

new/usr/src/cmd/sgs/elfdump/common/gen_struct_layout.c

1

```
*****
17264 Fri May 2 17:45:50 2014
new/usr/src/cmd/sgs/elfdump/common/gen_struct_layout.c
4839 3294 generated elfdump struct layouts by hand, shouldn't have
*****
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 */
22 /*
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25  */
26 #pragma ident "%Z%M% %I% %E% SMI"
27 #include <stdlib.h>
28 #include <stddef.h>
29 #include <stdio.h>
30 #include <string.h>
31 #include <fcntl.h>
32 #include <sys/types.h>
33 #include <sys/stat.h>
34 #include <unistd.h>
35 #include <sys/sysmacros.h>
36 #include <sys/corectl.h>
37 #include <procfs.h>
38 #include <sys/auxv.h>
39 #include <sys/old_procfs.h>
40 #include <sys/utsname.h>
44 /*
45  * This standalone program is used to generate the contents
46  * of the struct_layout_XXX.c files that contain per-architecture
47  * structure layout information.
48  *
49  * Although not part of elfdump, it is built by the makefile
50  * along with it.
51  * To use it:
52  *
53  * 1) Run it, capturing the output in a file.
54  * 2) If this is a replacement for an existing file,
55  * diff the new and old copies to ensure only
56  * the changes you expected are present.
57  * 3) Put the new file in the common directory under the name
58  * struct_layout_XXX.c, where XXX is the name of
59  * the architecture (i386, amd64, sparc, sparcv9, etc).
```

new/usr/src/cmd/sgs/elfdump/common/gen_struct_layout.c

2

```
60 * 2) Add any necessary header and copyright comments.
61 * 3) If this is a new architecture:
62 * - Add an extern statement for struct_layout_XXX()
63 * to struct_layout.h
64 * - Add a case for it to the function sl_struct_layout()
65 * in struct_layout.c.
66 */
69 /*
70  * Which machine is this build for?
71  */
72 #if defined(__i386)
74 #define MACH "i386"
76 #elif defined(__amd64)
78 #define MACH "amd64"
80 #elif defined(__sparcv9)
82 #define MACH "sparcv9"
84 #elif defined(__sparc)
86 #define MACH "sparc"
88 #else
90 #error "unrecognized build host type"
92 #endif
95 /*
96  * START and END bracket a struct layout definition. They issue
97  * the typedef boilerplate, and the standard first element (sizeof)
98  * which captures the overall size of the structure.
99  *
100 * SCALAR_FIELD is for scalar struct fields
101 *
102 * ARRAY_FIELD is for array struct fields
103 *
104 * ARRAY is for plain (non-struct) array types
105 */
106 #define START(_name, _type) \
107     (void) printf("\n\nstatic const sl_" #_name \
108     "_layout_t " #_name "_layout = {\n"); \
109     (void) printf("\t{ 0,\t%d,\t0,\t0 }\t\t/* sizeof (%s) */\n", \
110     sizeof (_type), #_type)
111 #define SCALAR_FIELD(_type, _field, _sign) \
112     (void) printf("\t{ %d,\t%d,\t0,\t%d },\t\t/* " #_field " */\n", \
113     offsetof(_type, _field), sizeof (((_type *)0)->_field), _sign)
114 #define ARRAY_FIELD(_type, _field, _sign) \
115     (void) printf("\t{ %d,\t%d,\t%d,\t%d },\t\t/* " #_field "[] */\n", \
116     offsetof(_type, _field), sizeof (((_type *)0)->_field[0]), \
117     sizeof (((_type *)0)->_field) / sizeof (((_type *)0)->_field[0]), \
118     _sign)
119 #define ARRAY(_type, _sign) \
120     (void) printf("\t{ 0,\t%d,\t%d,\t%d },\t\t/* elt0 */\n", \
121     sizeof *((_type *)0)[0]), \
122     sizeof (_type) / sizeof *((_type *)0)[0]), _sign)
123 #define END (void) printf("};\n")
```

```

126 /* auxv_t, <sys/auxv.h> */
127 static void
128 gen_auxv(void)
129 {
130     START(auxv, auxv_t);
131
132     SCALAR_FIELD(auxv_t, a_type, 1);
133     SCALAR_FIELD(auxv_t, a_un.a_val, 1);
134     SCALAR_FIELD(auxv_t, a_un.a_ptr, 0);
135     SCALAR_FIELD(auxv_t, a_un.a_fcn, 0);
136
137     END;
138 }
139
140 unchanged_portion_omitted
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157 static void
158 gen_prfdinfo(void)
159 {
160     START(prfdinfo, prfdinfo_t);
161
162     SCALAR_FIELD(prfdinfo_t, pr_fd, 0);
163     SCALAR_FIELD(prfdinfo_t, pr_mode, 0);
164     SCALAR_FIELD(prfdinfo_t, pr_uid, 0);
165     SCALAR_FIELD(prfdinfo_t, pr_gid, 0);
166     SCALAR_FIELD(prfdinfo_t, pr_major, 0);
167     SCALAR_FIELD(prfdinfo_t, pr_minor, 0);
168     SCALAR_FIELD(prfdinfo_t, pr_rmajor, 0);
169     SCALAR_FIELD(prfdinfo_t, pr_rminor, 0);
170     SCALAR_FIELD(prfdinfo_t, pr_ino, 0);
171     SCALAR_FIELD(prfdinfo_t, pr_offset, 0);
172     SCALAR_FIELD(prfdinfo_t, pr_size, 0);
173     SCALAR_FIELD(prfdinfo_t, pr_fileflags, 0);
174     SCALAR_FIELD(prfdinfo_t, pr_fdflags, 0);
175     ARRAY_FIELD(prfdinfo_t, pr_path, 0);
176
177     END;
178 }
179
180 #endif /* ! codereview */
181
182 /*ARGSUSED*/
183 int
184 main(int argc, char *argv[])
185 {
186     const char *fmt = "\t&%s_layout,\n";
187
188     printf("#include <struct_layout.h>\n");
189
190     gen_auxv();
191     gen_prgregset();
192     gen_lwpstatus();
193     gen_pstatus();
194     gen_prstatus();
195     gen_psinfo();
196     gen_prpsinfo();
197     gen_lwpsinfo();
198     gen_prcred();
199     gen_prpriv();
200     gen_priv_impl_info();
201     genfltset();
202     gensiginfo();
203     gensigset();
204     gensigaction();
205     genstack();
206     gensysset();
207     gentimestruc();

```

```

608     gen_utsname();
609     gen_prfdinfo();
610 #endif /* ! codereview */
611
612
613 /*
614  * Generate the full arch_layout description
615  */
616 (void) printf(
617     "\n\n\nstatic const sl_arch_layout_t layout_%s = {\n",
618     MACH);
619 (void) printf(fmt, "auxv");
620 (void) printf(fmt, "fltset");
621 (void) printf(fmt, "lwpsinfo");
622 (void) printf(fmt, "lwpstatus");
623 (void) printf(fmt, "prcred");
624 (void) printf(fmt, "priv_impl_info");
625 (void) printf(fmt, "prpriv");
626 (void) printf(fmt, "psinfo");
627 (void) printf(fmt, "pstatus");
628 (void) printf(fmt, "prgregset");
629 (void) printf(fmt, "prpsinfo");
630 (void) printf(fmt, "prstatus");
631 (void) printf(fmt, "sigaction");
632 (void) printf(fmt, "siginfo");
633 (void) printf(fmt, "sigset");
634 (void) printf(fmt, "stack");
635 (void) printf(fmt, "sysset");
636 (void) printf(fmt, "timestruc");
637 (void) printf(fmt, "utsname");
638 (void) printf(fmt, "prfdinfo");
639 #endif /* ! codereview */
640 (void) printf("};\n");
641
642 /*
643  * A public function, to make the information available
644  */
645 (void) printf("\n\nconst sl_arch_layout_t *\n");
646 (void) printf("struct_layout_%s(void)\n", MACH);
647 (void) printf("{\n\treturn (&layout_%s);\n}\n", MACH);
648
649 return (0);
650 }

```

new/usr/src/cmd/sgs/elfdump/common/struct_layout_amd64.c

1

```
*****
12240 Fri May 2 17:45:50 2014
new/usr/src/cmd/sgs/elfdump/common/struct_layout_amd64.c
4839 3294 generated elfdump struct layouts by hand, shouldn't have
*****
_____unchanged_portion_omitted_____
```

```
361 static const sl_prfdinfo_layout_t prfdinfo_layout = {
362     { 0, 1088, 0, 0 }, /* sizeof (prfdinfo_t) */
363     { 0, 4, 0, 0 }, /* pr_fd */
364     { 4, 4, 0, 0 }, /* pr_mode */
365     { 8, 4, 0, 0 }, /* pr_uid */
366     { 12, 4, 0, 0 }, /* pr_gid */
367     { 16, 4, 0, 0 }, /* pr_major */
368     { 20, 4, 0, 0 }, /* pr_minor */
369     { 24, 4, 0, 0 }, /* pr_rmajor */
370     { 28, 4, 0, 0 }, /* pr_rminor */
371     { 32, 8, 0, 0 }, /* pr_ino */
372     { 40, 8, 0, 0 }, /* pr_offset */
373     { 48, 8, 0, 0 }, /* pr_size */
374     { 56, 4, 0, 0 }, /* pr_fileflags */
374     { 56, 4, 0, 0 }, /* pr_fileflags */
375     { 60, 4, 0, 0 }, /* pr_fdflags */
376     { 64, 1, 1024, 0 }, /* pr_path[] */
376     { 64, 1, 1024, 0 }, /* pr_path */
377 };
```

```
382 #endif /* ! codereview */
383 static const sl_arch_layout_t layout_amd64 = {
384     &auxv_layout,
385     &fltset_layout,
386     &lwpsinfo_layout,
387     &lwpstatus_layout,
388     &prcred_layout,
389     &priv_impl_info_layout,
390     &prpriv_layout,
391     &psinfo_layout,
392     &pstatus_layout,
393     &prgregset_layout,
394     &prpsinfo_layout,
395     &prstatus_layout,
396     &sigaction_layout,
397     &siginfo_layout,
398     &sigset_layout,
399     &stack_layout,
400     &sysset_layout,
401     &timestruc_layout,
402     &utsname_layout,
403     &prfdinfo_layout,
404 };
```

```
407 const sl_arch_layout_t *
408 struct_layout_amd64(void)
409 {
410     return (&layout_amd64);
411 }
```

```

*****
12195 Fri May 2 17:45:50 2014
new/usr/src/cmd/sfs/elfdump/common/struct_layout_i386.c
4839 3294 generated elfdump struct layouts by hand, shouldn't have
*****
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 */

22 /*
23  * Copyright 2008 Sun Microsystems, Inc. All rights reserved.
24  * Use is subject to license terms.
25 */
26 /*
27  * Copyright 2012 DEY Storage Systems, Inc. All rights reserved.
28 */

30 #include <struct_layout.h>

34 #endif /* ! codereview */
35 static const sl_auxv_layout_t auxv_layout = {
36     { 0, 8, 0, 0 }, /* sizeof (auxv_t) */
37     { 0, 4, 0, 1 }, /* a_type */
38     { 4, 4, 0, 1 }, /* a_un.a_val */
39     { 4, 4, 0, 0 }, /* a_un.a_ptr */
40     { 4, 4, 0, 0 }, /* a_un.a_fcn */
41 };

44 static const sl_prgregset_layout_t prgregset_layout = {
45     { 0, 76, 0, 0 }, /* sizeof (prgregset_t) */
46     { 0, 4, 19, 0 }, /* elt0 */
47 };

50 static const sl_lwpstatus_layout_t lwpstatus_layout = {
51     { 0, 800, 0, 0 }, /* sizeof (lwpstatus_t) */
52     { 0, 4, 0, 0 }, /* pr_flags */
53     { 4, 4, 0, 0 }, /* pr_lwpid */
54     { 8, 2, 0, 0 }, /* pr_why */
55     { 10, 2, 0, 0 }, /* pr_what */
56     { 12, 2, 0, 0 }, /* pr_cursig */
57     { 16, 128, 0, 0 }, /* pr_info */
58     { 144, 16, 0, 0 }, /* pr_lwppend */
59     { 160, 16, 0, 0 }, /* pr_lwphold */
60     { 176, 32, 0, 0 }, /* pr_action */
61     { 208, 12, 0, 0 }, /* pr_altstack */

```

```

62     { 220, 4, 0, 0 }, /* pr_oldcontext */
63     { 224, 2, 0, 0 }, /* pr_syscall */
64     { 226, 2, 0, 0 }, /* pr_nsysarg */
65     { 228, 4, 0, 0 }, /* pr_errno */
66     { 232, 4, 8, 0 }, /* pr_sysarg[] */
67     { 264, 4, 0, 0 }, /* pr_rvall */
68     { 268, 4, 0, 0 }, /* pr_rval2 */
69     { 272, 1, 8, 0 }, /* pr_clname[] */
70     { 280, 8, 0, 0 }, /* pr_tstamp */
71     { 288, 8, 0, 0 }, /* pr_utime */
72     { 296, 8, 0, 0 }, /* pr_stime */
73     { 332, 4, 0, 0 }, /* pr_errpriv */
74     { 336, 4, 0, 0 }, /* pr_ustack */
75     { 340, 4, 0, 0 }, /* pr_instr */
76     { 344, 76, 0, 0 }, /* pr_reg */
77     { 420, 380, 0, 0 }, /* pr_fpreg */
78 };

81 static const sl_pstatus_layout_t pstatus_layout = {
82     { 0, 1136, 0, 0 }, /* sizeof (pstatus_t) */
83     { 0, 4, 0, 1 }, /* pr_flags */
84     { 4, 4, 0, 1 }, /* pr_nlwp */
85     { 8, 4, 0, 0 }, /* pr_pid */
86     { 12, 4, 0, 0 }, /* pr_ppid */
87     { 16, 4, 0, 0 }, /* pr_pgid */
88     { 20, 4, 0, 0 }, /* pr_sid */
89     { 24, 4, 0, 1 }, /* pr_aslwpid */
90     { 28, 4, 0, 1 }, /* pr_agentid */
91     { 32, 16, 0, 0 }, /* pr_sigpend */
92     { 48, 4, 0, 0 }, /* pr_brkbase */
93     { 52, 4, 0, 0 }, /* pr_brksize */
94     { 56, 4, 0, 0 }, /* pr_stkbase */
95     { 60, 4, 0, 0 }, /* pr_stksize */
96     { 64, 8, 0, 0 }, /* pr_utime */
97     { 72, 8, 0, 0 }, /* pr_stime */
98     { 80, 8, 0, 0 }, /* pr_cutime */
99     { 88, 8, 0, 0 }, /* pr_cstime */
100    { 96, 16, 0, 0 }, /* pr_sigtrace */
101    { 112, 16, 0, 0 }, /* pr_fltrace */
102    { 128, 64, 0, 0 }, /* pr_sysentry */
103    { 192, 64, 0, 0 }, /* pr_sysexit */
104    { 256, 1, 0, 0 }, /* pr_dmodel */
105    { 260, 4, 0, 1 }, /* pr_taskid */
106    { 264, 4, 0, 1 }, /* pr_projid */
107    { 268, 4, 0, 1 }, /* pr_nzomb */
108    { 272, 4, 0, 1 }, /* pr_zoneid */
109    { 336, 800, 0, 0 }, /* pr_lwp */
110 };

113 static const sl_prstatus_layout_t prstatus_layout = {
114     { 0, 432, 0, 0 }, /* sizeof (prstatus_t) */
115     { 0, 4, 0, 1 }, /* pr_flags */
116     { 4, 2, 0, 1 }, /* pr_why */
117     { 6, 2, 0, 1 }, /* pr_what */
118     { 8, 128, 0, 0 }, /* pr_info */
119     { 136, 2, 0, 1 }, /* pr_cursig */
120     { 138, 2, 0, 0 }, /* pr_nlwp */
121     { 140, 16, 0, 0 }, /* pr_sigpend */
122     { 156, 16, 0, 0 }, /* pr_sighold */
123     { 172, 12, 0, 0 }, /* pr_altstack */
124     { 184, 32, 0, 0 }, /* pr_action */
125     { 216, 4, 0, 0 }, /* pr_pid */
126     { 220, 4, 0, 0 }, /* pr_ppid */
127     { 224, 4, 0, 0 }, /* pr_pgrp */

```

```

128     228, 4, 0, 0, /* pr_sid */
129     232, 8, 0, 0, /* pr_utime */
130     240, 8, 0, 0, /* pr_stime */
131     248, 8, 0, 0, /* pr_cutime */
132     256, 8, 0, 0, /* pr_cstime */
133     264, 1, 8, 0, /* pr_clname[] */
134     272, 2, 0, 1, /* pr_syscall */
135     274, 2, 0, 1, /* pr_nsargs */
136     276, 4, 8, 1, /* pr_sysarg[] */
137     308, 4, 0, 0, /* pr_who */
138     312, 16, 0, 0, /* pr_lwppend */
139     328, 4, 0, 0, /* pr_oldcontext */
140     332, 4, 0, 0, /* pr_brkbase */
141     336, 4, 0, 0, /* pr_brksize */
142     340, 4, 0, 0, /* pr_stkbase */
143     344, 4, 0, 0, /* pr_stksize */
144     348, 2, 0, 1, /* pr_processor */
145     350, 2, 0, 1, /* pr_bind */
146     352, 4, 0, 1, /* pr_instr */
147     356, 76, 0, 0, /* pr_reg */
148 };

151 static const sl_psinfolayout_t psinfo_layout = {
152     0, 336, 0, 0, /* sizeof (psinfo_t) */
153     0, 4, 0, 1, /* pr_flag */
154     4, 4, 0, 1, /* pr_nlwpp */
155     8, 4, 0, 0, /* pr_pid */
156     12, 4, 0, 0, /* pr_ppid */
157     16, 4, 0, 0, /* pr_pgid */
158     20, 4, 0, 0, /* pr_sid */
159     24, 4, 0, 0, /* pr_uid */
160     28, 4, 0, 0, /* pr_euid */
161     32, 4, 0, 0, /* pr_gid */
162     36, 4, 0, 0, /* pr_egid */
163     40, 4, 0, 0, /* pr_addr */
164     44, 4, 0, 0, /* pr_size */
165     48, 4, 0, 0, /* pr_rssize */
166     56, 4, 0, 0, /* pr_ttydev */
167     60, 2, 0, 0, /* pr_pctcpu */
168     62, 2, 0, 0, /* pr_pctmem */
169     64, 8, 0, 0, /* pr_start */
170     72, 8, 0, 0, /* pr_time */
171     80, 8, 0, 0, /* pr_ctime */
172     88, 1, 16, 0, /* pr_fname[] */
173     104, 1, 80, 0, /* pr_psargs[] */
174     184, 4, 0, 1, /* pr_wstat */
175     188, 4, 0, 1, /* pr_argc */
176     192, 4, 0, 0, /* pr_argv */
177     196, 4, 0, 0, /* pr_envp */
178     200, 1, 0, 0, /* pr_dmodel */
179     204, 4, 0, 0, /* pr_taskid */
180     208, 4, 0, 0, /* pr_projid */
181     212, 4, 0, 1, /* pr_nzomb */
182     216, 4, 0, 0, /* pr_poolid */
183     220, 4, 0, 0, /* pr_zoneid */
184     224, 4, 0, 0, /* pr_contract */
185     232, 104, 0, 0, /* pr_lwp */
186 };

189 static const sl_prpsinfo_layout_t prpsinfo_layout = {
190     0, 260, 0, 0, /* sizeof (prpsinfo_t) */
191     0, 1, 0, 0, /* pr_state */
192     1, 1, 0, 0, /* pr_sname */
193     2, 1, 0, 0, /* pr_zomb */

```

```

194     3, 1, 0, 0, /* pr_nice */
195     4, 4, 0, 0, /* pr_flag */
196     8, 4, 0, 0, /* pr_uid */
197     12, 4, 0, 0, /* pr_gid */
198     16, 4, 0, 0, /* pr_pid */
199     20, 4, 0, 0, /* pr_ppid */
200     24, 4, 0, 0, /* pr_pgrp */
201     28, 4, 0, 0, /* pr_sid */
202     32, 4, 0, 0, /* pr_addr */
203     36, 4, 0, 0, /* pr_size */
204     40, 4, 0, 0, /* pr_rssize */
205     44, 4, 0, 0, /* pr_wchan */
206     48, 8, 0, 0, /* pr_start */
207     56, 8, 0, 0, /* pr_time */
208     64, 4, 0, 1, /* pr_pri */
209     68, 1, 0, 0, /* pr_oldpri */
210     69, 1, 0, 0, /* pr_cpu */
211     70, 2, 0, 0, /* pr_ottydev */
212     72, 4, 0, 0, /* pr_lttydev */
213     76, 1, 8, 0, /* pr_clname[] */
214     84, 1, 16, 0, /* pr_fname[] */
215     100, 1, 80, 0, /* pr_psargs[] */
216     180, 2, 0, 1, /* pr_syscall */
217     184, 8, 0, 0, /* pr_ctime */
218     192, 4, 0, 0, /* pr_bysize */
219     196, 4, 0, 0, /* pr_byrssize */
220     200, 4, 0, 1, /* pr_argc */
221     204, 4, 0, 0, /* pr_argv */
222     208, 4, 0, 0, /* pr_envp */
223     212, 4, 0, 1, /* pr_wstat */
224     216, 2, 0, 0, /* pr_pctcpu */
225     218, 2, 0, 0, /* pr_pctmem */
226     220, 4, 0, 0, /* pr_euid */
227     224, 4, 0, 0, /* pr_egid */
228     228, 4, 0, 0, /* pr_aslwpid */
229     232, 1, 0, 0, /* pr_dmodel */
230 };

233 static const sl_lwpsinfo_layout_t lwpsinfo_layout = {
234     0, 104, 0, 0, /* sizeof (lwpsinfo_t) */
235     0, 4, 0, 1, /* pr_flag */
236     4, 4, 0, 0, /* pr_lwpid */
237     8, 4, 0, 0, /* pr_addr */
238     12, 4, 0, 0, /* pr_wchan */
239     16, 1, 0, 0, /* pr_stype */
240     17, 1, 0, 0, /* pr_state */
241     18, 1, 0, 0, /* pr_sname */
242     19, 1, 0, 0, /* pr_nice */
243     20, 2, 0, 0, /* pr_syscall */
244     22, 1, 0, 0, /* pr_oldpri */
245     23, 1, 0, 0, /* pr_cpu */
246     24, 4, 0, 1, /* pr_pri */
247     28, 2, 0, 0, /* pr_pctcpu */
248     32, 8, 0, 0, /* pr_start */
249     40, 8, 0, 0, /* pr_time */
250     48, 1, 8, 0, /* pr_clname[] */
251     56, 1, 16, 0, /* pr_name[] */
252     72, 4, 0, 1, /* pr_onpro */
253     76, 4, 0, 1, /* pr_bindpro */
254     80, 4, 0, 1, /* pr_bindpset */
255     84, 4, 0, 1, /* pr_lgrp */
256 };

259 static const sl_prcred_layout_t prcred_layout = {

```

```

260     { 0, 32, 0, 0 }, /* sizeof (prcred_t) */
261     { 0, 4, 0, 0 }, /* pr_euid */
262     { 4, 4, 0, 0 }, /* pr_ruid */
263     { 8, 4, 0, 0 }, /* pr_suid */
264     { 12, 4, 0, 0 }, /* pr_egid */
265     { 16, 4, 0, 0 }, /* pr_rgid */
266     { 20, 4, 0, 0 }, /* pr_sgid */
267     { 24, 4, 0, 1 }, /* pr_ngroups */
268     { 28, 4, 1, 0 }, /* pr_groups[] */
269 };

272 static const sl_prpriv_layout_t prpriv_layout = {
273     { 0, 16, 0, 0 }, /* sizeof (prpriv_t) */
274     { 0, 4, 0, 0 }, /* pr_nsets */
275     { 4, 4, 0, 0 }, /* pr_setsize */
276     { 8, 4, 0, 0 }, /* pr_info_size */
277     { 12, 4, 1, 0 }, /* pr_sets[] */
278 };

281 static const sl_priv_impl_info_layout_t priv_impl_info_layout = {
282     { 0, 28, 0, 0 }, /* sizeof (priv_impl_info_t) */
283     { 0, 4, 0, 0 }, /* priv_headersize */
284     { 4, 4, 0, 0 }, /* priv_flags */
285     { 8, 4, 0, 0 }, /* priv_nsets */
286     { 12, 4, 0, 0 }, /* priv_setsize */
287     { 16, 4, 0, 0 }, /* priv_max */
288     { 20, 4, 0, 0 }, /* priv_info_size */
289     { 24, 4, 0, 0 }, /* priv_globalinfo_size */
290 };

293 static const sl_fltset_layout_t fltset_layout = {
294     { 0, 16, 0, 0 }, /* sizeof (fltset_t) */
295     { 0, 4, 4, 0 }, /* word[] */
296 };

299 static const sl_siginfo_layout_t siginfo_layout = {
300     { 0, 128, 0, 0 }, /* sizeof (siginfo_t) */
301     { 0, 4, 0, 0 }, /* si_signo */
302     { 8, 4, 0, 0 }, /* si_errno */
303     { 4, 4, 0, 1 }, /* si_code */
304     { 20, 4, 0, 0 }, /* si_value.sival_int */
305     { 20, 4, 0, 0 }, /* si_value.sival_ptr */
306     { 12, 4, 0, 0 }, /* si_pid */
307     { 16, 4, 0, 0 }, /* si_uid */
308     { 28, 4, 0, 0 }, /* si_ctid */
309     { 32, 4, 0, 0 }, /* si_zoneid */
310     { 12, 4, 0, 0 }, /* si_entity */
311     { 12, 4, 0, 0 }, /* si_addr */
312     { 20, 4, 0, 0 }, /* si_status */
313     { 16, 4, 0, 0 }, /* si_band */
314 };

317 static const sl_sigset_layout_t sigset_layout = {
318     { 0, 16, 0, 0 }, /* sizeof (sigset_t) */
319     { 0, 4, 4, 0 }, /* __sigbits[] */
320 };

323 static const sl_sigaction_layout_t sigaction_layout = {
324     { 0, 32, 0, 0 }, /* sizeof (struct sigaction) */
325     { 0, 4, 0, 0 }, /* sa_flags */

```

```

326     { 4, 4, 0, 0 }, /* sa_handler */
327     { 4, 4, 0, 0 }, /* sa_sigaction */
328     { 8, 16, 0, 0 }, /* sa_mask */
329 };

332 static const sl_stack_layout_t stack_layout = {
333     { 0, 12, 0, 0 }, /* sizeof (stack_t) */
334     { 0, 4, 0, 0 }, /* ss_sp */
335     { 4, 4, 0, 0 }, /* ss_size */
336     { 8, 4, 0, 0 }, /* ss_flags */
337 };

340 static const sl_sysset_layout_t sysset_layout = {
341     { 0, 64, 0, 0 }, /* sizeof (sysset_t) */
342     { 0, 4, 16, 0 }, /* word[] */
343 };

346 static const sl_timestruc_layout_t timestruc_layout = {
347     { 0, 8, 0, 0 }, /* sizeof (timestruc_t) */
348     { 0, 4, 0, 0 }, /* tv_sec */
349     { 4, 4, 0, 0 }, /* tv_nsec */
350 };

353 static const sl_utsname_layout_t utsname_layout = {
354     { 0, 1285, 0, 0 }, /* sizeof (struct utsname) */
355     { 0, 1, 257, 0 }, /* sysname[] */
356     { 257, 1, 257, 0 }, /* nodename[] */
357     { 514, 1, 257, 0 }, /* release[] */
358     { 771, 1, 257, 0 }, /* version[] */
359     { 1028, 1, 257, 0 }, /* machine[] */
360 };

363 static const sl_prfdinfo_layout_t prfdinfo_layout = {
364     { 0, 1088, 0, 0 }, /* sizeof (prfdinfo_t) */
365     { 0, 4, 0, 0 }, /* pr_fd */
366     { 4, 4, 0, 0 }, /* pr_mode */
367     { 8, 4, 0, 0 }, /* pr_uid */
368     { 12, 4, 0, 0 }, /* pr_gid */
369     { 16, 4, 0, 0 }, /* pr_major */
370     { 20, 4, 0, 0 }, /* pr_minor */
371     { 24, 4, 0, 0 }, /* pr_rmajor */
372     { 28, 4, 0, 0 }, /* pr_rminor */
373     { 32, 8, 0, 0 }, /* pr_ino */
374     { 40, 8, 0, 0 }, /* pr_offset */
375     { 48, 8, 0, 0 }, /* pr_size */
376     { 56, 4, 0, 0 }, /* pr_fileflags */
377     { 56, 4, 0, 0 }, /* pr_filefags */
378     { 60, 4, 0, 0 }, /* pr_fdflags */
379     { 64, 1, 1024, 0 }, /* pr_path[] */
380     { 64, 1, 1024, 0 }, /* pr_path */
381 };

384 #endif /* ! codereview */
385 static const sl_arch_layout_t layout_i386 = {
386     &auxv_layout,
387     &fltset_layout,
388     &lwpsinfo_layout,
389     &lwpsstatus_layout,

```

```
390     &prcred_layout,  
391     &priv_impl_info_layout,  
392     &prpriv_layout,  
393     &psinfo_layout,  
394     &pstatus_layout,  
395     &prgregset_layout,  
396     &prpsinfo_layout,  
397     &prstatus_layout,  
398     &sigaction_layout,  
399     &siginfo_layout,  
400     &sigset_layout,  
401     &stack_layout,  
402     &sysset_layout,  
403     &timestruc_layout,  
404     &utsname_layout,  
405     &prfdinfo_layout,  
406 };
```

```
409 const sl_arch_layout_t *  
410 struct_layout_i386(void)  
411 {  
412     return (&layout_i386);  
413 }
```

new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparc.c

1

12197 Fri May 2 17:45:51 2014

new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparc.c

4839 3294 generated elfdump struct layouts by hand, shouldn't have

_____unchanged_portion_omitted_

```
360 static const sl_prfdinfo_layout_t prfdinfo_layout = {
361     { 0, 1088, 0, 0 }, /* sizeof (prfdinfo_t) */
362     { 0, 4, 0, 0 }, /* pr_fd */
363     { 4, 4, 0, 0 }, /* pr_mode */
364     { 8, 4, 0, 0 }, /* pr_uid */
365     { 12, 4, 0, 0 }, /* pr_gid */
366     { 16, 4, 0, 0 }, /* pr_major */
367     { 20, 4, 0, 0 }, /* pr_minor */
368     { 24, 4, 0, 0 }, /* pr_rmajor */
369     { 28, 4, 0, 0 }, /* pr_rminor */
370     { 32, 8, 0, 0 }, /* pr_ino */
371     { 40, 8, 0, 0 }, /* pr_offset */
372     { 48, 8, 0, 0 }, /* pr_size */
373     { 56, 4, 0, 0 }, /* pr_fileflags */
373     { 56, 4, 0, 0 }, /* pr_fileflags */
374     { 60, 4, 0, 0 }, /* pr_fdflags */
375     { 64, 1, 1024, 0 }, /* pr_path[] */
375     { 64, 1, 1024, 0 }, /* pr_path */
376 };
```

_____unchanged_portion_omitted_

new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparcv9.c

1

12244 Fri May 2 17:45:51 2014

new/usr/src/cmd/sgs/elfdump/common/struct_layout_sparcv9.c

4839 3294 generated elfdump struct layouts by hand, shouldn't have

_____unchanged_portion_omitted_

```
361 static const sl_prfdinfo_layout_t prfdinfo_layout = {
362     { 0, 1088, 0, 0 }, /* sizeof (prfdinfo_t) */
363     { 0, 4, 0, 0 }, /* pr_fd */
364     { 4, 4, 0, 0 }, /* pr_mode */
365     { 8, 4, 0, 0 }, /* pr_uid */
366     { 12, 4, 0, 0 }, /* pr_gid */
367     { 16, 4, 0, 0 }, /* pr_major */
368     { 20, 4, 0, 0 }, /* pr_minor */
369     { 24, 4, 0, 0 }, /* pr_rmajor */
370     { 28, 4, 0, 0 }, /* pr_rminor */
371     { 32, 8, 0, 0 }, /* pr_ino */
372     { 40, 8, 0, 0 }, /* pr_offset */
373     { 48, 8, 0, 0 }, /* pr_size */
374     { 56, 4, 0, 0 }, /* pr_fileflags */
374     { 56, 4, 0, 0 }, /* pr_fileflags */
375     { 60, 4, 0, 0 }, /* pr_fdflags */
376     { 64, 1, 1024, 0 }, /* pr_path[] */
376     { 64, 1, 1024, 0 }, /* pr_path */
377 };
```

_____unchanged_portion_omitted_