

```

*****
5512 Mon Aug 26 23:30:17 2013
new/usr/src/cmd/mdb/common/mdb/mdb_gcore.h
27908 ::gcore breaks sparcs build
*****
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 (c) 2013 by Delphix. All rights reserved.
13 */
14
15 #ifndef _MDB_GCORE_H
16 #define _MDB_GCORE_H
17
18 /*
19  * The kernel has its own definition of exit which has a different signature
20  * than the user space definition. This seems to be the standard way to deal
21  * with this.
22  */
23 #define exit kern_exit
24
25 #include <sys/cpuvar.h>
26 #include <sys/cred_impl.h>
27 #include <sys/procfs.h>
28 #include <vm/anon.h>
29
30 #undef exit
31
32 /* mdb versions of kernel structures used for ctf read calls */
33 typedef struct mdb_proc {
34     uintptr_t    p_as;
35     uintptr_t    p_brkbase;
36     size_t       p_brksize;
37     uintptr_t    p_usrstack;
38     size_t       p_stksize;
39     user_t       p_user;
40     uintptr_t    p_agenttp;
41     uintptr_t    p_tlist;
42     uintptr_t    p_zone;
43     uintptr_t    p_ldt;
44     kcondvar_t   p_holdlwp;
45     int          p_lwpcnt;
46     uintptr_t    p_lwpdir;
47     uint_t       p_lwpdir_sz;
48     uintptr_t    p_cred;
49     uint_t       p_flag;
50     int          p_zombcnt;
51     uintptr_t    p_pidp;
52     pid_t        p_ppid;
53     uintptr_t    p_pgidp;
54     uintptr_t    p_sessp;
55     uintptr_t    p_task;
56     uintptr_t    p_pool;
57     model_t      p_model;
58     char         p_wcode;
59     ushort_t     p_ldtlimit;
60     uintptr_t    p_exec;
61     uint_t       p_proc_flag;

```

```

62     ushort_t     p_pidflag;
63     k_sigset_t   p_ignore;
64     k_sigset_t   p_siginfo;
65     k_sigset_t   p_sig;
66     k_sigset_t   p_sigmask;
67     k_fltset_t   p_fltmask;
68     int          p_wdata;
69 } mdb_proc_t;
70
71 typedef struct mdb_kthread {
72     ushort_t     t_proc_flag;
73     uint_t       t_state;
74     lwpchan_t    t_lwpchan;
75     ushort_t     t_whystop;
76     uint8_t      t_dtrace_stop;
77     uintptr_t    t_forw;
78     uintptr_t    t_lwp;
79     id_t         t_tid;
80     short        t_sysnum;
81     pri_t        t_pri;
82     time_t       t_start;
83     id_t         t_cid;
84     uintptr_t    t_cpu;
85     int          t_bind_pset;
86     short        t_bind_cpu;
87     uintptr_t    t_lpl;
88     ushort_t     t_schedflag;
89     ushort_t     t_whatstop;
90     k_sigset_t   t_sig;
91     uintptr_t    t_schedctl;
92     k_sigset_t   t_hold;
93     hrtime_t     t_stoptime;
94 } mdb_kthread_t;
95
96 typedef struct mdb_seg {
97     uintptr_t    s_base;
98     size_t       s_size;
99     uintptr_t    s_ops;
100    uintptr_t    s_data;
101    uintptr_t    s_as;
102 } mdb_seg_t;
103
104 typedef struct mdb_as {
105     uintptr_t    a_proc;
106 } mdb_as_t;
107
108 typedef struct mdb_segvn_data {
109     uintptr_t    vp;
110     uint64_t     offset;
111     uint16_t     flags;
112     uint8_t      pageprot;
113     uint8_t      prot;
114     uintptr_t    amp;
115     struct vpage *vpage;
116     uint64_t     anon_index;
117     uint8_t      type;
118 } mdb_segvn_data_t;
119
120 typedef struct mdb_vnode {
121     enum vtype   v_type;
122     uintptr_t    v_data;
123     uintptr_t    v_op;
124     uintptr_t    v_path;
125 } mdb_vnode_t;
126
127 typedef struct mdb_znode {

```

```

128     uint64_t      z_size;
129 } mdb_znode_t;

131 typedef struct mdb_tmpnode {
132     vattr_t      tn_attr;
133 } mdb_tmpnode_t;

135 typedef struct mdb_vnodeops {
136     uintptr_t    vnop_name;
137 } mdb_vnodeops_t;

139 typedef struct mdb_shm_data {
140     uintptr_t    shm_sptseg;
141 } mdb_shm_data_t;

143 typedef struct mdb_watched_page {
144     uintptr_t    wp_vaddr;
145     uint8_t      wp_oprot;
146 } mdb_watched_page_t;

148 typedef struct mdb_pid {
149     pid_t        pid_id;
150 } mdb_pid_t;

152 typedef struct mdb_sess {
153     uintptr_t    s_sidp;
154 } mdb_sess_t;

156 typedef struct mdb_task {
157     taskid_t     tk_tkid;
158     uintptr_t    tk_proj;
159 } mdb_task_t;

161 typedef struct mdb_kproject {
162     projid_t     kpj_id;
163 } mdb_kproject_t;

165 typedef struct mdb_zone {
166     zoneid_t     zone_id;
167     uintptr_t    zone_name;
168 } mdb_zone_t;

170 typedef struct mdb_sc_shared {
171     char         sc_sigblock;
172 } mdb_sc_shared_t;

174 typedef struct mdb_klwp {
175     uintptr_t    lwp_regs;
176     struct pcb   lwp_pcb;
177     uchar_t     lwp_asleep;
178     uchar_t     lwp_cursig;
179     uintptr_t   lwp_curinfo;
180     k_siginfo_t lwp_siginfo;
181     stack_t     lwp_sigaltstack;
182     uintptr_t   lwp_oldcontext;
183     short       lwp_badpriv;
184     uintptr_t   lwp_ustack;
185     char        lwp_eosys;
186 } mdb_klwp_t;

188 typedef struct mdb_cpu {
189     processorid_t cpu_id;
190 } mdb_cpu_t;

192 typedef struct mdb_lpl {
193     lgrp_id_t    lpl_lgrpid;

```

```

194 } mdb_lpl_t;

196 typedef struct mdb_sigqueue {
197     k_siginfo_t  sq_info;
198 } mdb_sigqueue_t;

200 typedef struct mdb_pool {
201     poolid_t     pool_id;
202 } mdb_pool_t;

204 typedef struct mdb_amp {
205     uintptr_t    ahp;
206 } mdb_amp_t;

208 typedef struct mdb_anon_hdr {
209     pgcnt_t      size;
210     uintptr_t    array_chunk;
211     int          flags;
212 } mdb_anon_hdr_t;

214 typedef struct mdb_anon {
215     uintptr_t    an_vp;
216     anoff_t      an_off;
217 } mdb_anon_t;

219 /* Used to construct a linked list of prmap_ts */
220 typedef struct prmap_node {
221     struct prmap_node *next;
222     prmap_t       m;
223 } prmap_node_t;

225 /* Fields common to psinfo_t and pstatus_t */
226 typedef struct pcommon {
227     int          pc_nlwp;
228     int          pc_nzomb;
229     pid_t        pc_pid;
230     pid_t        pc_ppid;
231     pid_t        pc_pgid;
232     pid_t        pc_sid;
233     taskid_t     pc_taskid;
234     projid_t     pc_projid;
235     zoneid_t     pc_zoneid;
236     char         pc_dmodel;
237 } pcommon_t;

239 /* AVL walk callback structures */
240 typedef struct read_maps_cbarg {
241     mdb_proc_t   *p;
242     uintptr_t    brkseg;
243     uintptr_t    stkseg;
244     prmap_node_t *map_head;
245     prmap_node_t *map_tail;
246     int          map_len;
247 } read_maps_cbarg_t;

249 typedef struct as_segat_cbarg {
250     uintptr_t    addr;
251     uintptr_t    res;
252 } as_segat_cbarg_t;

254 typedef struct getwatchprot_cbarg {
255     uintptr_t    wp_vaddr;
256     mdb_watched_page_t wp;
257     boolean_t    found;
258 } getwatchprot_cbarg_t;

```

```
260 struct gcore_segops;
261 typedef struct gcore_seg {
262     mdb_seg_t      *gs_seg;
263     void           *gs_data;
264     struct gcore_segops *gs_ops;
265 } gcore_seg_t;

267 /*
268  * These are the ISA-dependent functions that need to be
269  * implemented for ::gcore.
270  */
271 extern uintptr_t gcore_prgetstackbase(mdb_proc_t *);
272 extern int gcore_prfetchinstr(mdb_klwp_t *, ulong_t *);
273 extern int gcore_prisstep(mdb_klwp_t *);
274 extern void gcore_getgregs(mdb_klwp_t *, gregset_t);
275 extern int gcore_prgetrvls(mdb_klwp_t *, long *, long *);

277 #endif /* _MDB_GCORE_H */
```

new/usr/src/cmd/mdb/common/modules/genunix/Makefile.files

1

\*\*\*\*\*  
1850 Mon Aug 26 23:30:18 2013

new/usr/src/cmd/mdb/common/modules/genunix/Makefile.files

27908 :gcore breaks sparc build

\*\*\*\*\*

```
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 2011 Nexenta Systems, Inc. All rights reserved.
23 # Copyright (c) 1999, 2010, Oracle and/or its affiliates. All rights reserved.
24 # Copyright (c) 2013, Joyent, Inc. All rights reserved.
25 # Copyright (c) 2013 by Delphix. All rights reserved.
26 #
```

```
28 #
29 # This file simply contains the list of sources files compiled together
30 # to create the genunix mdb module. Having them in one place saves
31 # a bunch of unnecessary replication.
32 #
```

```
33 GENUNIX_SRCS = \
34     avl.c \
35     bio.c \
36     bitset.c \
37     combined.c \
38     contract.c \
39     cpupart.c \
40     cred.c \
41     ctxop.c \
42     cyclic.c \
43     damap.c \
44     ddi_periodic.c \
45     devinfo.c \
46     dist.c \
47     findstack.c \
48     findstack_subr.c \
49     fm.c \
50     gcore.c \
51     gcore_isadep.c \
52     genunix.c \
53     group.c \
54     hotplug.c \
55     irm.c \
56     kgrep.c \
57     kmem.c \
58     ldi.c \
59     leaky.c \
60     leaky_subr.c \
61     lgrp.c \
```

new/usr/src/cmd/mdb/common/modules/genunix/Makefile.files

2

```
62     list.c \
63     log.c \
64     mdi.c \
65     memory.c \
66     mmd.c \
67     modhash.c \
68     ndievents.c \
69     net.c \
70     netstack.c \
71     nvpair.c \
72     pg.c \
73     rctl.c \
74     sobj.c \
75     streams.c \
76     sysevent.c \
77     taskq.c \
78     thread.c \
79     tsd.c \
80     tsol.c \
81     vfs.c \
82     zone.c \
```

```

*****
46377 Mon Aug 26 23:30:19 2013
new/usr/src/cmd/mdb/common/modules/genunix/gcore.c
27908 ::gcore breaks sparc build
*****
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 (c) 2013 by Delphix. All rights reserved.
13  */
14
15 /*
16  * This file implements the mdb ::gcore command. The command relies on the
17  * libproc Pgc core function to actually generate the core file but we provide
18  * our own ops vector to populate data required by Pgc core. The ops vector
19  * function implementations simulate the functionality implemented by proofs.
20  * The data provided by some of the ops vector functions is not complete
21  * (missing data is documented in function headers) but there is enough
22  * information to generate a core file that can be loaded into mdb.
23  *
24  * Currently only x86 is supported. ISA-dependent functions are implemented
25  * in gcore_isadep.c.
26  * Currently only x86 is supported!
27  */
28 #ifndef _KMDB
29
30 /*
31  * The kernel has its own definition of exit which has a different signature
32  * than the user space definition. This seems to be the standard way to deal
33  * with this.
34  */
35 #define exit kern_exit
36
37 #include <mdb/mdb_modapi.h>
38 #include <mdb/mdb_param.h>
39 #include <mdb/mdb_ks.h>
40 #include <mdb/mdb_ctf.h>
41 #include <mdb/mdb_debug.h>
42 #include <mdb/mdb_gcore.h>
43
44 #include <sys/class.h>
45 #include <sys/cpuvar.h>
46 #include <sys/proc.h>
47 #include <sys/cred_impl.h>
48 #include <sys/lgrp.h>
49 #include <sys/pool.h>
50 #include <sys/project.h>
51 #include <sys/regset.h>
52 #include <sys/schedctl.h>
53 #include <sys/session.h>
54 #include <sys/syscall.h>
55 #include <sys/task.h>
56 #include <sys/var.h>
57 #include <sys/privregs.h>
58 #include <sys/psw.h>
59 #include <sys/fault.h>
60 #include <sys/proofs.h>

```

```

58 #include <sys/sysmacros.h>
59 #include <sys/wait.h>
60 #include <vm/seg.h>
61 #include <vm/vpage.h>
62 #include <fs/proc/prdata.h>
63
64 #undef exit
65
66 #include <stdio.h>
67 #include <stdbool.h>
68 #include <string.h>
69 #include <libproc.h>
70
71 #include "avl.h"
72
73 #ifdef _LP64
74 #define LSPAN(type) (P2ROUNDUP(sizeof (type), 16))
75 #else
76 #define LSPAN(type) (P2ROUNDUP(sizeof (type), 8))
77 #endif
78
79 #define vpgtob(n) ((n) * sizeof (struct vpage))
80
81 /* Macros to invoke gcore seg operations */
82 #define GSOP_INIT(_gs) (_gs)->gs_ops->gsop_init((_gs))
83 #define GSOP_FINI(_gs) (_gs)->gs_ops->gsop_fini((_gs))
84 #define GSOP_INCORE(_gs, _addr, _eaddr) \
85     (_gs)->gs_ops->gsop_inc core((_gs), (_addr), (_eaddr))
86 #define GSOP_GETPROT(_gs, _addr) \
87     (_gs)->gs_ops->gsop_getprot((_gs), (_addr))
88 #define GSOP_GETOFFSET(_gs, _addr) \
89     (_gs)->gs_ops->gsop_getoffset((_gs), (_addr))
90 #define GSOP_GETTYPE(_gs, _addr) \
91     (_gs)->gs_ops->gsop_gettype((_gs), (_addr))
92 #define GSOP_NAME(_gs, _name, _size) \
93     (_gs)->gs_ops->gsop_name((_gs), (_name), (_size))
94 #define GSOP_NORESERVE(_gs) \
95     (_gs)->gs_ops->gsop_noreserve((_gs))
96
97 #ifdef GCORE_DEBUG
98 #define dprintf(...) mdb_printf(__VA_ARGS__)
99 #else
100 #define dprintf(...)
101 #endif
102
103
104 /* mdb versions of kernel structures used for ctf read calls */
105 typedef struct mdb_proc {
106     uintptr_t p_as;
107     uintptr_t p_brkbase;
108     size_t p_brksize;
109     uintptr_t p_usrstack;
110     size_t p_stksize;
111     user_t p_user;
112     uintptr_t p_agenttp;
113     uintptr_t p_tlist;
114     uintptr_t p_zone;
115     uintptr_t p_ldt;
116     kcondvar_t p_holdlwps;
117     int p_lwpcnt;
118     uintptr_t p_lwpcdir;
119     uint_t p_lwpcdir_sz;
120     uintptr_t p_cred;
121     uint_t p_flag;
122     int p_zombcnt;
123     uintptr_t p_pidp;
124     pid_t p_ppid;

```

```

125     uintptr_t     p_pgidp;
126     uintptr_t     p_sessp;
127     uintptr_t     p_task;
128     uintptr_t     p_pool;
129     model_t       p_model;
130     char          p_wcode;
131     ushort_t      p_ldtlimit;
132     uintptr_t     p_exec;
133     uint_t        p_proc_flag;
134     ushort_t      p_pidflag;
135     k_sigset_t    p_ignore;
136     k_sigset_t    p_siginfo;
137     k_sigset_t    p_sig;
138     k_sigset_t    p_sigmask;
139     k_fltset_t    p_fltmask;
140     int           p_wdata;
141 } mdb_proc_t;

143 typedef struct mdb_kthread {
144     ushort_t      t_proc_flag;
145     uint_t        t_state;
146     lwpchan_t     t_lwpchan;
147     ushort_t      t_whystop;
148     uint8_t       t_dtrace_stop;
149     uintptr_t     t_forw;
150     uintptr_t     t_lwp;
151     id_t          t_tid;
152     short         t_sysnum;
153     pri_t         t_pri;
154     time_t        t_start;
155     id_t          t_cid;
156     uintptr_t     t_cpu;
157     int           t_bind_pset;
158     short         t_bind_cpu;
159     uintptr_t     t_lpl;
160     ushort_t      t_schedflag;
161     ushort_t      t_whatstop;
162     k_sigset_t    t_sig;
163     uintptr_t     t_schedctl;
164     k_sigset_t    t_hold;
165     hrtime_t      t_stoptime;
166 } mdb_kthread_t;

168 typedef struct mdb_seg {
169     uintptr_t     s_base;
170     size_t        s_size;
171     uintptr_t     s_ops;
172     uintptr_t     s_data;
173     uintptr_t     s_as;
174 } mdb_seg_t;

176 typedef struct mdb_as {
177     uintptr_t     a_proc;
178 } mdb_as_t;

180 typedef struct mdb_segvn_data {
181     uintptr_t     vp;
182     uint64_t      offset;
183     uint16_t      flags;
184     uint8_t       pageprot;
185     uint8_t       prot;
186     uintptr_t     amp;
187     struct vpage  *vpage;
188     uint64_t      anon_index;
189     uint8_t       type;
190 } mdb_segvn_data_t;

```

```

192 typedef struct mdb_vnode {
193     enum vtype    v_type;
194     uintptr_t     v_data;
195     uintptr_t     v_op;
196     uintptr_t     v_path;
197 } mdb_vnode_t;

199 typedef struct mdb_znode {
200     uint64_t      z_size;
201 } mdb_znode_t;

203 typedef struct mdb_tmppnode {
204     vattr_t       tn_attr;
205 } mdb_tmppnode_t;

207 typedef struct mdb_vnodeops {
208     uintptr_t     vnop_name;
209 } mdb_vnodeops_t;

211 typedef struct mdb_shm_data {
212     uintptr_t     shm_sptseg;
213 } mdb_shm_data_t;

215 typedef struct mdb_watched_page {
216     uintptr_t     wp_vaddr;
217     uint8_t       wp_oprot;
218 } mdb_watched_page_t;

220 typedef struct mdb_pid {
221     pid_t         pid_id;
222 } mdb_pid_t;

224 typedef struct mdb_sess {
225     uintptr_t     s_sidp;
226 } mdb_sess_t;

228 typedef struct mdb_task {
229     taskid_t      tk_tkid;
230     uintptr_t     tk_proj;
231 } mdb_task_t;

233 typedef struct mdb_kproject {
234     projid_t      kpj_id;
235 } mdb_kproject_t;

237 typedef struct mdb_zone {
238     zoneid_t      zone_id;
239     uintptr_t     zone_name;
240 } mdb_zone_t;

242 typedef struct mdb_sc_shared {
243     char          sc_sigblock;
244 } mdb_sc_shared_t;

246 typedef struct mdb_lwp {
247     uintptr_t     lwp_regs;
248     struct pcb    lwp_pcb;
249     uchar_t       lwp_asleep;
250     uchar_t       lwp_cursig;
251     uintptr_t     lwp_curinfo;
252     k_siginfo_t   lwp_siginfo;
253     stack_t       lwp_sigaltstack;
254     uintptr_t     lwp_oldcontext;
255     short         lwp_badpriv;
256     uintptr_t     lwp_ustack;

```

```

257     char          lwp_eosys;
258 } mdb_klwp_t;

260 typedef struct mdb_cpu {
261     processorid_t  cpu_id;
262 } mdb_cpu_t;

264 typedef struct mdb_lpl {
265     lgrp_id_t      lpl_lgrpid;
266 } mdb_lpl_t;

268 typedef struct mdb_sigqueue {
269     k_siginfo_t    sq_info;
270 } mdb_sigqueue_t;

272 typedef struct mdb_pool {
273     poolid_t       pool_id;
274 } mdb_pool_t;

276 typedef struct mdb_amp {
277     uintptr_t      ahp;
278 } mdb_amp_t;

280 typedef struct mdb_anon_hdr {
281     pgcnt_t        size;
282     uintptr_t      array_chunk;
283     int            flags;
284 } mdb_anon_hdr_t;

286 typedef struct mdb_anon {
287     uintptr_t      an_vp;
288     anoff_t        an_off;
289 } mdb_anon_t;

291 /* Used to construct a linked list of prmap_ts */
292 typedef struct prmap_node {
293     struct prmap_node *next;
294     prmap_t         m;
295 } prmap_node_t;

297 /* Fields common to psinfo_t and pstatus_t */
298 typedef struct pcommon {
299     int             pc_nlwp;
300     int             pc_nzomb;
301     pid_t           pc_pid;
302     pid_t           pc_ppid;
303     pid_t           pc_pgid;
304     pid_t           pc_sid;
305     taskid_t        pc_taskid;
306     projid_t        pc_projid;
307     zoneid_t        pc_zoneid;
308     char            pc_dmodel;
309 } pcommon_t;

311 /* AVL walk callback structures */
312 typedef struct read_maps_cbarg {
313     mdb_proc_t      *p;
314     uintptr_t        brkseg;
315     uintptr_t        stkseg;
316     prmap_node_t    *map_head;
317     prmap_node_t    *map_tail;
318     int              map_len;
319 } read_maps_cbarg_t;

321 typedef struct as_segat_cbarg {
322     uintptr_t        addr;

```

```

323     uintptr_t        res;
324 } as_segat_cbarg_t;

326 typedef struct getwatchprot_cbarg {
327     uintptr_t        wp_vaddr;
328     mdb_watched_page_t wp;
329     boolean_t        found;
330 } getwatchprot_cbarg_t;

332 struct gcore_segops;
333 typedef struct gcore_seg {
334     mdb_seg_t        *gs_seg;
335     void              *gs_data;
336     struct gcore_segops *gs_ops;
337 } gcore_seg_t;

103 /* Callback function type for processing lwp entries */
104 typedef int (*lwp_callback_t)(mdb_proc_t *, lwpent_t *, void *);

106 /* Private data */
107 static uintptr_t gcore_segvn_ops;
108 static priv_impl_info_t prinfo;
109 static sclass_t *gcore_sclass;
110 static uintptr_t gcore_kas;
111 static boolean_t gcore_initialized = B_FALSE;

113 typedef int (*gsop_init_t)(gcore_seg_t *);
114 typedef void (*gsop_fini_t)(gcore_seg_t *);
115 typedef u_offset_t (*gsop_incore_t)(gcore_seg_t *, u_offset_t, u_offset_t);
116 typedef uint_t (*gsop_getprot_t)(gcore_seg_t *, u_offset_t);
117 typedef int (*gsop_getoffset_t)(gcore_seg_t *, u_offset_t);
118 typedef void (*gsop_name_t)(gcore_seg_t *, char *name, size_t size);
119 typedef int (*gsop_gettype_t)(gcore_seg_t *, u_offset_t);
120 typedef boolean_t (*gsop_noreserve_t)(gcore_seg_t *);

122 typedef struct gcore_segops {
123     gsop_init_t      gsop_init;
124     gsop_fini_t      gsop_fini;
125     gsop_incore_t    gsop_incore;
126     gsop_getprot_t   gsop_getprot;
127     gsop_getoffset_t gsop_getoffset;
128     gsop_name_t      gsop_name;
129     gsop_gettype_t   gsop_gettype;
130     gsop_noreserve_t gsop_noreserve;
131 } gcore_segops_t;
132 unchanged portion omitted

727 /* ISA dependent function. */
728 static uintptr_t
729 gcore_prgetstackbase(mdb_proc_t *p)
730 {
731     return (p->p_usrstack - p->p_stksize);
732 }

491 static u_offset_t
492 gcore_vnode_size(uintptr_t vnode_addr)
493 {
494     mdb_vnode_t      vnode;
495     mdb_vnodeops_t   vnodeops;
496     char              vops_name[128];

498     if (mdb_ctf_vread(&vnode, "vnode_t", "mdb_vnode_t", vnode_addr, 0) ==
499         -1) {
500         return (-1);
501     }

```

```

503     if (mdb_ctf_vread(&vnnodeops, "vnnodeops_t", "mdb_vnnodeops_t",
504         vnode.v_op, 0) == -1) {
505         return (-1);
506     }

508     if (mdb_readstr(vops_name, sizeof (vops_name), vnnodeops.vnop_name) ==
509         -1) {
510         mdb_warn("Failed to read vnop_name from %p\n",
511             vnnodeops.vnop_name);
512         return (-1);
513     }

515     if (strcmp(vops_name, "zfs") == 0) {
516         mdb_znode_t znode;

518         if (mdb_ctf_vread(&znode, "znode_t", "mdb_znode_t",
519             vnode.v_data, 0) == -1) {
520             return (-1);
521         }
522         return (znode.z_size);
523     }

525     if (strcmp(vops_name, "tmpfs") == 0) {
526         mdb_tmpnode_t tnode;

528         if (mdb_ctf_vread(&tnode, "struct tmpnode", "mdb_tmpnode_t",
529             vnode.v_data, 0) == -1) {
530             return (-1);
531         }
532         return (tnode.tn_attr.va_size);
533     }

535     /* Unknown file system type. */
536     mdb_warn("Unknown fs type: %s\n", vops_name);
537     return (-1);
538 }

```

unchanged\_portion\_omitted

```

1388 /* ISA dependent function. */
1389 static int
1390 gcore_prfetchinstr(mdb_klwp_t *lwp, ulong_t *ip)
1391 {
1392     *ip = (ulong_t)(instr_t)lwp->lwp_pcb.pcb_instr;
1393     return (lwp->lwp_pcb.pcb_flags & INSTR_VALID);
1394 }

1396 /* ISA dependent function. */
1397 static int
1398 gcore_prisstep(mdb_klwp_t *lwp)
1399 {
1400     return ((lwp->lwp_pcb.pcb_flags &
1401         (NORMAL_STEP|WATCH_STEP|DEBUG_PENDING)) != 0);
1402 }

1404 /* ISA dependent function. */
1405 static void
1406 gcore_getgregs(mdb_klwp_t *lwp, gregset_t grp)
1407 {
1408     struct regs rgs;
1409     struct regs *rp;

1411     if (mdb_vread(&rgs, sizeof (rgs), lwp->lwp_regs) != sizeof (rgs)) {
1412         mdb_warn("Failed to read regs from %p\n", lwp->lwp_regs);
1413         return;
1414     }
1415     rp = &rgs;

```

```

1417 #if defined(__amd64)
1418     struct pcb *pcb = &lwp->lwp_pcb;

1420     grp[REG_RDI] = rp->r_rdi;
1421     grp[REG_RSI] = rp->r_rsi;
1422     grp[REG_RDX] = rp->r_rdx;
1423     grp[REG_RCX] = rp->r_rcx;
1424     grp[REG_R8] = rp->r_r8;
1425     grp[REG_R9] = rp->r_r9;
1426     grp[REG_RAX] = rp->r_rax;
1427     grp[REG_RBX] = rp->r_rbx;
1428     grp[REG_RBP] = rp->r_rbp;
1429     grp[REG_R10] = rp->r_r10;
1430     grp[REG_R11] = rp->r_r11;
1431     grp[REG_R12] = rp->r_r12;
1432     grp[REG_R13] = rp->r_r13;
1433     grp[REG_R14] = rp->r_r14;
1434     grp[REG_R15] = rp->r_r15;
1435     grp[REG_FSBASE] = pcb->pcb_fsbase;
1436     grp[REG_GSBASE] = pcb->pcb_gsbase;
1437     if (pcb->pcb_rupdate == 1) {
1438         grp[REG_DS] = pcb->pcb_ds;
1439         grp[REG_ES] = pcb->pcb_es;
1440         grp[REG_FS] = pcb->pcb_fs;
1441         grp[REG_GS] = pcb->pcb_gs;
1442     } else {
1443         grp[REG_DS] = rp->r_ds;
1444         grp[REG_ES] = rp->r_es;
1445         grp[REG_FS] = rp->r_fs;
1446         grp[REG_GS] = rp->r_gs;
1447     }
1448     grp[REG_TRAPNO] = rp->r_trapno;
1449     grp[REG_ERR] = rp->r_err;
1450     grp[REG_RIP] = rp->r_rip;
1451     grp[REG_CS] = rp->r_cs;
1452     grp[REG_SS] = rp->r_ss;
1453     grp[REG_RFL] = rp->r_rfl;
1454     grp[REG_RSP] = rp->r_rsp;
1455 #else
1456     bcopy(&rp->r_gs, grp, sizeof (gregset_t));
1457 #endif
1458 }

1460 /* ISA dependent functions. */
1461 static int
1462 gcore_prgetrvals(mdb_klwp_t *lwp, long *rval1, long *rval2)
1463 {
1464     struct regs *r = lwptoregs(lwp);

1466     if (r->r_ps & PS_C)
1467         return (r->r_r0);
1468     if (lwp->lwp_eosys == JUSTRETURN) {
1469         *rval1 = 0;
1470         *rval2 = 0;
1471     } else {
1472         *rval1 = r->r_r0;
1473         *rval2 = r->r_r1;
1474     }
1475     return (0);
1476 }

1145 static void
1146 gcore_prgetprregs(mdb_klwp_t *lwp, prgregset_t prp)
1147 {
1148     gcore_getgregs(lwp, prp);

```



new/usr/src/cmd/mdb/common/modules/genunix/gcore.c

9

1149 }

unchanged\_portion\_omitted

new/usr/src/cmd/mdb/intel/amd64/genunix/Makefile

1

```
*****
2282 Mon Aug 26 23:30:20 2013
new/usr/src/cmd/mdb/intel/amd64/genunix/Makefile
27908 :gcore breaks sparc build
*****
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

22 #
23 #
24 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #
27 # Copyright (c) 2013 by Delphix. All rights reserved.
28 #

30 MODULE = genunix.so
31 MDBTGT = kvm

33 include ../../../../common/modules/genunix/Makefile.files

35 COMMONSRCS = \
36     $(GENUNIX_SRCS)

38 KMODSRCS = \
39     $(COMMONSRCS)

41 MODSRCS = \
42     $(COMMONSRCS) \
43     typegraph.c

45 #
46 # This signals that $(KMODSRCS) != $(MODSRCS). Typegraph is not usable under
47 # kmdb. As such, we don't bother compiling it.
48 KMOD_SOURCES_DIFFERENT=$(POUND_SIGN)

50 include ../../../../Makefile.cmd
51 include ../../../../Makefile.cmd.64
52 include ../../Makefile.amd64
53 include ../../../../Makefile.module

55 dmod/$(MODULE) := LDLIBS += -lm -lproc

57 #
58 # We are not actually hardwiring some dependency on i86pc, we just need to
59 # include sys/param.h with _MACHDEP defined, and this forces the inclusion of
60 # machparam.h, even though we don't use anything there. This is a temporary
61 # kludge until we invent -DDONTINCLUDEMACHPARAM or something.
```

new/usr/src/cmd/mdb/intel/amd64/genunix/Makefile

2

```
62 #
63 CPPFLAGS += -I$(SRC)/uts/i86pc
64 CPPFLAGS += -I$(SRC)/uts/i86xpv

66 # Needed to include c2/audit.h (from cred.h)
67 CPPFLAGS += -I$(SRC)/uts/common

69 # Needed to find include file mutex_impl.h
70 CPPFLAGS += -I$(SRC)/uts/intel

72 CERRWARN += -_gcc=-Wno-char-subscripts
73 CERRWARN += -_gcc=-Wno-unused-label
74 CERRWARN += -_gcc=-Wno-uninitialized
75 CERRWARN += -_gcc=-Wno-parentheses
76 CERRWARN += -_gcc=-Wno-type-limits

78 LINTFLAGS64 += -erroff=E_EMPTY_TRANSLATION_UNIT

80 MODSRCS_DIR = ../../../../intel/modules/genunix
```

new/usr/src/cmd/mdb/intel/ia32/genunix/Makefile

1

```
*****
2240 Mon Aug 26 23:30:22 2013
new/usr/src/cmd/mdb/intel/ia32/genunix/Makefile
27908 :gcore breaks sparc build
*****
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

21 #
22 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
23 # Use is subject to license terms.
24 #
25 # Copyright (c) 2013 by Delphix. All rights reserved.
26 #

28 MODULE = genunix.so
29 MDBTGT = kvm

31 include ../../../../common/modules/genunix/Makefile.files

33 COMMONSRCS = \
34     $(GENUNIX_SRCS)

36 KMODSRCS = \
37     $(COMMONSRCS)

39 MODSRCS = \
40     $(COMMONSRCS) \
41     typegraph.c

43 #
44 # This signals that $(KMODSRCS) != $(MODSRCS). Typegraph is not usable under
45 # kmdb. As such, we don't bother compiling it.
46 KMOD_SOURCES_DIFFERENT=$(POUND_SIGN)

48 include ../../../../Makefile.cmd
49 include ../../Makefile.ia32
50 include ../../../../Makefile.module

52 dmod/$(MODULE) := LDLIBS += -lm -lproc

54 #
55 # We are not actually hardwiring some dependency on i86pc, we just need to
56 # include sys/param.h with _MACHDEP defined, and this forces the inclusion of
57 # machparam.h, even though we don't use anything there. This is a temporary
58 # kludge until we invent -DDONTINCLUDEMACHPARAM or something.
59 #
60 CPPFLAGS += -I$(SRC)/uts/i86pc
61 CPPFLAGS += -I$(SRC)/uts/i86xpv
```

new/usr/src/cmd/mdb/intel/ia32/genunix/Makefile

2

```
63 # Needed to include c2/audit.h (from cred.h)
64 CPPFLAGS += -I$(SRC)/uts/common

66 # Needed to find include file mutex_impl.h
67 CPPFLAGS += -I$(SRC)/uts/intel

69 CERRWARN += -_gcc=-Wno-char-subscripts
70 CERRWARN += -_gcc=-Wno-unused-label
71 CERRWARN += -_gcc=-Wno-uninitialized
72 CERRWARN += -_gcc=-Wno-parentheses
73 CERRWARN += -_gcc=-Wno-type-limits

75 LINTFLAGS += -erroff=E_EMPTY_TRANSLATION_UNIT

77 MODSRCS_DIR = ../../../../intel/modules/genunix
```

```

*****
2545 Mon Aug 26 23:30:23 2013
new/usr/src/cmd/mdb/intel/modules/genunix/gcore_isadep.c
27908 :gcore breaks sparc build
*****
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 (c) 2013 by Delphix. All rights reserved.
13 */

15 #include <mdb/mdb_modapi.h>
16 #include <mdb/mdb_gcore.h>
17 #include <mdb/mdb_debug.h>

19 #include <sys/psw.h>
20 #include <sys/privregs.h>

22 uintptr_t
23 gcore_prgetstackbase(mdb_proc_t *p)
24 {
25     return (p->p_usrstack - p->p_stksize);
26 }

28 int
29 gcore_prfetchinstr(mdb_klwp_t *lwp, ulong_t *ip)
30 {
31     *ip = (ulong_t)(instr_t)lwp->lwp_pcb.pcb_instr;
32     return (lwp->lwp_pcb.pcb_flags & INSTR_VALID);
33 }

35 int
36 gcore_prisstep(mdb_klwp_t *lwp)
37 {
38     return ((lwp->lwp_pcb.pcb_flags &
39             (NORMAL_STEP|WATCH_STEP|DEBUG_PENDING)) != 0);
40 }

42 void
43 gcore_getgregs(mdb_klwp_t *lwp, gregset_t grp)
44 {
45     struct regs rgs;
46     struct regs *rp;

48     if (mdb_vread(&rgs, sizeof (rgs), lwp->lwp_regs) != sizeof (rgs)) {
49         mdb_warn("Failed to read regs from %p\n", lwp->lwp_regs);
50         return;
51     }
52     rp = &rgs;

54 #if defined(__amd64)
55     struct pcb *pcb = &lwp->lwp_pcb;

57     grp[REG_RDI] = rp->r_rdi;
58     grp[REG_RSI] = rp->r_rsi;
59     grp[REG_RDX] = rp->r_rdx;
60     grp[REG_RCX] = rp->r_rcx;
61     grp[REG_R8] = rp->r_r8;

```

```

62     grp[REG_R9] = rp->r_r9;
63     grp[REG_RAX] = rp->r_rax;
64     grp[REG_RBX] = rp->r_rbx;
65     grp[REG_RBP] = rp->r_rbp;
66     grp[REG_R10] = rp->r_r10;
67     grp[REG_R11] = rp->r_r11;
68     grp[REG_R12] = rp->r_r12;
69     grp[REG_R13] = rp->r_r13;
70     grp[REG_R14] = rp->r_r14;
71     grp[REG_R15] = rp->r_r15;
72     grp[REG_FSBASE] = pcb->pcb_fsbase;
73     grp[REG_GSBASE] = pcb->pcb_gsbase;
74     if (pcb->pcb_rupdate == 1) {
75         grp[REG_DS] = pcb->pcb_ds;
76         grp[REG_ES] = pcb->pcb_es;
77         grp[REG_FS] = pcb->pcb_fs;
78         grp[REG_GS] = pcb->pcb_gs;
79     } else {
80         grp[REG_DS] = rp->r_ds;
81         grp[REG_ES] = rp->r_es;
82         grp[REG_FS] = rp->r_fs;
83         grp[REG_GS] = rp->r_gs;
84     }
85     grp[REG_TRAPNO] = rp->r_trapno;
86     grp[REG_ERR] = rp->r_err;
87     grp[REG_RIP] = rp->r_rip;
88     grp[REG_CS] = rp->r_cs;
89     grp[REG_SS] = rp->r_ss;
90     grp[REG_RFL] = rp->r_rfl;
91     grp[REG_RSP] = rp->r_rsp;
92 #else
93     bcopy(&rp->r_gs, grp, sizeof (gregset_t));
94 #endif
95 }

97 int
98 gcore_prgetrvals(mdb_klwp_t *lwp, long *rval1, long *rval2)
99 {
100     struct regs *r = lwptoregs(lwp);

102     if (r->r_ps & PS_C)
103         return (r->r_r0);
104     if (lwp->lwp_eosys == JUSTRETURN) {
105         *rval1 = 0;
106         *rval2 = 0;
107     } else {
108         *rval1 = r->r_r0;
109         *rval2 = r->r_r1;
110     }
111     return (0);
112 }

```

```
new/usr/src/cmd/mdb/sparc/modules/genunix/gcore_isadep.c
```

1

```
*****
```

```
903 Mon Aug 26 23:30:23 2013
```

```
new/usr/src/cmd/mdb/sparc/modules/genunix/gcore_isadep.c
```

```
27908 ::gcore breaks sparc build
```

```
*****
```

```
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 (c) 2013 by Delphix. All rights reserved.
13 */
14
15 /*
16  * ::gcore is not supported on sparc, so these functions are not
17  * implemented.
18 */
19
20 #ifndef _KMDB
21
22 #include <mdb/mdb_gcore.h>
23
24 uintptr_t
25 gcore_prgetstackbase(mdb_proc_t *p)
26 {
27 }
28
29 int
30 gcore_prfetchinstr(mdb_klwp_t *lwp, ulong_t *ip)
31 {
32 }
33
34 int
35 gcore_prisstep(mdb_klwp_t *lwp)
36 {
37 }
38
39 void
40 gcore_getgregs(mdb_klwp_t *lwp, gregset_t grp)
41 {
42 }
43
44 int
45 gcore_prgetrvals(mdb_klwp_t *lwp, long *rval1, long *rval2)
46 {
47 }
48
49 #endif /* _KMDB */
```

new/usr/src/cmd/mdb/sparc/v9/genunix/Makefile

1

```
*****
2142 Mon Aug 26 23:30:24 2013
new/usr/src/cmd/mdb/sparc/v9/genunix/Makefile
27908 :gcore breaks sparc build
*****
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

22 #
23 #
24 # Copyright 2007 Sun Microsystems, Inc. All rights reserved.
25 # Use is subject to license terms.
26 #

28 MODULE = genunix.so
29 MDBTGT = kvm

31 include ../../../../common/modules/genunix/Makefile.files

33 COMMONSRCS = \
34     $(GENUNIX_SRCS)

36 KMODSRCS = \
37     $(COMMONSRCS)

39 MODSRCS = \
40     $(COMMONSRCS) \
41     typegraph.c

43 #
44 # This signals that $(KMODSRCS) != $(MODSRCS). Typegraph is not usable under
45 # kmdb. As such, we don't bother compiling it.
46 KMOD_SOURCES_DIFFERENT=$(POUND_SIGN)

48 include ../../../../Makefile.cmd
49 include ../../../../Makefile.cmd.64
50 include ../../Makefile.sparcv9
51 include ../../../../Makefile.module

53 dmod/$(MODULE) := LDLIBS += -lm

55 #
56 # We are not actually hardwiring some dependency on sun4u, we just need to
57 # include sys/param.h with _MACHDEP defined, and this forces the inclusion of
58 # machparam.h, even though we don't use anything there. This is a temporary
59 # kludge until we invent -DDONTINCLUDEMACHPARAM or something.
60 #
61 CPPFLAGS += -I$(SRC)/uts/sun4u
```

new/usr/src/cmd/mdb/sparc/v9/genunix/Makefile

2

```
63 # Needed to include c2/audit.h (from cred.h)
64 CPPFLAGS += -I$(SRC)/uts/common
65 # Needed to find include file mutex_impl.h
66 CPPFLAGS += -I$(SRC)/uts/sparc/v9

68 CERRWARN += -_gcc=-Wno-char-subscripts
69 CERRWARN += -_gcc=-Wno-unused-label
70 CERRWARN += -_gcc=-Wno-uninitialized
71 CERRWARN += -_gcc=-Wno-parentheses
72 CERRWARN += -_gcc=-Wno-type-limits

74 MODSRCS_DIR = ../../../../sparc/modules/genunix
```

\*\*\*\*\*

9130 Mon Aug 26 23:30:25 2013

new/usr/src/lib/libproc/common/Pservice.c

27908 :gcore breaks sparc build

\*\*\*\*\*

\_\_\_\_\_unchanged\_portion\_omitted\_

```

170 #if defined(sparc) || defined(__sparc)

172 ps_err_e
173 ps_lgetxregsize(struct ps_prochandle *P, lwpid_t lwpid, int *xrsize)
174 {
175     char fname[PATH_MAX];
176     struct stat statb;

178     if (P->state == PS_DEAD) {
179         core_info_t *core = P->data;
180         lwp_info_t *lwp = list_next(&core->core_lwp_head);
181         lwp_info_t *lwp = list_next(&P->core->core_lwp_head);
182         uint_t i;

183         for (i = 0; i < core->core_nlwp; i++, lwp = list_next(lwp)) {
184             for (i = 0; i < P->core->core_nlwp; i++, lwp = list_next(lwp)) {
185                 if (lwp->lwp_id == lwpid) {
186                     if (lwp->lwp_xregs != NULL)
187                         *xrsize = sizeof (prxregset_t);
188                     else
189                         *xrsize = 0;
190                     return (PS_OK);
191                 }
192             }
193         }
194         return (PS_BADLID);
195     }

196     (void) snprintf(fname, sizeof (fname), "%s/%d/lwp/%d/xregs",
197                    procfs_path, (int)P->status.pr_pid, (int)lwpid);

199     if (stat(fname, &statb) != 0)
200         return (PS_BADLID);

202     *xrsize = (int)statb.st_size;
203     return (PS_OK);
204 }
_____unchanged_portion_omitted_

```

\*\*\*\*\*

8723 Mon Aug 26 23:30:26 2013

new/usr/src/lib/libproc/sparc/Pisadep.c

27908 ::gcore breaks sparc build

\*\*\*\*\*

unchanged portion omitted

```
184 static int
185 read_gwin(struct ps_prochandle *P, struct rwindow *rwp, uintptr_t sp)
186 {
187     gwin_query_t gq;
188
189     if (P->state == PS_DEAD) {
190         core_info_t *core = P->data;
191         lwp_info_t *lwp = list_next(&core->core_lwp_head);
192         lwp_info_t *lwp = list_next(&P->core->core_lwp_head);
193         uint_t n;
194         int i;
195         for (n = 0; n < core->core_nlwp; n++, lwp = list_next(lwp)) {
196             for (n = 0; n < P->core->core_nlwp; n++, lwp = list_next(lwp)) {
197                 gwindows_t *gwin = lwp->lwp_gwins;
198
199                 if (gwin == NULL)
200                     continue; /* No gwindows for this lwp */
201
202                 /*
203                  * If this lwp has gwindows associated with it, iterate
204                  * through the buffers looking for a stack pointer
205                  * match; if one is found, copy out the register window.
206                  */
207                 for (i = 0; i < gwin->wbcnt; i++) {
208                     if (gwin->spbuf[i] == (greg_t *)sp) {
209                         (void) memcpy(rwp, &gwin->wbuf[i],
210                                     sizeof (struct rwindow));
211                         return (0); /* We're done */
212                     }
213                 }
214             }
215             return (-1); /* No gwindows match found */
216         }
217     }
218
219     gq.gq_proc = P;
220     gq.gq_rwin = rwp;
221     gq.gq_addr = sp;
222
223     return (Plwp_iter(P, (proc_lwp_f *)find_gwin, &gq) ? 0 : -1);
224 }
```

unchanged portion omitted



```
*****
12592 Mon Aug 26 23:30:27 2013
new/usr/src/lib/libproc/sparcv9/Pisadep.c
27908 ::gcore breaks sparc build
*****
_____unchanged_portion_omitted_____

232 static int
233 read_gwin(struct ps_prochandle *P, struct rwindow *rwp, uintptr_t sp)
234 {
235     gwin_query_t gq;

237     if (P->state == PS_DEAD) {
238         core_info_t *core = P->data;
239         lwp_info_t *lwp = list_next(&core->core_lwp_head);
238         lwp_info_t *lwp = list_next(&P->core->core_lwp_head);
240         uint_t n;
241         int i;

243         for (n = 0; n < core->core_nlwp; n++, lwp = list_next(lwp)) {
242             for (n = 0; n < P->core->core_nlwp; n++, lwp = list_next(lwp)) {
244                 gwindows_t *gwin = lwp->lwp_gwins;

246                 if (gwin == NULL)
247                     continue; /* No gwindows for this lwp */

249                 /*
250                  * If this lwp has gwindows associated with it, iterate
251                  * through the buffers looking for a stack pointer
252                  * match; if one is found, copy out the register window.
253                  */
254                 for (i = 0; i < gwin->wbcnt; i++) {
255                     if (gwin->spbbuf[i] == (greg_t *)sp) {
256                         (void) memcpy(rwp, &gwin->wbuf[i],
257                                     sizeof (struct rwindow));
258                         return (0); /* We're done */
259                     }
260                 }
261             }

263             return (-1); /* No gwindows match found */

265         }

267         gq.gq_proc = P;
268         gq.gq_rwin = rwp;
269         gq.gq_addr = sp;

271         return (Plwp_iter(P, (proc_lwp_f *)find_gwin, &gq) ? 0 : -1);
272     }
_____unchanged_portion_omitted_____
```