and each thread consumes at most three segment structures,
304 * so we must maintain a 12-seg reserve.
305 */
306 #define VMEM_POPULATE_RESERVE  12

308 /*
309 * vmem_populate() ensures that each arena has VMEM_MINFREE seg structures
310 * so that it can satisfy the worst-case allocation *and* participate in
311 * worst-case allocation from vmem_seg_arena.
312 */
313 #define VMEM_MINFREE           (VMEM_POPULATE_RESERVE + VMEM_SEGS_PER_ALLOC_MAX)

315 static vmem_t vmem0[VMEM_INITIAL];
316 static vmem_t *vmem_populator[VMEM_INITIAL];
317 static uint32_t vmem_id;
318 static uint32_t vmem_populators;
319 static vmem_seg_t vmem_seg0[VMEM_SEG_INITIAL];
320 static vmem_seg_t *vmem_segfree;
321 static kmutex_t vmem_list_lock;
322 static kmutex_t vmem_segfree_lock;
323 static kmutex_t vmem_sleep_lock;
324 static kmutex_t vmem_nosleep_lock;

```

```

325 static kmutex_t vmem_pushpage_lock;
326 static kmutex_t vmem_panic_lock;
327 static vmem_t *vmem_list;
328 static vmem_t *vmem_metadata_arena;
329 static vmem_t *vmem_seg_arena;
330 static vmem_t *vmem_hash_arena;
331 static vmem_t *vmem_vmem_arena;
332 static long vmem_update_interval = 15; /* vmem_update() every 15 seconds */
333 uint32_t vmem_mtbtf; /* mean time between failures [default: off] */
334 size_t vmem_seg_size = sizeof (vmem_seg_t);

336 static vmem_kstat_t vmem_kstat_template = {
337     { "mem_inuse",          KSTAT_DATA_UINT64 },
338     { "mem_import",        KSTAT_DATA_UINT64 },
339     { "mem_total",         KSTAT_DATA_UINT64 },
340     { "vmem_source",       KSTAT_DATA_UINT32 },
341     { "alloc",             KSTAT_DATA_UINT64 },
342     { "free",              KSTAT_DATA_UINT64 },
343     { "wait",              KSTAT_DATA_UINT64 },
344     { "fail",              KSTAT_DATA_UINT64 },
345     { "lookup",            KSTAT_DATA_UINT64 },
346     { "search",            KSTAT_DATA_UINT64 },
347     { "populate_wait",     KSTAT_DATA_UINT64 },
348     { "populate_fail",     KSTAT_DATA_UINT64 },
349     { "contains",          KSTAT_DATA_UINT64 },
350     { "contains_search",   KSTAT_DATA_UINT64 },
351 };

353 /*
354 * Insert/delete from arena list (type 'a') or next-of-kin list (type 'k').
355 */
356 #define VMEM_INSERT(vprev, vsp, type) \
357 { \
358     vmem_seg_t *vnext = (vprev)->vs_##type##next; \
359     (vsp)->vs_##type##next = (vnext); \
360     (vsp)->vs_##type##prev = (vprev); \
361     (vprev)->vs_##type##next = (vsp); \
362     (vnext)->vs_##type##prev = (vsp); \
363 }

365 #define VMEM_DELETE(vsp, type) \
366 { \
367     vmem_seg_t *vprev = (vsp)->vs_##type##prev; \
368     vmem_seg_t *vnext = (vsp)->vs_##type##next; \
369     (vprev)->vs_##type##next = (vnext); \
370     (vnext)->vs_##type##prev = (vprev); \
371 }

373 /*
374 * Get a vmem_seg_t from the global segfree list.
375 */
376 static vmem_seg_t *
377 vmem_getseg_global(void)
378 {
379     vmem_seg_t *vsp;

381     mutex_enter(&vmem_segfree_lock);
382     if ((vsp = vmem_segfree) != NULL)
383         vmem_segfree = vsp->vs_knext;
384     mutex_exit(&vmem_segfree_lock);

386     return (vsp);
387 }

389 /*
390 * Put a vmem_seg_t on the global segfree list.

```

```

391 */
392 static void
393 vmem_putseg_global(vmem_seg_t *vsp)
394 {
395     mutex_enter(&vmem_segfree_lock);
396     vsp->vs_knext = vmem_segfree;
397     vmem_segfree = vsp;
398     mutex_exit(&vmem_segfree_lock);
399 }
401 /*
402 * Get a vmem_seg_t from vmp's segfree list.
403 */
404 static vmem_seg_t *
405 vmem_getseg(vmem_t *vmp)
406 {
407     vmem_seg_t *vsp;
409     ASSERT(vmp->vm_nsegfree > 0);
411     vsp = vmp->vm_segfree;
412     vmp->vm_segfree = vsp->vs_knext;
413     vmp->vm_nsegfree--;
415     return (vsp);
416 }
418 /*
419 * Put a vmem_seg_t on vmp's segfree list.
420 */
421 static void
422 vmem_putseg(vmem_t *vmp, vmem_seg_t *vsp)
423 {
424     vsp->vs_knext = vmp->vm_segfree;
425     vmp->vm_segfree = vsp;
426     vmp->vm_nsegfree++;
427 }
429 /*
430 * Add vsp to the appropriate freelist.
431 */
432 static void
433 vmem_freelist_insert(vmem_t *vmp, vmem_seg_t *vsp)
434 {
435     vmem_seg_t *vprev;
437     ASSERT(*VMEM_HASH(vmp, vsp->vs_start) != vsp);
439     vprev = (vmem_seg_t *)&vmp->vm_freelist[highbit(VS_SIZE(vsp)) - 1];
440     vsp->vs_type = VMEM_FREE;
441     vmp->vm_freemap |= VS_SIZE(vprev);
442     VMEM_INSERT(vprev, vsp, k);
444     cv_broadcast(&vmp->vm_cv);
445 }
447 /*
448 * Take vsp from the freelist.
449 */
450 static void
451 vmem_freelist_delete(vmem_t *vmp, vmem_seg_t *vsp)
452 {
453     ASSERT(*VMEM_HASH(vmp, vsp->vs_start) != vsp);
454     ASSERT(vsp->vs_type == VMEM_FREE);
456     if (vsp->vs_knext->vs_start == 0 && vsp->vs_kprev->vs_start == 0) {

```

```

457     /*
458     * The segments on both sides of 'vsp' are freelist heads,
459     * so taking vsp leaves the freelist at vsp->vs_kprev empty.
460     */
461     ASSERT(vmp->vm_freemap & VS_SIZE(vsp->vs_kprev));
462     vmp->vm_freemap ^= VS_SIZE(vsp->vs_kprev);
463 }
464     VMEM_DELETE(vsp, k);
465 }
467 /*
468 * Add vsp to the allocated-segment hash table and update kstats.
469 */
470 static void
471 vmem_hash_insert(vmem_t *vmp, vmem_seg_t *vsp)
472 {
473     vmem_seg_t **bucket;
475     vsp->vs_type = VMEM_ALLOC;
476     bucket = VMEM_HASH(vmp, vsp->vs_start);
477     vsp->vs_knext = *bucket;
478     *bucket = vsp;
480     if (vmem_seg_size == sizeof (vmem_seg_t)) {
481         vsp->vs_depth = (uint8_t)getpcstack(vsp->vs_stack,
482             VMEM_STACK_DEPTH);
483         vsp->vs_thread = curthread;
484         vsp->vs_timestamp = gethrtime();
485     } else {
486         vsp->vs_depth = 0;
487     }
489     vmp->vm_kstat.vk_alloc.value.ui64++;
490     vmp->vm_kstat.vk_mem_inuse.value.ui64 += VS_SIZE(vsp);
491 }
493 /*
494 * Remove vsp from the allocated-segment hash table and update kstats.
495 */
496 static vmem_seg_t *
497 vmem_hash_delete(vmem_t *vmp, uintptr_t addr, size_t size)
498 {
499     vmem_seg_t *vsp, **prev_vspp;
501     prev_vspp = VMEM_HASH(vmp, addr);
502     while ((vsp = *prev_vspp) != NULL) {
503         if (vsp->vs_start == addr) {
504             *prev_vspp = vsp->vs_knext;
505             break;
506         }
507         vmp->vm_kstat.vk_lookup.value.ui64++;
508         prev_vspp = &vsp->vs_knext;
509     }
511     if (vsp == NULL)
512         panic("vmem_hash_delete(%p, %lx, %lu): bad free",
513             (void *)vmp, addr, size);
514     if (VS_SIZE(vsp) != size)
515         panic("vmem_hash_delete(%p, %lx, %lu): wrong size (expect %lu)",
516             (void *)vmp, addr, size, VS_SIZE(vsp));
518     vmp->vm_kstat.vk_free.value.ui64++;
519     vmp->vm_kstat.vk_mem_inuse.value.ui64 -= size;
521     return (vsp);
522 }

```

```

524 /*
525  * Create a segment spanning the range [start, end) and add it to the arena.
526  */
527 static vmem_seg_t *
528 vmem_seg_create(vmem_t *vmp, vmem_seg_t *vprev, uintptr_t start, uintptr_t end)
529 {
530     vmem_seg_t *newseg = vmem_getseg(vmp);
531
532     newseg->vs_start = start;
533     newseg->vs_end = end;
534     newseg->vs_type = 0;
535     newseg->vs_import = 0;
536
537     VMEM_INSERT(vprev, newseg, a);
538
539     return (newseg);
540 }
541
542 /*
543  * Remove segment vsp from the arena.
544  */
545 static void
546 vmem_seg_destroy(vmem_t *vmp, vmem_seg_t *vsp)
547 {
548     ASSERT(vsp->vs_type != VMEM_ROTOR);
549     VMEM_DELETE(vsp, a);
550
551     vmem_putseg(vmp, vsp);
552 }
553
554 /*
555  * Add the span [vaddr, vaddr + size) to vmp and update kstats.
556  */
557 static vmem_seg_t *
558 vmem_span_create(vmem_t *vmp, void *vaddr, size_t size, uint8_t import)
559 {
560     vmem_seg_t *newseg, *span;
561     uintptr_t start = (uintptr_t)vaddr;
562     uintptr_t end = start + size;
563
564     ASSERT(MUTEX_HELD(&vmp->vm_lock));
565
566     if ((start | end) & (vmp->vm_quantum - 1))
567         panic("vmem_span_create(%p, %p, %lu): misaligned",
568             (void *)vmp, vaddr, size);
569
570     span = vmem_seg_create(vmp, vmp->vm_seg0.vs_aprev, start, end);
571     span->vs_type = VMEM_SPAN;
572     span->vs_import = import;
573     VMEM_INSERT(vmp->vm_seg0.vs_kprev, span, k);
574
575     newseg = vmem_seg_create(vmp, span, start, end);
576     vmem_freelist_insert(vmp, newseg);
577
578     if (import)
579         vmp->vm_kstat.vk_mem_import.value.ui64 += size;
580     vmp->vm_kstat.vk_mem_total.value.ui64 += size;
581
582     return (newseg);
583 }
584
585 /*
586  * Remove span vsp from vmp and update kstats.
587  */
588 static void

```

```

589 vmem_span_destroy(vmem_t *vmp, vmem_seg_t *vsp)
590 {
591     vmem_seg_t *span = vsp->vs_aprev;
592     size_t size = VS_SIZE(vsp);
593
594     ASSERT(MUTEX_HELD(&vmp->vm_lock));
595     ASSERT(span->vs_type == VMEM_SPAN);
596
597     if (span->vs_import)
598         vmp->vm_kstat.vk_mem_import.value.ui64 -= size;
599     vmp->vm_kstat.vk_mem_total.value.ui64 -= size;
600
601     VMEM_DELETE(span, k);
602
603     vmem_seg_destroy(vmp, vsp);
604     vmem_seg_destroy(vmp, span);
605 }
606
607 /*
608  * Allocate the subrange [addr, addr + size) from segment vsp.
609  * If there are leftovers on either side, place them on the freelist.
610  * Returns a pointer to the segment representing [addr, addr + size).
611  */
612 static vmem_seg_t *
613 vmem_seg_alloc(vmem_t *vmp, vmem_seg_t *vsp, uintptr_t addr, size_t size)
614 {
615     uintptr_t vs_start = vsp->vs_start;
616     uintptr_t vs_end = vsp->vs_end;
617     size_t vs_size = vs_end - vs_start;
618     size_t realsize = P2ROUNDUP(size, vmp->vm_quantum);
619     uintptr_t addr_end = addr + realsize;
620
621     ASSERT(P2PHASE(vs_start, vmp->vm_quantum) == 0);
622     ASSERT(P2PHASE(addr, vmp->vm_quantum) == 0);
623     ASSERT(vsp->vs_type == VMEM_FREE);
624     ASSERT(addr >= vs_start && addr_end - 1 <= vs_end - 1);
625     ASSERT(addr - 1 <= addr_end - 1);
626
627     /*
628      * If we're allocating from the start of the segment, and the
629      * remainder will be on the same freelist, we can save quite
630      * a bit of work.
631      */
632     if (P2SAMEHIGHBIT(vs_size, vs_size - realsize) && addr == vs_start) {
633         ASSERT(highbit(vs_size) == highbit(vs_size - realsize));
634         vsp->vs_start = addr_end;
635         vsp = vmem_seg_create(vmp, vsp->vs_aprev, addr, addr + size);
636         vmem_hash_insert(vmp, vsp);
637         return (vsp);
638     }
639
640     vmem_freelist_delete(vmp, vsp);
641
642     if (vs_end != addr_end)
643         vmem_freelist_insert(vmp,
644             vmem_seg_create(vmp, vsp, addr_end, vs_end));
645
646     if (vs_start != addr)
647         vmem_freelist_insert(vmp,
648             vmem_seg_create(vmp, vsp->vs_aprev, vs_start, addr));
649
650     vsp->vs_start = addr;
651     vsp->vs_end = addr + size;
652
653     vmem_hash_insert(vmp, vsp);
654     return (vsp);

```

```

655 }
657 /*
658  * Returns 1 if we are populating, 0 otherwise.
659  * Call it if we want to prevent recursion from HAT.
660  */
661 int
662 vmem_is_populator()
663 {
664     return (mutex_owner(&vmem_sleep_lock) == curthread ||
665             mutex_owner(&vmem_nosleep_lock) == curthread ||
666             mutex_owner(&vmem_pushpage_lock) == curthread ||
667             mutex_owner(&vmem_panic_lock) == curthread);
668 }
670 /*
671  * Populate vmp's segfree list with VMEM_MINFREE vmem_seg_t structures.
672  */
673 static int
674 vmem_populate(vmem_t *vmp, int vmflag)
675 {
676     char *p;
677     vmem_seg_t *vsp;
678     ssize_t nseg;
679     size_t size;
680     kmutex_t *lp;
681     int i;
683     while (vmp->vm_nsegfree < VMEM_MINFREE &&
684            (vsp = vmem_getseg_global()) != NULL)
685         vmem_putseg(vmp, vsp);
687     if (vmp->vm_nsegfree >= VMEM_MINFREE)
688         return (1);
690     /*
691      * If we're already populating, tap the reserve.
692      */
693     if (vmem_is_populator()) {
694         ASSERT(vmp->vm_cflags & VMC_POPULATOR);
695         return (1);
696     }
698     mutex_exit(&vmp->vm_lock);
700     if (panic_thread == curthread)
701         lp = &vmem_panic_lock;
702     else if (vmflag & VM_NOSLEEP)
703         lp = &vmem_nosleep_lock;
704     else if (vmflag & VM_PUSHPAGE)
705         lp = &vmem_pushpage_lock;
706     else
707         lp = &vmem_sleep_lock;
709     mutex_enter(lp);
711     nseg = VMEM_MINFREE + vmem_populators * VMEM_POPULATE_RESERVE;
712     size = P2ROUNDUP(nseg * vmem_seg_size, vmem_seg_arena->vm_quantum);
713     nseg = size / vmem_seg_size;
715     /*
716      * The following vmem_alloc() may need to populate vmem_seg_arena
717      * and all the things it imports from. When doing so, it will tap
718      * each arena's reserve to prevent recursion (see the block comment
719      * above the definition of VMEM_POPULATE_RESERVE).
720      */

```

```

721     p = vmem_alloc(vmem_seg_arena, size, vmflag & VM_KMFLAGS);
722     if (p == NULL) {
723         mutex_exit(lp);
724         mutex_enter(&vmp->vm_lock);
725         vmp->vm_kstat.vk_populate_fail.value.ui64++;
726         return (0);
727     }
729     /*
730      * Restock the arenas that may have been depleted during population.
731      */
732     for (i = 0; i < vmem_populators; i++) {
733         mutex_enter(&vmem_populator[i]->vm_lock);
734         while (vmem_populator[i]->vm_nsegfree < VMEM_POPULATE_RESERVE)
735             vmem_putseg(vmem_populator[i],
736                         (vmem_seg_t *) (p + --nseg * vmem_seg_size));
737         mutex_exit(&vmem_populator[i]->vm_lock);
738     }
740     mutex_exit(lp);
741     mutex_enter(&vmp->vm_lock);
743     /*
744      * Now take our own segments.
745      */
746     ASSERT(nseg >= VMEM_MINFREE);
747     while (vmp->vm_nsegfree < VMEM_MINFREE)
748         vmem_putseg(vmp, (vmem_seg_t *) (p + --nseg * vmem_seg_size));
750     /*
751      * Give the remainder to charity.
752      */
753     while (nseg > 0)
754         vmem_putseg_global((vmem_seg_t *) (p + --nseg * vmem_seg_size));
756     return (1);
757 }
759 /*
760  * Advance a walker from its previous position to 'afterme'.
761  * Note: may drop and reacquire vmp->vm_lock.
762  */
763 static void
764 vmem_advance(vmem_t *vmp, vmem_seg_t *walker, vmem_seg_t *afterme)
765 {
766     vmem_seg_t *vprev = walker->vs_aprev;
767     vmem_seg_t *vnext = walker->vs_anext;
768     vmem_seg_t *vsp = NULL;
770     VMEM_DELETE(walker, a);
772     if (afterme != NULL)
773         VMEM_INSERT(afterme, walker, a);
775     /*
776      * The walker segment's presence may have prevented its neighbors
777      * from coalescing. If so, coalesce them now.
778      */
779     if (vprev->vs_type == VMEM_FREE) {
780         if (vnext->vs_type == VMEM_FREE) {
781             ASSERT(vprev->vs_end == vnext->vs_start);
782             vmem_freelist_delete(vmp, vnext);
783             vmem_freelist_delete(vmp, vprev);
784             vprev->vs_end = vnext->vs_end;
785             vmem_freelist_insert(vmp, vprev);
786             vmem_seg_destroy(vmp, vnext);

```

```

787     }
788     vsp = vprev;
789 } else if (vnext->vs_type == VMEM_FREE) {
790     vsp = vnext;
791 }
792
793 /*
794  * vsp could represent a complete imported span,
795  * in which case we must return it to the source.
796  */
797 if (vsp != NULL && vsp->vs_aprev->vs_import &&
798     vmp->vm_source_free != NULL &&
799     vsp->vs_aprev->vs_type == VMEM_SPAN &&
800     vsp->vs_anext->vs_type == VMEM_SPAN) {
801     void *vaddr = (void *)vsp->vs_start;
802     size_t size = VS_SIZE(vsp);
803     ASSERT(size == VS_SIZE(vsp->vs_aprev));
804     vmem_freelist_delete(vmp, vsp);
805     vmem_span_destroy(vmp, vsp);
806     mutex_exit(&vmp->vm_lock);
807     vmp->vm_source_free(vmp->vm_source, vaddr, size);
808     mutex_enter(&vmp->vm_lock);
809 }
810 }
811
812 /*
813  * VM_NEXTFIT allocations deliberately cycle through all virtual addresses
814  * in an arena, so that we avoid reusing addresses for as long as possible.
815  * This helps to catch used-after-freed bugs. It's also the perfect policy
816  * for allocating things like process IDs, where we want to cycle through
817  * all values in order.
818  */
819 static void *
820 vmem_nextfit_alloc(vmem_t *vmp, size_t size, int vmflag)
821 {
822     vmem_seg_t *vsp, *rotor;
823     uintptr_t addr;
824     size_t realsize = P2ROUNDUP(size, vmp->vm_quantum);
825     size_t vs_size;
826
827     mutex_enter(&vmp->vm_lock);
828
829     if (vmp->vm_nsegfree < VMEM_MINFREE && !vmem_populate(vmp, vmflag)) {
830         mutex_exit(&vmp->vm_lock);
831         return (NULL);
832     }
833
834     /*
835      * The common case is that the segment right after the rotor is free,
836      * and large enough that extracting 'size' bytes won't change which
837      * freelist it's on. In this case we can avoid a *lot* of work.
838      * Instead of the normal vmem_seg_alloc(), we just advance the start
839      * address of the victim segment. Instead of moving the rotor, we
840      * create the new segment structure *behind* the rotor*, which has
841      * the same effect. And finally, we know we don't have to coalesce
842      * the rotor's neighbors because the new segment lies between them.
843      */
844     rotor = &vmp->vm_rotor;
845     vsp = rotor->vs_anext;
846     if (vsp->vs_type == VMEM_FREE && (vs_size = VS_SIZE(vsp)) > realsize &&
847         P2SAMEHIGHBIT(vs_size, vs_size - realsize)) {
848         ASSERT(highbit(vs_size) == highbit(vs_size - realsize));
849         addr = vsp->vs_start;
850         vsp->vs_start = addr + realsize;
851         vmem_hash_insert(vmp,
852             vmem_seg_create(vmp, rotor->vs_aprev, addr, addr + size));

```

```

853         mutex_exit(&vmp->vm_lock);
854         return ((void *)addr);
855     }
856
857     /*
858      * Starting at the rotor, look for a segment large enough to
859      * satisfy the allocation.
860      */
861     for (;;) {
862         vmp->vm_kstat.vk_search.value.ui64++;
863         if (vsp->vs_type == VMEM_FREE && VS_SIZE(vsp) >= size)
864             break;
865         vsp = vsp->vs_anext;
866         if (vsp == rotor) {
867             /*
868              * We've come full circle. One possibility is that the
869              * there's actually enough space, but the rotor itself
870              * is preventing the allocation from succeeding because
871              * it's sitting between two free segments. Therefore,
872              * we advance the rotor and see if that liberates a
873              * suitable segment.
874              */
875             vmem_advance(vmp, rotor, rotor->vs_anext);
876             vsp = rotor->vs_aprev;
877             if (vsp->vs_type == VMEM_FREE && VS_SIZE(vsp) >= size)
878                 break;
879             /*
880              * If there's a lower arena we can import from, or it's
881              * a VM_NOSLEEP allocation, let vmem_xalloc() handle it.
882              * Otherwise, wait until another thread frees something.
883              */
884             if (vmp->vm_source_alloc != NULL ||
885                 (vmflag & VM_NOSLEEP)) {
886                 mutex_exit(&vmp->vm_lock);
887                 return (vmem_xalloc(vmp, size, vmp->vm_quantum,
888                     0, 0, NULL, NULL, vmflag & VM_KMFLAGS));
889             }
890             vmp->vm_kstat.vk_wait.value.ui64++;
891             cv_wait(&vmp->vm_cv, &vmp->vm_lock);
892             vsp = rotor->vs_anext;
893         }
894     }
895
896     /*
897      * We found a segment. Extract enough space to satisfy the allocation.
898      */
899     addr = vsp->vs_start;
900     vsp = vmem_seg_alloc(vmp, vsp, addr, size);
901     ASSERT(vsp->vs_type == VMEM_ALLOC &&
902         vsp->vs_start == addr && vsp->vs_end == addr + size);
903
904     /*
905      * Advance the rotor to right after the newly-allocated segment.
906      * That's where the next VM_NEXTFIT allocation will begin searching.
907      */
908     vmem_advance(vmp, rotor, vsp);
909     mutex_exit(&vmp->vm_lock);
910     return ((void *)addr);
911 }
912
913 /*
914  * Checks if vmp is guaranteed to have a size-byte buffer somewhere on its
915  * freelist. If size is not a power-of-2, it can return a false-negative.
916  *
917  * Used to decide if a newly imported span is superfluous after re-acquiring
918  * the arena lock.

```



```

919 */
920 static int
921 vmem_canalloc(vmem_t *vmp, size_t size)
922 {
923     int hb;
924     int flist = 0;
925     ASSERT(MUTEX_HELD(&vmp->vm_lock));

927     if (ISP2(size))
928         flist = lowbit(P2ALIGN(vmp->vm_freemap, size));
929     else if ((hb = highbit(size)) < VMEM_FREELISTS)
930         flist = lowbit(P2ALIGN(vmp->vm_freemap, 1UL << hb));

932     return (flist);
933 }

935 /*
936  * Allocate size bytes at offset phase from an align boundary such that the
937  * resulting segment [addr, addr + size) is a subset of [minaddr, maxaddr)
938  * that does not straddle a nocross-aligned boundary.
939  */
940 void *
941 vmem_xalloc(vmem_t *vmp, size_t size, size_t align_arg, size_t phase,
942             size_t nocross, void *minaddr, void *maxaddr, int vmflag)
943 {
944     vmem_seg_t *vsp;
945     vmem_seg_t *vbest = NULL;
946     uintptr_t addr, taddr, start, end;
947     uintptr_t align = (align_arg != 0) ? align_arg : vmp->vm_quantum;
948     void *vaddr, *xvaddr = NULL;
949     size_t xsize;
950     int hb, flist, resv;
951     uint32_t mtbf;

953     if ((align | phase | nocross) & (vmp->vm_quantum - 1))
954         panic("vmem_xalloc(%p, %lu, %lu, %lu, %lu, %p, %p, %x): "
955              "parameters not vm_quantum aligned",
956              (void *)vmp, size, align_arg, phase, nocross,
957              minaddr, maxaddr, vmflag);

959     if (nocross != 0 &&
960         (align > nocross || P2ROUNDUP(phase + size, align) > nocross))
961         panic("vmem_xalloc(%p, %lu, %lu, %lu, %lu, %p, %p, %x): "
962              "overconstrained allocation",
963              (void *)vmp, size, align_arg, phase, nocross,
964              minaddr, maxaddr, vmflag);

966     if (phase >= align || !ISP2(align) || !ISP2(nocross))
967         panic("vmem_xalloc(%p, %lu, %lu, %lu, %lu, %p, %p, %x): "
968              "parameters inconsistent or invalid",
969              (void *)vmp, size, align_arg, phase, nocross,
970              minaddr, maxaddr, vmflag);

972     if ((mtbf = vmem_mtbf | vmp->vm_mtbf) != 0 && gethrtime() % mtbf == 0 &&
973         (vmflag & (VM_NOSLEEP | VM_PANIC)) == VM_NOSLEEP)
974         return (NULL);

976     mutex_enter(&vmp->vm_lock);
977     for (;;) {
978         if (vmp->vm_nsegfree < VMEM_MINFREE &&
979             !vmem_populate(vmp, vmflag))
980             break;
981     do_alloc:
982         /*
983          * highbit() returns the highest bit + 1, which is exactly
984          * what we want: we want to search the first freelist whose

```

```

985     * members are *definitely* large enough to satisfy our
986     * allocation. However, there are certain cases in which we
987     * want to look at the next-smallest freelist (which *might*
988     * be able to satisfy the allocation):
989     *
990     * (1) The size is exactly a power of 2, in which case
991     *     the smaller freelist is always big enough;
992     *
993     * (2) All other freelists are empty;
994     *
995     * (3) We're in the highest possible freelist, which is
996     *     always empty (e.g. the 4GB freelist on 32-bit systems);
997     *
998     * (4) We're doing a best-fit or first-fit allocation.
999     */
1000     if (ISP2(size)) {
1001         flist = lowbit(P2ALIGN(vmp->vm_freemap, size));
1002     } else {
1003         hb = highbit(size);
1004         if ((vmp->vm_freemap >> hb) == 0 ||
1005             hb == VMEM_FREELISTS |
1006             (vmflag & (VM_BESTFIT | VM_FIRSTFIT)))
1007             hb--;
1008         flist = lowbit(P2ALIGN(vmp->vm_freemap, 1UL << hb));
1009     }

1011     for (vbest = NULL, vsp = (flist == 0) ? NULL :
1012          vmp->vm_freelist[flist - 1].vs_knext;
1013          vsp != NULL; vsp = vsp->vs_knext) {
1014         vmp->vm_kstat.vk_search.value.ui64++;
1015         if (vsp->vs_start == 0) {
1016             /*
1017              * We're moving up to a larger freelist,
1018              * so if we've already found a candidate,
1019              * the fit can't possibly get any better.
1020              */
1021             if (vbest != NULL)
1022                 break;
1023             /*
1024              * Find the next non-empty freelist.
1025              */
1026             flist = lowbit(P2ALIGN(vmp->vm_freemap,
1027                                  VS_SIZE(vsp)));
1028             if (flist-- == 0)
1029                 break;
1030             vsp = (vmem_seg_t *)&vmp->vm_freelist[flist];
1031             ASSERT(vsp->vs_knext->vs_type == VMEM_FREE);
1032             continue;
1033         }
1034         if (vsp->vs_end - 1 < (uintptr_t)minaddr)
1035             continue;
1036         if (vsp->vs_start > (uintptr_t)maxaddr - 1)
1037             continue;
1038         start = MAX(vsp->vs_start, (uintptr_t)minaddr);
1039         end = MIN(vsp->vs_end - 1, (uintptr_t)maxaddr - 1) + 1;
1040         taddr = P2PHASEUP(start, align, phase);
1041         if (P2BOUNDARY(taddr, size, nocross))
1042             taddr +=
1043                 P2ROUNDUP(P2NPHASE(taddr, nocross), align);
1044         if ((taddr - start) + size > end - start ||
1045             (vbest != NULL && VS_SIZE(vsp) >= VS_SIZE(vbest)))
1046             continue;
1047         vbest = vsp;
1048         addr = taddr;
1049         if (!(vmflag & VM_BESTFIT) || VS_SIZE(vbest) == size)
1050             break;

```

```

1051     }
1052     if (vbest != NULL)
1053         break;
1054     ASSERT(xvaddr == NULL);
1055     if (size == 0)
1056         panic("vmem_xalloc(): size == 0");
1057     if (vmp->vm_source_alloc != NULL && nocross == 0 &&
1058         minaddr == NULL && maxaddr == NULL) {
1059         size_t aneeded, asize;
1060         size_t aquantum = MAX(vmp->vm_quantum,
1061             vmp->vm_source->vm_quantum);
1062         size_t aphase = phase;
1063         if ((align > aquantum) &&
1064             !(vmp->vm_cflags & VMC_XALIGN)) {
1065             aphase = (P2PHASE(phase, aquantum) != 0) ?
1066                 align - vmp->vm_quantum : align - aquantum;
1067             ASSERT(aphase >= phase);
1068         }
1069         aneeded = MAX(size + aphase, vmp->vm_min_import);
1070         asize = P2ROUNDUP(aneeded, aquantum);
1071
1072         if (asize < size) {
1073             /*
1074              * The rounding induced overflow; return NULL
1075              * if we are permitted to fail the allocation
1076              * (and explicitly panic if we aren't).
1077              */
1078             if ((vmflag & VM_NOSLEEP) &&
1079                 !(vmflag & VM_PANIC)) {
1080                 mutex_exit(&vmp->vm_lock);
1081                 return (NULL);
1082             }
1083
1084             panic("vmem_xalloc(): size overflow");
1085         }
1086
1087         /*
1088          * Determine how many segment structures we'll consume.
1089          * The calculation must be precise because if we're
1090          * here on behalf of vmem_populate(), we are taking
1091          * segments from a very limited reserve.
1092          */
1093         if (size == asize && !(vmp->vm_cflags & VMC_XALLOC))
1094             resv = VMEM_SEGS_PER_SPAN_CREATE +
1095                 VMEM_SEGS_PER_EXACT_ALLOC;
1096         else if (phase == 0 &&
1097             align <= vmp->vm_source->vm_quantum)
1098             resv = VMEM_SEGS_PER_SPAN_CREATE +
1099                 VMEM_SEGS_PER_LEFT_ALLOC;
1100         else
1101             resv = VMEM_SEGS_PER_ALLOC_MAX;
1102
1103         ASSERT(vmp->vm_nsegfree >= resv);
1104         vmp->vm_nsegfree -= resv;          /* reserve our segs */
1105         mutex_exit(&vmp->vm_lock);
1106         if (vmp->vm_cflags & VMC_XALLOC) {
1107             size_t oasize = asize;
1108             vaddr = ((vmem_ximport_t *)
1109                 vmp->vm_source_alloc)(vmp->vm_source,
1110                 &asize, vmflag & VM_KMFLAGS);
1111             ASSERT(asize >= oasize);
1112             ASSERT(P2PHASE(asize,
1113                 vmp->vm_source->vm_quantum) == 0);
1114             ASSERT(!(vmp->vm_cflags & VMC_XALIGN) ||
1115                 IS_P2ALIGNED(vaddr, align));
1116         } else {

```

```

1117             vaddr = vmp->vm_source_alloc(vmp->vm_source,
1118                 asize, vmflag & VM_KMFLAGS);
1119         }
1120         mutex_enter(&vmp->vm_lock);
1121         vmp->vm_nsegfree += resv;        /* claim reservation */
1122         aneeded = size + align - vmp->vm_quantum;
1123         aneeded = P2ROUNDUP(aneeded, vmp->vm_quantum);
1124         if (vaddr != NULL) {
1125             /*
1126              * Since we dropped the vmem lock while
1127              * calling the import function, other
1128              * threads could have imported space
1129              * and made our import unnecessary. In
1130              * order to save space, we return
1131              * excess imports immediately.
1132              */
1133             if (asize > aneeded &&
1134                 vmp->vm_source_free != NULL &&
1135                 vmem_canalloc(vmp, aneeded)) {
1136                 ASSERT(resv >=
1137                     VMEM_SEGS_PER_MIDDLE_ALLOC);
1138                 xvaddr = vaddr;
1139                 xsz = asize;
1140                 goto do_alloc;
1141             }
1142             vbest = vmem_span_create(vmp, vaddr, asize, 1);
1143             addr = P2PHASEUP(vbest->vs_start, align, phase);
1144             break;
1145         } else if (vmem_canalloc(vmp, aneeded)) {
1146             /*
1147              * Our import failed, but another thread
1148              * added sufficient free memory to the arena
1149              * to satisfy our request. Go back and
1150              * grab it.
1151              */
1152             ASSERT(resv >= VMEM_SEGS_PER_MIDDLE_ALLOC);
1153             goto do_alloc;
1154         }
1155     }
1156
1157     /*
1158      * If the requestor chooses to fail the allocation attempt
1159      * rather than reap wait and retry - get out of the loop.
1160      */
1161     if (vmflag & VM_ABORT)
1162         break;
1163     mutex_exit(&vmp->vm_lock);
1164     if (vmp->vm_cflags & VMC_IDENTIFIER)
1165         kmem_reap_idspace();
1166     else
1167         kmem_reap();
1168     mutex_enter(&vmp->vm_lock);
1169     if (vmflag & VM_NOSLEEP)
1170         break;
1171     vmp->vm_kstat.vk_wait.value.ui64++;
1172     cv_wait(&vmp->vm_cv, &vmp->vm_lock);
1173 }
1174 if (vbest != NULL) {
1175     ASSERT(vbest->vs_type == VMEM_FREE);
1176     ASSERT(vbest->vs_knext != vbest);
1177     /* re-position to end of buffer */
1178     if (vmflag & VM_ENDALLOC) {
1179         addr += ((vbest->vs_end - (addr + size)) / align) *
1180             align;
1181     }
1182     (void) vmem_seg_alloc(vmp, vbest, addr, size);

```

```

1183     mutex_exit(&vmp->vm_lock);
1184     if (xvaddr)
1185         vmp->vm_source_free(vmp->vm_source, xvaddr, xsize);
1186     ASSERT(P2PHASE(addr, align) == phase);
1187     ASSERT(!P2BOUNDARY(addr, size, nocross));
1188     ASSERT(addr >= (uintptr_t)minaddr);
1189     ASSERT(addr + size - 1 <= (uintptr_t)maxaddr - 1);
1190     return ((void *)addr);
1191 }
1192 vmp->vm_kstat.vk_fail.value.ui64++;
1193 mutex_exit(&vmp->vm_lock);
1194 if (vmflag & VM_PANIC)
1195     panic("vmem_xalloc(%p, %lu, %lu, %lu, %lu, %p, %p, %x): "
1196         "cannot satisfy mandatory allocation",
1197         (void *)vmp, size, align_arg, phase, nocross,
1198         minaddr, maxaddr, vmflag);
1199 ASSERT(xvaddr == NULL);
1200 return (NULL);
1201 }
1202
1203 /*
1204 * Free the segment [vaddr, vaddr + size], where vaddr was a constrained
1205 * allocation. vmem_xalloc() and vmem_xfree() must always be paired because
1206 * both routines bypass the quantum caches.
1207 */
1208 void
1209 vmem_xfree(vmem_t *vmp, void *vaddr, size_t size)
1210 {
1211     vmem_seg_t *vsp, *vnext, *vprev;
1212
1213     mutex_enter(&vmp->vm_lock);
1214
1215     vsp = vmem_hash_delete(vmp, (uintptr_t)vaddr, size);
1216     vsp->vs_end = P2ROUNDUP(vsp->vs_end, vmp->vm_quantum);
1217
1218     /*
1219      * Attempt to coalesce with the next segment.
1220      */
1221     vnext = vsp->vs_anext;
1222     if (vnext->vs_type == VMEM_FREE) {
1223         ASSERT(vsp->vs_end == vnext->vs_start);
1224         vmem_freelist_delete(vmp, vnext);
1225         vsp->vs_end = vnext->vs_end;
1226         vmem_seg_destroy(vmp, vnext);
1227     }
1228
1229     /*
1230      * Attempt to coalesce with the previous segment.
1231      */
1232     vprev = vsp->vs_aprev;
1233     if (vprev->vs_type == VMEM_FREE) {
1234         ASSERT(vprev->vs_end == vsp->vs_start);
1235         vmem_freelist_delete(vmp, vprev);
1236         vprev->vs_end = vsp->vs_end;
1237         vmem_seg_destroy(vmp, vprev);
1238         vsp = vprev;
1239     }
1240
1241     /*
1242      * If the entire span is free, return it to the source.
1243      */
1244     if (vsp->vs_aprev->vs_import && vmp->vm_source_free != NULL &&
1245         vsp->vs_aprev->vs_type == VMEM_SPAN &&
1246         vsp->vs_anext->vs_type == VMEM_SPAN) {
1247         vaddr = (void *)vsp->vs_start;
1248         size = VS_SIZE(vsp);

```

```

1249         ASSERT(size == VS_SIZE(vsp->vs_aprev));
1250         vmem_span_destroy(vmp, vsp);
1251         mutex_exit(&vmp->vm_lock);
1252         vmp->vm_source_free(vmp->vm_source, vaddr, size);
1253     } else {
1254         vmem_freelist_insert(vmp, vsp);
1255         mutex_exit(&vmp->vm_lock);
1256     }
1257 }
1258
1259 /*
1260 * Allocate size bytes from arena vmp. Returns the allocated address
1261 * on success, NULL on failure. vmflag specifies VM_SLEEP or VM_NOSLEEP,
1262 * and may also specify best-fit, first-fit, or next-fit allocation policy
1263 * instead of the default instant-fit policy. VM_SLEEP allocations are
1264 * guaranteed to succeed.
1265 */
1266 void *
1267 vmem_alloc(vmem_t *vmp, size_t size, int vmflag)
1268 {
1269     vmem_seg_t *vsp;
1270     uintptr_t addr;
1271     int hb;
1272     int flist = 0;
1273     uint32_t mtbf;
1274
1275     if (size - 1 < vmp->vm_qcache_max)
1276         return (kmem_cache_alloc(vmp->vm_qcache[(size - 1) >>
1277             vmp->vm_qshift], vmflag & VM_KMFLAGS));
1278
1279     if ((mtbf = vmem_mtbf | vmp->vm_mtbf) != 0 && gethrtime() % mtbf == 0 &&
1280         (vmflag & (VM_NOSLEEP | VM_PANIC)) == VM_NOSLEEP)
1281         return (NULL);
1282
1283     if (vmflag & VM_NEXTFIT)
1284         return (vmem_nextfit_alloc(vmp, size, vmflag));
1285
1286     if (vmflag & (VM_BESTFIT | VM_FIRSTFIT))
1287         return (vmem_xalloc(vmp, size, vmp->vm_quantum, 0, 0,
1288             NULL, NULL, vmflag));
1289
1290     /*
1291      * Unconstrained instant-fit allocation from the segment list.
1292      */
1293     mutex_enter(&vmp->vm_lock);
1294
1295     if (vmp->vm_nsegfree >= VMEM_MINFREE || vmem_populate(vmp, vmflag)) {
1296         if (ISP2(size))
1297             flist = lowbit(P2ALIGN(vmp->vm_freemap, size));
1298         else if ((hb = highbit(size)) < VMEM_FREELISTS)
1299             flist = lowbit(P2ALIGN(vmp->vm_freemap, 1UL << hb));
1300     }
1301
1302     if (flist-- == 0) {
1303         mutex_exit(&vmp->vm_lock);
1304         return (vmem_xalloc(vmp, size, vmp->vm_quantum,
1305             0, 0, NULL, NULL, vmflag));
1306     }
1307
1308     ASSERT(size <= (1UL << flist));
1309     vsp = vmp->vm_freelist[flist].vs_knext;
1310     addr = vsp->vs_start;
1311     if (vmflag & VM_ENDALLOC) {
1312         addr += vsp->vs_end - (addr + size);
1313     }
1314     (void) vmem_seg_alloc(vmp, vsp, addr, size);

```

```

1315     mutex_exit(&vmp->vm_lock);
1316     return ((void *)addr);
1317 }

1319 /*
1320  * Free the segment [vaddr, vaddr + size).
1321  */
1322 void
1323 vmem_free(vmem_t *vmp, void *vaddr, size_t size)
1324 {
1325     if (size - 1 < vmp->vm_qcache_max)
1326         kmem_cache_free(vmp->vm_qcache[(size - 1) >> vmp->vm_qshift],
1327             vaddr);
1328     else
1329         vmem_xfree(vmp, vaddr, size);
1330 }

1332 /*
1333  * Determine whether arena vmp contains the segment [vaddr, vaddr + size).
1334  */
1335 int
1336 vmem_contains(vmem_t *vmp, void *vaddr, size_t size)
1337 {
1338     uintptr_t start = (uintptr_t)vaddr;
1339     uintptr_t end = start + size;
1340     vmem_seg_t *vsp;
1341     vmem_seg_t *seg0 = &vmp->vm_seg0;

1343     mutex_enter(&vmp->vm_lock);
1344     vmp->vm_kstat.vk_contains.value.ui64++;
1345     for (vsp = seg0->vs_knext; vsp != seg0; vsp = vsp->vs_knext) {
1346         vmp->vm_kstat.vk_contains_search.value.ui64++;
1347         ASSERT(vsp->vs_type == VMEM_SPAN);
1348         if (start >= vsp->vs_start && end - 1 <= vsp->vs_end - 1)
1349             break;
1350     }
1351     mutex_exit(&vmp->vm_lock);
1352     return (vsp != seg0);
1353 }

1355 /*
1356  * Add the span [vaddr, vaddr + size) to arena vmp.
1357  */
1358 void *
1359 vmem_add(vmem_t *vmp, void *vaddr, size_t size, int vmflag)
1360 {
1361     if (vaddr == NULL || size == 0)
1362         panic("vmem_add(%p, %p, %lu): bad arguments",
1363             (void *)vmp, vaddr, size);

1365     ASSERT(!vmem_contains(vmp, vaddr, size));

1367     mutex_enter(&vmp->vm_lock);
1368     if (vmem_populate(vmp, vmflag))
1369         (void) vmem_span_create(vmp, vaddr, size, 0);
1370     else
1371         vaddr = NULL;
1372     mutex_exit(&vmp->vm_lock);
1373     return (vaddr);
1374 }

1376 /*
1377  * Walk the vmp arena, applying func to each segment matching typemask.
1378  * If VMEM_REENTRANT is specified, the arena lock is dropped across each
1379  * call to func(); otherwise, it is held for the duration of vmem_walk()
1380  * to ensure a consistent snapshot. Note that VMEM_REENTRANT callbacks

```

```

1381  * are *not* necessarily consistent, so they may only be used when a hint
1382  * is adequate.
1383  */
1384 void
1385 vmem_walk(vmem_t *vmp, int typemask,
1386     void (*func)(void *, void *, size_t), void *arg)
1387 {
1388     vmem_seg_t *vsp;
1389     vmem_seg_t *seg0 = &vmp->vm_seg0;
1390     vmem_seg_t walker;

1392     if (typemask & VMEM_WALKER)
1393         return;

1395     bzero(&walker, sizeof (walker));
1396     walker.vs_type = VMEM_WALKER;

1398     mutex_enter(&vmp->vm_lock);
1399     VMEM_INSERT(seg0, &walker, a);
1400     for (vsp = seg0->vs_anext; vsp != seg0; vsp = vsp->vs_anext) {
1401         if (vsp->vs_type & typemask) {
1402             void *start = (void *)vsp->vs_start;
1403             size_t size = VS_SIZE(vsp);
1404             if (typemask & VMEM_REENTRANT) {
1405                 vmem_advance(vmp, &walker, vsp);
1406                 mutex_exit(&vmp->vm_lock);
1407                 func(arg, start, size);
1408                 mutex_enter(&vmp->vm_lock);
1409                 vsp = &walker;
1410             } else {
1411                 func(arg, start, size);
1412             }
1413         }
1414     }
1415     vmem_advance(vmp, &walker, NULL);
1416     mutex_exit(&vmp->vm_lock);
1417 }

1419 /*
1420  * Return the total amount of memory whose type matches typemask. Thus:
1421  *
1422  * typemask VMEM_ALLOC yields total memory allocated (in use).
1423  * typemask VMEM_FREE yields total memory free (available).
1424  * typemask (VMEM_ALLOC | VMEM_FREE) yields total arena size.
1425  */
1426 size_t
1427 vmem_size(vmem_t *vmp, int typemask)
1428 {
1429     uint64_t size = 0;

1431     if (typemask & VMEM_ALLOC)
1432         size += vmp->vm_kstat.vk_mem_inuse.value.ui64;
1433     if (typemask & VMEM_FREE)
1434         size += vmp->vm_kstat.vk_mem_total.value.ui64 -
1435             vmp->vm_kstat.vk_mem_inuse.value.ui64;
1436     return ((size_t)size);
1437 }

1439 /*
1440  * Create an arena called name whose initial span is [base, base + size).
1441  * The arena's natural unit of currency is quantum, so vmem_alloc()
1442  * guarantees quantum-aligned results. The arena may import new spans
1443  * by invoking afunc() on source, and may return those spans by invoking
1444  * ffunc() on source. To make small allocations fast and scalable,
1445  * the arena offers high-performance caching for each integer multiple
1446  * of quantum up to qcache_max.

```

```

1447 */
1448 static vmem_t *
1449 vmem_create_common(const char *name, void *base, size_t size, size_t quantum,
1450 void *(*afunc)(vmem_t *, size_t, int),
1451 void *(*ffunc)(vmem_t *, void *, size_t),
1452 vmem_t *source, size_t qcache_max, int vmflag)
1453 {
1454     int i;
1455     size_t nqcache;
1456     vmem_t *vmp, *cur, **vmpp;
1457     vmem_seg_t *vsp;
1458     vmem_freelist_t *vfp;
1459     uint32_t id = atomic_inc_32_nv(&vmem_id);

1461     if (vmem_vmem_arena != NULL) {
1462         vmp = vmem_alloc(vmem_vmem_arena, sizeof (vmem_t),
1463             vmflag & VM_KMFLAGS);
1464     } else {
1465         ASSERT(id <= VMEM_INITIAL);
1466         vmp = &vmem0[id - 1];
1467     }

1469     /* An identifier arena must inherit from another identifier arena */
1470     ASSERT(source == NULL || ((source->vm_cflags & VMC_IDENTIFIER) ==
1471         (vmflag & VMC_IDENTIFIER)));

1473     if (vmp == NULL)
1474         return (NULL);
1475     bzero(vmp, sizeof (vmem_t));

1477     (void) snprintf(vmp->vm_name, VMEM_NAMELEN, "%s", name);
1478     mutex_init(&vmp->vm_lock, NULL, MUTEX_DEFAULT, NULL);
1479     cv_init(&vmp->vm_cv, NULL, CV_DEFAULT, NULL);
1480     vmp->vm_cflags = vmflag;
1481     vmflag &= VM_KMFLAGS;

1483     vmp->vm_quantum = quantum;
1484     vmp->vm_qshift = highbit(quantum) - 1;
1485     nqcache = MIN(qcache_max >> vmp->vm_qshift, VMEM_NQCACHE_MAX);

1487     for (i = 0; i <= VMEM_FREELISTS; i++) {
1488         vfp = &vmp->vm_freelist[i];
1489         vfp->vs_end = 1UL << i;
1490         vfp->vs_knext = (vmem_seg_t *) (vfp + 1);
1491         vfp->vs_kprev = (vmem_seg_t *) (vfp - 1);
1492     }

1494     vmp->vm_freelist[0].vs_kprev = NULL;
1495     vmp->vm_freelist[VMEM_FREELISTS].vs_knext = NULL;
1496     vmp->vm_freelist[VMEM_FREELISTS].vs_end = 0;
1497     vmp->vm_hash_table = vmp->vm_hash0;
1498     vmp->vm_hash_mask = VMEM_HASH_INITIAL - 1;
1499     vmp->vm_hash_shift = highbit(vmp->vm_hash_mask);

1501     vsp = &vmp->vm_seg0;
1502     vsp->vs_anext = vsp;
1503     vsp->vs_aprev = vsp;
1504     vsp->vs_knext = vsp;
1505     vsp->vs_kprev = vsp;
1506     vsp->vs_type = VMEM_SPAN;

1508     vsp = &vmp->vm_rotor;
1509     vsp->vs_type = VMEM_ROTOR;
1510     VMEM_INSERT(&vmp->vm_seg0, vsp, a);

1512     bcopy(&vmem_kstat_template, &vmp->vm_kstat, sizeof (vmem_kstat_t));

```

```

1514     vmp->vm_id = id;
1515     if (source != NULL)
1516         vmp->vm_kstat.vk_source_id.value.ui32 = source->vm_id;
1517     vmp->vm_source = source;
1518     vmp->vm_source_alloc = afunc;
1519     vmp->vm_source_free = ffunc;

1521     /*
1522     * Some arenas (like vmem metadata and kmem metadata) cannot
1523     * use quantum caching to lower fragmentation. Instead, we
1524     * increase their imports, giving a similar effect.
1525     */
1526     if (vmp->vm_cflags & VMC_NO_QCACHE) {
1527         vmp->vm_min_import =
1528             VMEM_QCACHE_SLABSIZE(nqcache << vmp->vm_qshift);
1529         nqcache = 0;
1530     }

1532     if (nqcache != 0) {
1533         ASSERT(!(vmflag & VM_NOSLEEP));
1534         vmp->vm_qcache_max = nqcache << vmp->vm_qshift;
1535         for (i = 0; i < nqcache; i++) {
1536             char buf[VMEM_NAMELEN + 21];
1537             (void) sprintf(buf, "%s %lu", vmp->vm_name,
1538                 (i + 1) * quantum);
1539             vmp->vm_qcache[i] = kmem_cache_create(buf,
1540                 (i + 1) * quantum, quantum, NULL, NULL, NULL,
1541                 NULL, vmp, KMC_QCACHE | KMC_NOTOUCH);
1542         }
1543     }

1545     if ((vmp->vm_ksp = kstat_create("vmem", vmp->vm_id, vmp->vm_name,
1546         "vmem", KSTAT_TYPE_NAMED, sizeof (vmem_kstat_t) /
1547         sizeof (kstat_named_t), KSTAT_FLAG_VIRTUAL)) != NULL) {
1548         vmp->vm_ksp->ks_data = &vmp->vm_kstat;
1549         kstat_install(vmp->vm_ksp);
1550     }

1552     mutex_enter(&vmem_list_lock);
1553     vmpp = &vmem_list;
1554     while ((cur = *vmpp) != NULL)
1555         vmpp = &cur->vm_next;
1556     *vmpp = vmp;
1557     mutex_exit(&vmem_list_lock);

1559     if (vmp->vm_cflags & VMC_POPULATOR) {
1560         ASSERT(vmem_populators < VMEM_INITIAL);
1561         vmem_populator[atomic_inc_32_nv(&vmem_populators) - 1] = vmp;
1562         mutex_enter(&vmp->vm_lock);
1563         (void) vmem_populate(vmp, vmflag | VM_PANIC);
1564         mutex_exit(&vmp->vm_lock);
1565     }

1567     if ((base || size) && vmem_add(vmp, base, size, vmflag) == NULL) {
1568         vmem_destroy(vmp);
1569         return (NULL);
1570     }

1572     return (vmp);
1573 }

1575 vmem_t *
1576 vmem_xcreate(const char *name, void *base, size_t size, size_t quantum,
1577     vmem_ximport_t *afunc, vmem_free_t *ffunc, vmem_t *source,
1578     size_t qcache_max, int vmflag)

```

```

1579 {
1580     ASSERT(!(vmflag & (VMC_POPULATOR | VMC_XALLOC)));
1581     vmflag &= ~(VMC_POPULATOR | VMC_XALLOC);

1583     return (vmem_create_common(name, base, size, quantum,
1584         (vmem_alloc_t *)afunc, ffunc, source, qcache_max,
1585         vmflag | VMC_XALLOC));
1586 }

1588 vmem_t *
1589 vmem_create(const char *name, void *base, size_t size, size_t quantum,
1590     vmem_alloc_t *afunc, vmem_free_t *ffunc, vmem_t *source,
1591     size_t qcache_max, int vmflag)
1592 {
1593     ASSERT(!(vmflag & (VMC_XALLOC | VMC_XALIGN)));
1594     vmflag &= ~(VMC_XALLOC | VMC_XALIGN);

1596     return (vmem_create_common(name, base, size, quantum,
1597         afunc, ffunc, source, qcache_max, vmflag));
1598 }

1600 /*
1601  * Destroy arena vmp.
1602  */
1603 void
1604 vmem_destroy(vmem_t *vmp)
1605 {
1606     vmem_t *cur, **vmpp;
1607     vmem_seg_t *seg0 = &vmp->vm_seg0;
1608     vmem_seg_t *vsp, *anext;
1609     size_t leaked;
1610     int i;

1612     mutex_enter(&vmem_list_lock);
1613     vmpp = &vmem_list;
1614     while ((cur = *vmpp) != vmp)
1615         vmpp = &cur->vm_next;
1616     *vmpp = vmp->vm_next;
1617     mutex_exit(&vmem_list_lock);

1619     for (i = 0; i < VMEM_NQCACHE_MAX; i++)
1620         if (vmp->vm_qcache[i])
1621             kmem_cache_destroy(vmp->vm_qcache[i]);

1623     leaked = vmem_size(vmp, VMEM_ALLOC);
1624     if (leaked != 0)
1625         cmn_err(CE_WARN, "vmem_destroy('%s'): leaked %lu %s",
1626             vmp->vm_name, leaked, (vmp->vm_cflags & VMC_IDENTIFIER) ?
1627             "identifiers" : "bytes");

1629     if (vmp->vm_hash_table != vmp->vm_hash0)
1630         vmem_free(vmem_hash_arena, vmp->vm_hash_table,
1631             (vmp->vm_hash_mask + 1) * sizeof(void *));

1633     /*
1634      * Give back the segment structures for anything that's left in the
1635      * arena, e.g. the primary spans and their free segments.
1636      */
1637     VMEM_DELETE(&vmp->vm_rotor, a);
1638     for (vsp = seg0->vs_anext; vsp != seg0; vsp = anext) {
1639         anext = vsp->vs_anext;
1640         vmem_putseg_global(vsp);
1641     }

1643     while (vmp->vm_nsegfree > 0)
1644         vmem_putseg_global(vmem_getseg(vmp));

```

```

1646     kstat_delete(vmp->vm_ksp);

1648     mutex_destroy(&vmp->vm_lock);
1649     cv_destroy(&vmp->vm_cv);
1650     vmem_free(vmem_vmem_arena, vmp, sizeof(vmem_t));
1651 }

1653 /*
1654  * Only shrink vmem hashtable if it is 1<<vmem_rescale_minshift times (8x)
1655  * larger than necessary.
1656  */
1657 int vmem_rescale_minshift = 3;

1659 /*
1660  * Resize vmp's hash table to keep the average lookup depth near 1.0.
1661  */
1662 static void
1663 vmem_hash_rescale(vmem_t *vmp)
1664 {
1665     vmem_seg_t **old_table, **new_table, *vsp;
1666     size_t old_size, new_size, h, nseg;

1668     nseg = (size_t)(vmp->vm_kstat.vk_alloc.value.ui64 -
1669         vmp->vm_kstat.vk_free.value.ui64);

1671     new_size = MAX(VMEM_HASH_INITIAL, 1 << (highbit(3 * nseg + 4) - 2));
1672     old_size = vmp->vm_hash_mask + 1;

1674     if ((old_size >> vmem_rescale_minshift) <= new_size &&
1675         new_size <= (old_size << 1))
1676         return;

1678     new_table = vmem_alloc(vmem_hash_arena, new_size * sizeof(void *),
1679         VM_NOSLEEP);
1680     if (new_table == NULL)
1681         return;
1682     bzero(new_table, new_size * sizeof(void *));

1684     mutex_enter(&vmp->vm_lock);

1686     old_size = vmp->vm_hash_mask + 1;
1687     old_table = vmp->vm_hash_table;

1689     vmp->vm_hash_mask = new_size - 1;
1690     vmp->vm_hash_table = new_table;
1691     vmp->vm_hash_shift = highbit(vmp->vm_hash_mask);

1693     for (h = 0; h < old_size; h++) {
1694         vsp = old_table[h];
1695         while (vsp != NULL) {
1696             uintptr_t addr = vsp->vs_start;
1697             vmem_seg_t *next_vsp = vsp->vs_knext;
1698             vmem_seg_t **hash_bucket = VMEM_HASH(vmp, addr);
1699             vsp->vs_knext = *hash_bucket;
1700             *hash_bucket = vsp;
1701             vsp = next_vsp;
1702         }
1703     }

1705     mutex_exit(&vmp->vm_lock);

1707     if (old_table != vmp->vm_hash0)
1708         vmem_free(vmem_hash_arena, old_table,
1709             old_size * sizeof(void *));
1710 }

```

```

1712 /*
1713  * Perform periodic maintenance on all vmem arenas.
1714  */
1715 void
1716 vmem_update(void *dummy)
1717 {
1718     vmem_t *vmp;

1720     mutex_enter(&vmem_list_lock);
1721     for (vmp = vmem_list; vmp != NULL; vmp = vmp->vm_next) {
1722         /*
1723          * If threads are waiting for resources, wake them up
1724          * periodically so they can issue another kmem_reap()
1725          * to reclaim resources cached by the slab allocator.
1726          */
1727         cv_broadcast(&vmp->vm_cv);

1729         /*
1730          * Rescale the hash table to keep the hash chains short.
1731          */
1732         vmem_hash_rescale(vmp);
1733     }
1734     mutex_exit(&vmem_list_lock);

1736     (void) timeout(vmem_update, dummy, vmem_update_interval * hz);
1737 }

1739 void
1740 vmem_qcache_reap(vmem_t *vmp)
1741 {
1742     int i;

1744     /*
1745      * Reap any quantum caches that may be part of this vmem.
1746      */
1747     for (i = 0; i < VMEM_NQCACHE_MAX; i++)
1748         if (vmp->vm_qcache[i])
1749             kmem_cache_reap_now(vmp->vm_qcache[i]);
1750 }

1752 /*
1753  * Prepare vmem for use.
1754  */
1755 vmem_t *
1756 vmem_init(const char *heap_name,
1757          void *heap_start, size_t heap_size, size_t heap_quantum,
1758          void *(*heap_alloc)(vmem_t *, size_t, int),
1759          void (*heap_free)(vmem_t *, void *, size_t))
1760 {
1761     uint32_t id;
1762     int nseg = VMEM_SEG_INITIAL;
1763     vmem_t *heap;

1765     while (--nseg >= 0)
1766         vmem_putseg_global(&vmem_seg0[nseg]);

1768     heap = vmem_create(heap_name,
1769                      heap_start, heap_size, heap_quantum,
1770                      NULL, NULL, NULL, 0,
1771                      VM_SLEEP | VMC_POPULATOR);

1773     vmem_metadata_arena = vmem_create("vmem_metadata",
1774                                     NULL, 0, heap_quantum,
1775                                     vmem_alloc, vmem_free, heap, 8 * heap_quantum,
1776                                     VM_SLEEP | VMC_POPULATOR | VMC_NO_QCACHE);

```

```

1778     vmem_seg_arena = vmem_create("vmem_seg",
1779                                 NULL, 0, heap_quantum,
1780                                 heap_alloc, heap_free, vmem_metadata_arena, 0,
1781                                 VM_SLEEP | VMC_POPULATOR);

1783     vmem_hash_arena = vmem_create("vmem_hash",
1784                                 NULL, 0, 8,
1785                                 heap_alloc, heap_free, vmem_metadata_arena, 0,
1786                                 VM_SLEEP);

1788     vmem_vmem_arena = vmem_create("vmem_vmem",
1789                                 vmem0, sizeof (vmem0), 1,
1790                                 heap_alloc, heap_free, vmem_metadata_arena, 0,
1791                                 VM_SLEEP);

1793     for (id = 0; id < vmem_id; id++)
1794         (void) vmem_xalloc(vmem_vmem_arena, sizeof (vmem_t),
1795                          1, 0, 0, &vmem0[id], &vmem0[id + 1],
1796                          VM_NOSLEEP | VM_BESTFIT | VM_PANIC);

1798     return (heap);
1799 }

```