

```

*****
2722 Tue Jul 2 20:55:09 2013
new/usr/src/head/malloc.h
3853 __cplusplus change and headers incompatibility with clang
*****
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, Version 1.0 only
6  * (the "License"). You may not use this file except in compliance
7  * with the License.
8  *
9  * You can obtain a copy of the license at usr/src/OPENSOLARIS.LICENSE
10 * or http://www.opensolaris.org/os/licensing.
11 * See the License for the specific language governing permissions
12 * and limitations under the License.
13 *
14 * When distributing Covered Code, include this CDDL HEADER in each
15 * file and include the License file at usr/src/OPENSOLARIS.LICENSE.
16 * If applicable, add the following below this CDDL HEADER, with the
17 * fields enclosed by brackets "[]" replaced with your own identifying
18 * information: Portions Copyright [yyyy] [name of copyright owner]
19 *
20 * CDDL HEADER END
21 */
22 /*      Copyright (c) 1988 AT&T */
23 /*      All Rights Reserved */

26 #ifndef _MALLOC_H
27 #define _MALLOC_H

29 #pragma ident      "%Z%M% %I%      %E% SMI"      /* SVr4.0 1.7 */

29 #include <sys/types.h>

31 #ifdef __cplusplus
32 extern "C" {
33 #endif

35 /*
36  *      Constants defining mallopt operations
37  */
38 #define M_MXFAST      1      /* set size of blocks to be fast */
39 #define M_NLBLKS      2      /* set number of block in a holding block */
40 #define M_GRAIN      3      /* set number of sizes mapped to one, for */
41                               /* small blocks */
42 #define M_KEEP      4      /* retain contents of block after a free */
43                               /* until another allocation */
44 /*
45  *      structure filled by
46  */
47 struct mallinfo {
48     unsigned long arena;      /* total space in arena */
49     unsigned long ordblks;    /* number of ordinary blocks */
50     unsigned long smlbks;    /* number of small blocks */
51     unsigned long hblks;     /* number of holding blocks */
52     unsigned long hblkhd;    /* space in holding block headers */
53     unsigned long usmlbks;    /* space in small blocks in use */
54     unsigned long fsmblks;    /* space in free small blocks */
55     unsigned long uordblks;   /* space in ordinary blocks in use */
56     unsigned long fordblks;   /* space in free ordinary blocks */
57     unsigned long keepcost;   /* cost of enabling keep option */
58 };

```

```

60 #if defined(__STDC__)

62 #if (!defined(_STRICT_STDC) && !defined(_XOPEN_OR_POSIX)) || \
63     defined(_XPG3)
64 #if __cplusplus >= 199711L
65 namespace std {
66 #endif

68 #endif /* ! codereview */
69 void *malloc(size_t);
70 void free(void *);
71 void *realloc(void *, size_t);
72 void *calloc(size_t, size_t);

74 #if __cplusplus >= 199711L
75 } /* end of namespace std */

77 using std::malloc;
78 using std::free;
79 using std::realloc;
80 using std::calloc;
81 #endif /* __cplusplus >= 199711L */
82 #endif /* (!defined(_STRICT_STDC) && !defined(_XOPEN_OR_POSIX)) || ... */

84 #endif /* ! codereview */
85 int mallopt(int, int);
86 struct mallinfo mallinfo(void);
64 void *calloc(size_t, size_t);

88 #else

90 void *malloc();
91 void free();
92 void *realloc();
93 void *calloc();
94 #endif /* ! codereview */
95 int mallopt();
96 struct mallinfo mallinfo();
71 void *calloc();

98 #endif /* __STDC__ */

100 #ifdef __cplusplus
101 }
_____unchanged_portion_omitted_____

```

new/usr/src/head/unistd.h

1

```
*****
38295 Tue Jul  2 20:55:12 2013
new/usr/src/head/unistd.h
3853 __cplusplus change and headers incompatibility with clang
*****
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 (c) 1989, 2010, Oracle and/or its affiliates. All rights reserved.
24 */

26 /*      Copyright (c) 1988 AT&T */
27 /*      All Rights Reserved      */

29 /* Copyright (c) 2013, OmniTI Computer Consulting, Inc. All rights reserved. */

31 #ifndef _UNISTD_H
32 #define _UNISTD_H

34 #include <sys/feature_tests.h>

36 #include <sys/types.h>
37 #include <sys/unistd.h>

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

43 /* Symbolic constants for the "access" routine: */
44 #define R_OK  4 /* Test for Read permission */
45 #define W_OK  2 /* Test for Write permission */
46 #define X_OK  1 /* Test for eXecute permission */
47 #define F_OK  0 /* Test for existence of File */

49 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
50 #define F_ULOCK 0 /* Unlock a previously locked region */
51 #define F_LOCK  1 /* Lock a region for exclusive use */
52 #define F_TLOCK 2 /* Test and lock a region for exclusive use */
53 #define F_TEST  3 /* Test a region for other processes locks */
54 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2)... */

56 /* Symbolic constants for the "lseek" routine: */

58 #ifndef SEEK_SET
59 #define SEEK_SET 0 /* Set file pointer to "offset" */
60 #endif
```

new/usr/src/head/unistd.h

2

```
62 #ifndef SEEK_CUR
63 #define SEEK_CUR 1 /* Set file pointer to current plus "offset" */
64 #endif

66 #ifndef SEEK_END
67 #define SEEK_END 2 /* Set file pointer to EOF plus "offset" */
68 #endif

70 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
71 #ifndef SEEK_DATA
72 #define SEEK_DATA 3 /* Set file pointer to next data past offset */
73 #endif

75 #ifndef SEEK_HOLE
76 #define SEEK_HOLE 4 /* Set file pointer to next hole past offset */
77 #endif
78 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

80 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
81 /* Path names: */
82 #define GF_PATH "/etc/group" /* Path name of the "group" file */
83 #define PF_PATH "/etc/passwd" /* Path name of the "passwd" file */
84 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

86 /*
87  * compile-time symbolic constants,
88  * Support does not mean the feature is enabled.
89  * Use pathconf/sysconf to obtain actual configuration value.
90 */

92 /* Values unchanged in UNIX 03 */
93 #define _POSIX_ASYNC_IO 1
94 #define _POSIX_JOB_CONTROL 1
95 #define _POSIX_SAVED_IDS 1
96 #define _POSIX_SYNC_IO 1

98 /*
99  * POSIX.1b compile-time symbolic constants.
100 */
101 #if defined(__XPG6)
102 #define _POSIX_ASYNCHRONOUS_IO 200112L
103 #define _POSIX_FSYNC 200112L
104 #define _POSIX_MAPPED_FILES 200112L
105 #define _POSIX_MEMLOCK 200112L
106 #define _POSIX_MEMLOCK_RANGE 200112L
107 #define _POSIX_MEMORY_PROTECTION 200112L
108 #define _POSIX_MESSAGE_PASSING 200112L
109 #define _POSIX_PRIORITY_SCHEDULING 200112L
110 #define _POSIX_REALTIME_SIGNALS 200112L
111 #define _POSIX_SEMAPHORES 200112L
112 #define _POSIX_SHARED_MEMORY_OBJECTS 200112L
113 #define _POSIX_SYNCHRONIZED_IO 200112L
114 #else
115 #define _POSIX_ASYNCHRONOUS_IO 1
116 #define _POSIX_FSYNC 1
117 #define _POSIX_MAPPED_FILES 1
118 #define _POSIX_MEMLOCK 1
119 #define _POSIX_MEMLOCK_RANGE 1
120 #define _POSIX_MEMORY_PROTECTION 1
121 #define _POSIX_MESSAGE_PASSING 1
122 #define _POSIX_PRIORITY_SCHEDULING 1
123 #define _POSIX_REALTIME_SIGNALS 1
124 #define _POSIX_SEMAPHORES 1
125 #define _POSIX_SHARED_MEMORY_OBJECTS 1
126 #define _POSIX_SYNCHRONIZED_IO 1
127 #endif
```

```

129 /*
130 * POSIX.1c compile-time symbolic constants.
131 */
132 #if defined(_XPG6)
133 #define _POSIX_THREAD_SAFE_FUNCTIONS      200112L
134 #define _POSIX_THREADS                    200112L
135 #define _POSIX_THREAD_ATTR_STACKADDR     200112L
136 #define _POSIX_THREAD_ATTR_STACKSIZE    200112L
137 #define _POSIX_THREAD_PROCESS_SHARED     200112L
138 #define _POSIX_THREAD_PRIORITY_SCHEDULING 200112L
139 #define _POSIX_TIMERS                     200112L
140 #else
141 #define _POSIX_THREAD_SAFE_FUNCTIONS      1
142 #define _POSIX_THREADS                    1
143 #define _POSIX_THREAD_ATTR_STACKADDR     1
144 #define _POSIX_THREAD_ATTR_STACKSIZE    1
145 #define _POSIX_THREAD_PROCESS_SHARED     1
146 #define _POSIX_THREAD_PRIORITY_SCHEDULING 1
147 #define _POSIX_TIMERS                     1
148 #endif

150 /* New in UNIX 03 */
151 #define _POSIX_ADVISORY_INFO              200112L
152 #define _POSIX_BARRIERS                   200112L
153 #define _POSIX_CLOCK_SELECTION           200112L
154 #define _POSIX_IPV6                       200112L
155 #define _POSIX_MONOTONIC_CLOCK           200112L
156 #define _POSIX_RAW_SOCKETS               200112L
157 #define _POSIX_READER_WRITER_LOCKS      200112L
158 #define _POSIX_SPAWN                      200112L
159 #define _POSIX_SPIN_LOCKS                200112L
160 #define _POSIX_TIMEOUTS                   200112L

162 /*
163 * Support for the POSIX.1 mutex protocol attribute. For realtime applications
164 * which need mutexes to support priority inheritance/ceiling.
165 */
166 #if defined(_XPG6)
167 #define _POSIX_THREAD_PRIO_INHERIT       200112L
168 #define _POSIX_THREAD_PRIO_PROTECT      200112L
169 #else
170 #define _POSIX_THREAD_PRIO_INHERIT       1
171 #define _POSIX_THREAD_PRIO_PROTECT      1
172 #endif

174 #ifndef _POSIX_VDISABLE
175 #define _POSIX_VDISABLE                  0
176 #endif

178 #ifndef NULL
179 #if defined(_LP64)
180 #define NULL 0L
181 #else
182 #define NULL 0
183 #endif
184 #endif

186 #define STDIN_FILENO 0
187 #define STDOUT_FILENO 1
188 #define STDERR_FILENO 2

190 /*
191 * Large File Summit-related announcement macros. The system supports both
192 * the additional and transitional Large File Summit interfaces. (The final
193 * two macros provide a finer granularity breakdown of _LFS64_LARGEFILE.)

```

```

194 */
195 #define _LFS_LARGEFILE 1
196 #define _LFS64_LARGEFILE 1
197 #define _LFS64_STDIO 1
198 #define _LFS64_ASYNCHRONOUS_IO 1

200 /* large file compilation environment setup */
201 #if !defined(_LP64) && _FILE_OFFSET_BITS == 64
202 #ifdef __PRAGMA_REDEFINE_EXTNAME
203 #pragma redefine_extname ftruncate ftruncate64
204 #pragma redefine_extname lseek lseek64
205 #pragma redefine_extname pread pread64
206 #pragma redefine_extname pwrite pwrite64
207 #pragma redefine_extname truncate truncate64
208 #pragma redefine_extname lockf lockf64
209 #pragma redefine_extname tell tell64
210 #else /* __PRAGMA_REDEFINE_EXTNAME */
211 #define ftruncate ftruncate64
212 #define lseek lseek64
213 #define pread pread64
214 #define pwrite pwrite64
215 #define truncate truncate64
216 #define lockf lockf64
217 #define tell tell64
218 #endif /* __PRAGMA_REDEFINE_EXTNAME */
219 #endif /* !_LP64 && _FILE_OFFSET_BITS == 64 */

221 /* In the LP64 compilation environment, the APIs are already large file */
222 #if defined(_LP64) && defined(_LARGEFILE64_SOURCE)
223 #ifdef __PRAGMA_REDEFINE_EXTNAME
224 #pragma redefine_extname ftruncate64 ftruncate
225 #pragma redefine_extname lseek64 lseek
226 #pragma redefine_extname pread64 pread
227 #pragma redefine_extname pwrite64 pwrite
228 #pragma redefine_extname truncate64 truncate
229 #pragma redefine_extname lockf64 lockf
230 #pragma redefine_extname tell64 tell
231 #else /* __PRAGMA_REDEFINE_EXTNAME */
232 #define ftruncate64 ftruncate
233 #define lseek64 lseek
234 #define pread64 pread
235 #define pwrite64 pwrite
236 #define truncate64 truncate
237 #define lockf64 lockf
238 #define tell64 tell
239 #endif /* __PRAGMA_REDEFINE_EXTNAME */
240 #endif /* _LP64 && _LARGEFILE64_SOURCE */

242 #if defined(__STDC__)

244 extern int access(const char *, int);
245 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
246 extern int acct(const char *);
247 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
248 extern unsigned alarm(unsigned);
249 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
250 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
251     defined(__EXTENSIONS__)
252 extern int brk(void *);
253 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
254 extern int chdir(const char *);
255 extern int chown(const char *, uid_t, gid_t);
256 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
257 #if !defined(_POSIX_C_SOURCE) || (defined(_XOPEN_SOURCE) && \
258     !defined(_XPG6)) || defined(__EXTENSIONS__)
259 extern int chroot(const char *);

```

```

260 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
261 extern int close(int);
262 #if defined(_XPG4) || defined(__EXTENSIONS__)
263 extern size_t confstr(int, char *, size_t);
264 extern char *crypt(const char *, const char *);
265 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
266 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
267     defined(__EXTENSIONS__)
268 extern char *ctermid(char *);
269 #endif /* (!defined(_POSIX_C_SOURCE) ... */
270 #if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
271 extern char *ctermid_r(char *);
272 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
273 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
274 #if !defined(_XPG6) || defined(__EXTENSIONS__)
275 extern char *cuserid(char *);
276 #endif
277 extern int dup(int);
278 extern int dup2(int, int);
279 extern int dup3(int, int, int);
280 #if defined(_XPG4) || defined(__EXTENSIONS__)
281 extern void encrypt(char *, int);
282 #endif /* defined(XPG4) || defined(__EXTENSIONS__) */
283 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
284 extern void endusershell(void);
285 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
286 extern int execl(const char *, const char *, ...);
287 extern int execlp(const char *, const char *, ...);
288 extern int execlp(const char *, const char *, ...);
289 extern int execv(const char *, char *const *);
290 extern int execve(const char *, char *const *, char *const *);
291 extern int execvp(const char *, char *const *);
292 extern void _exit(int);
293     _NORETURN;
294 /*
295  * The following fattach prototype is duplicated in <stropts.h>. The
296  * duplication is necessitated by XPG4.2 which requires the prototype
297  * be defined in <stropts.h>.
298  */
299 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
300 extern int fattach(int, const char *);
301 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
302 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
303 extern int fchdir(int);
304 extern int fchown(int, uid_t, gid_t);
305 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
306 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
307 extern int fchroot(int);
308 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
309 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
310     defined(__EXTENSIONS__)
311 extern int fdatsync(int);
312 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
313 /*
314  * The following fdetach prototype is duplicated in <stropts.h>. The
315  * duplication is necessitated by XPG4.2 which requires the prototype
316  * be defined in <stropts.h>.
317  */
318 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
319 extern int fdetach(const char *);
320 #endif /* !defined(__XOPEN_OR_POSIX)... */
321 extern pid_t fork(void);
322 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
323 extern pid_t fork1(void);
324 extern pid_t forkall(void);
325 #endif /* !defined(__XOPEN_OR_POSIX)... */

```

```

326 extern long fpathconf(int, int);
327 #if !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2) || \
328     defined(__EXTENSIONS__)
329 extern int fsync(int);
330 #endif /* !defined(_POSIX_C_SOURCE) || (_POSIX_C_SOURCE > 2)... */
331 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || defined(_XPG4_2) || \
332     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
333     defined(__EXTENSIONS__)
334 extern int ftruncate(int, off_t);
335 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
336 extern char *getcwd(char *, size_t);
337 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
338     defined(__EXTENSIONS__)
339 extern int getdtablesize(void);
340 #endif
341 extern gid_t getegid(void);
342 extern uid_t geteuid(void);
343 extern gid_t getgid(void);
344 extern int getgroups(int, gid_t *);
345 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
346 extern long gethostid(void);
347 #endif
348 #if defined(_XPG4_2)
349 extern int gethostname(char *, size_t);
350 #elif !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
351 extern int gethostname(char *, int);
352 #endif
353 extern char *getlogin(void);
354 #if defined(_XPG4) || defined(__EXTENSIONS__)
355 extern int getopt(int, char *const *, const char *);
356 extern char *optarg;
357 extern int opterr, optind, optopt;
358 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
359 #if !defined(_XPG6) || defined(__EXTENSIONS__)
360 extern char *getpass(const char *);
361 #endif
362 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
363 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
364 /* Marked as LEGACY in SUSv2 and removed in SUSv3 */
365 #if !defined(_XPG6) || defined(__EXTENSIONS__)
366 extern int getpagesize(void);
367 #endif
368 extern pid_t getpgid(pid_t);
369 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
370 extern pid_t getpid(void);
371 extern pid_t getppid(void);
372 extern pid_t getpgrp(void);

374 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
375 char *gettext(const char *, const char *);
376 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
377 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
378 extern pid_t getsid(pid_t);
379 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
380 extern uid_t getuid(void);
381 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
382 extern char *getusershell(void);
383 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
384 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
385 extern char *getwd(char *);
386 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
387 /*
388  * The following ioctl prototype is duplicated in <stropts.h>. The
389  * duplication is necessitated by XPG4.2 which requires the prototype
390  * be defined in <stropts.h>.
391  */

```

```

392 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
393 extern int ioctl(int, int, ...);
394 extern int isaexec(const char *, char *const *, char *const *);
395 extern int issetugid(void);
396 #endif
397 extern int isatty(int);
398 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
399 extern int lchown(const char *, uid_t, gid_t);
400 #endif
401 extern int link(const char *, const char *);
402 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
403 extern off_t llseek(int, off_t, int);
404 #endif
405 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
406     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
407     defined(__EXTENSIONS__)
408 extern int lockf(int, int, off_t);
409 #endif
410 extern off_t lseek(int, off_t, int);
411 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
412     defined(__EXTENSIONS__)
413 extern int nice(int);
414 #endif /* !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
415 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
416 extern int mincore(caddr_t, size_t, char *);
417 #endif
418 extern long pathconf(const char *, int);
419 extern int pause(void);
420 extern int pipe(int *);
421 extern int pipe2(int *, int);
422 #if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
423     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
424     defined(__EXTENSIONS__)
425 extern ssize_t pread(int, void *, size_t, off_t);
426 #endif
427 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
428 extern void profil(unsigned short *, size_t, unsigned long, unsigned int);
429 #endif
430 /*
431  * pthread_atfork() is also declared in <pthread.h> as per SUSv3. The
432  * declarations are identical. A change to either one may also require
433  * appropriate namespace updates in order to avoid redeclaration
434  * warnings in the case where both prototypes are exposed via inclusion
435  * of both <pthread.h> and <unistd.h>.
436  */
437 #if !defined(__XOPEN_OR_POSIX) || \
438     ((_POSIX_C_SOURCE > 2) && !defined(_XPG6)) || \
439     defined(__EXTENSIONS__)
440 extern int pthread_atfork(void (*) (void), void (*) (void), void (*) (void));
441 #endif /* !defined(__XOPEN_OR_POSIX) || ((_POSIX_C_SOURCE > 2) ... */
442 #if !defined(_LP64) && \
443     (defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__))
444 extern int ptrace(int, pid_t, int, int);
445 #endif
446 #if !defined(_POSIX_C_SOURCE) || defined(_XPG5) || \
447     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
448     defined(__EXTENSIONS__)
449 extern ssize_t pwrite(int, const void *, size_t, off_t);
450 #endif
451 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
452 /* per RFC 3542; This is also defined in netdb.h */
453 extern int rcmd_af(char **, unsigned short, const char *, const char *,
454     const char *, int *, int);
455 #endif
456 extern ssize_t read(int, void *, size_t);
457 #if !defined(__XOPEN_OR_POSIX) || \

```

```

458     defined(_XPG4_2) || defined(__EXTENSIONS__)
459 extern ssize_t readlink(const char *_RESTRICT_KYWD, char *_RESTRICT_KYWD,
460     size_t);
461 #endif
462 #if (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3) && !defined(_XPG4))) || \
463     defined(__EXTENSIONS__)
464 #if __cplusplus >= 199711L
465 namespace std {
466 #endif
467 #endif /* !codereview */
468 extern int rename(const char *, const char *);
469 #if __cplusplus >= 199711L
470 } /* end of namespace std */
471 #endif
472 using std::rename;
473 #endif /* __cplusplus >= 199711L */
474 #endif /* !codereview */
475 #endif /* (!defined(__XOPEN_OR_POSIX) || (defined(_XPG3)... */
476 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
477 extern int resolvepath(const char *, char *, size_t);
478 /* per RFC 3542; This is also defined in netdb.h */
479 extern int rexec_af(char **, unsigned short, const char *, const char *,
480     const char *, int *);
481 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
482 extern int rmdir(const char *);
483 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
484 /* per RFC 3542; This is also defined in netdb.h */
485 extern int rresvport_af(int *, int);
486 #endif
487
488 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
489     defined(__EXTENSIONS__)
490 extern void *sbrk(intptr_t);
491 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
492 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG6) || defined(__EXTENSIONS__)
493 extern int setegid(gid_t);
494 extern int seteuid(uid_t);
495 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG6) ... */
496 extern int setgid(gid_t);
497 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
498 extern int setgroups(int, const gid_t *);
499 extern int sethostnamel(char *, int);
500 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
501 extern int setpgid(pid_t, pid_t);
502 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
503 extern pid_t setpgrp(void);
504 extern int setregid(gid_t, gid_t);
505 extern int setreuid(uid_t, uid_t);
506 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
507 extern pid_t setsid(void);
508 extern int setuid(uid_t);
509 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
510 extern void setusershell(void);
511 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
512 extern unsigned sleep(unsigned);
513 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
514 extern int stime(const time_t *);
515 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
516 #if defined(_XPG4)
517 /* __EXTENSIONS__ makes the SVID Third Edition prototype in stdlib.h visible */
518 extern void swab(const void *_RESTRICT_KYWD, void *_RESTRICT_KYWD, ssize_t);
519 #endif /* defined(_XPG4) */
520 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
521 extern int symlink(const char *, const char *);
522 extern void sync(void);
523 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) */

```

```

524 #if defined(_XPG5) && !defined(_XPG6)
525 #ifdef __PRAGMA_REDEFINE_EXTNAME
526 #pragma redefine_extname sysconf __sysconf_xpg5
527 #else /* __PRAGMA_REDEFINE_EXTNAME */
528 #define sysconf __sysconf_xpg5
529 #endif /* __PRAGMA_REDEFINE_EXTNAME */
530 #endif /* defined(_XPG5) && !defined(_XPG6) */
531 extern long sysconf(int);
532 extern pid_t tcgetpgrp(int);
533 extern int tcsetpgrp(int, pid_t);
534 #if !defined(__XOPEN_OR_POSIX) || \
535     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
536     defined(__EXTENSIONS__)
537 extern off_t tell(int);
538 #endif /* !defined(__XOPEN_OR_POSIX)... */
539 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
540     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
541     defined(__EXTENSIONS__)
542 extern int truncate(const char *, off_t);
543 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
544 extern char *ttyname(int);
545 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
546 extern useconds_t ualarm(useconds_t, useconds_t);
547 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
548 extern int unlink(const char *);
549 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
550 extern int usleep(useconds_t);
551 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
552 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
553 extern pid_t vfork(void) _RETURNS_TWICE;
554 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
555 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
556 extern void vhangup(void);
557 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
558 extern ssize_t write(int, const void *, size_t);
559 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
560 extern void yield(void);
561 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

563 #if !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE) || \
564     defined(__EXTENSIONS__)
565 /* || defined(_XPG7) */
566 extern int faccessat(int, const char *, int, int);
567 extern int fchownat(int, const char *, uid_t, gid_t, int);
568 extern int linkat(int, const char *, int, const char *, int);
569 extern ssize_t readlinkat(int, const char *_RESTRICT_KYWD,
570     char *_RESTRICT_KYWD, size_t);
571 extern int renameat(int, const char *, int, const char *);
572 extern int symlinkat(const char *, int, const char *);
573 extern int unlinkat(int, const char *, int);
574 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE)... */
575 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
576 extern int get_nprocs(void);
577 extern int get_nprocs_conf(void);
578 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

580 /* transitional large file interface versions */
581 #if defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
582     !defined(__PRAGMA_REDEFINE_EXTNAME))
583 extern int ftruncate64(int, off64_t);
584 extern off64_t lseek64(int, off64_t, int);
585 extern ssize_t pread64(int, void *, size_t, off64_t);
586 extern ssize_t pwrite64(int, const void *, size_t, off64_t);
587 extern off64_t tell64(int);
588 extern int truncate64(const char *, off64_t);
589 extern int lockf64(int, int, off64_t);

```

```

590 #endif /* _LARGEFILE64_SOURCE */

592 #else /* __STDC__ */

594 extern int access();
595 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
596 extern int acct();
597 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
598 extern unsigned alarm();
599 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
600     defined(__EXTENSIONS__)
601 extern int brk();
602 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
603 extern int chdir();
604 extern int chown();
605 #if !defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE) || \
606     defined(__EXTENSIONS__)
607 extern int chroot();
608 #endif /* (!defined(_POSIX_C_SOURCE) || defined(_XOPEN_SOURCE)... */
609 extern int close();
610 #if defined(_XPG4) || defined(__EXTENSIONS__)
611 extern size_t confstr();
612 extern char *crypt();
613 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
614 #if !defined(_POSIX_C_SOURCE) || defined(_XPG3) || defined(__EXTENSIONS__)
615 extern char *ctermid();
616 #endif /* (!defined(_POSIX_C_SOURCE) || defined(_XPG3)... */
617 #if !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) || defined(__EXTENSIONS__)
618 extern char *ctermid_r();
619 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_REENTRANT) ... */
620 #if !defined(_XPG6) || defined(__EXTENSIONS__)
621 extern char *cuserid();
622 #endif
623 extern int dup();
624 extern int dup2();
625 extern int dup3();
626 #if defined(_XPG4) || defined(__EXTENSIONS__)
627 extern void encrypt();
628 #endif /* defined(_XPG4) || defined(__EXTENSIONS__) */
629 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
630 extern void endusershell();
631 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
632 extern int execl();
633 extern int execle();
634 extern int execlp();
635 extern int execv();
636 extern int execve();
637 extern int execvp();
638 extern void _exit();
639 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
640 extern int fattach();
641 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
642 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
643 extern int fchdir();
644 extern int fchown();
645 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
646 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
647 extern int fchroot();
648 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
649 #if !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2) || \
650     defined(__EXTENSIONS__)
651 extern int fdasyncl();
652 #endif /* !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE > 2)... */
653 #if !defined(__XOPEN_OR_POSIX)
654 extern int fdetach();
655 #endif /* !defined(__XOPEN_OR_POSIX) */

```

```

656 extern pid_t fork();
657 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
658 extern pid_t fork1();
659 extern pid_t forkall();
660 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
661 extern long fpathconf();
662 #if !defined(__POSIX_C_SOURCE) || (__POSIX_C_SOURCE > 2) || \
663     defined(__EXTENSIONS__)
664 extern int fsync();
665 #endif /* !defined(__POSIX_C_SOURCE) || (__POSIX_C_SOURCE > 2)... */
666 #if !defined(__XOPEN_OR_POSIX) || (__POSIX_C_SOURCE > 2) || defined(__XPG4_2) || \
667     (defined(__LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
668     defined(__EXTENSIONS__)
669 extern int ftruncate();
670 #endif /* !defined(__XOPEN_OR_POSIX) (__POSIX_C_SOURCE > 2)... */
671 extern char *getcwd();
672 #if !defined(__XOPEN_OR_POSIX) || (defined(__XPG4_2) && !defined(__XPG6)) || \
673     defined(__EXTENSIONS__)
674 extern int getdtablesize();
675 #endif
676 extern gid_t getegid();
677 extern uid_t geteuid();
678 extern gid_t getgid();
679 extern int getgroups();
680 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
681 extern long gethostid();
682 #endif
683 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
684 extern int gethostname();
685 #endif
686 extern char *getlogin();
687 #if defined(__XPG4) || defined(__EXTENSIONS__)
688 extern int getopt();
689 extern char *optarg;
690 extern int opterr, optind, optopt;
691 #if !defined(__XPG6) || defined(__EXTENSIONS__)
692 extern char *getpass();
693 #endif
694 #endif /* defined(__XPG4) || defined(__EXTENSIONS__) */
695 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
696 #if !defined(__XPG6) || defined(__EXTENSIONS__)
697 extern int getpagesize();
698 #endif
699 extern pid_t getpgid();
700 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2)... */
701 extern pid_t getpid();
702 extern pid_t getppid();
703 extern pid_t getpgrp();
704 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
705 char *gettxt();
706 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
707 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
708 extern pid_t getsid();
709 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) */
710 extern uid_t getuid();
711 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
712 extern char *getusershell();
713 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
714 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
715 extern char *getwd();
716 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2)... */
717 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
718 extern int ioctl();
719 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
720 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
721 extern int isaexec();

```

```

722 extern int issetugid();
723 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
724 extern int isatty();
725 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
726 extern int lchown();
727 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) */
728 extern int link();
729 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
730 extern off_t llseek();
731 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
732 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || \
733     (defined(__LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
734     defined(__EXTENSIONS__)
735 extern int lockf();
736 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2)... */
737 extern off_t lseek();
738 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
739 extern int mincore();
740 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
741 #if !defined(__POSIX_C_SOURCE) || defined(__XOPEN_SOURCE) || \
742     defined(__EXTENSIONS__)
743 extern int nice();
744 #endif /* !defined(__POSIX_C_SOURCE) || defined(__XOPEN_SOURCE)... */
745 extern long pathconf();
746 extern int pause();
747 extern int pipe();
748 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG5) || \
749     (defined(__LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
750     defined(__EXTENSIONS__)
751 extern ssize_t pread();
752 #endif
753 #if !defined(__LP64) && \
754     (!defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__))
755 extern void profil();
756 extern int ptrace();
757 #endif
758 #if !defined(__XOPEN_OR_POSIX) || \
759     ((__POSIX_C_SOURCE > 2) && !defined(__XPG6)) || \
760     defined(__EXTENSIONS__)
761 extern int pthread_atfork();
762 #endif /* !defined(__XOPEN_OR_POSIX) || ((__POSIX_C_SOURCE > 2) ... */
763 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG5) || \
764     (defined(__LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
765     defined(__EXTENSIONS__)
766 extern ssize_t pwrite();
767 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG5) */
768 extern ssize_t read();
769 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
770 /* per RFC 3542; This is also defined in netdb.h */
771 extern int rcmd_af();
772 #endif
773 #if !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2) || defined(__EXTENSIONS__)
774 extern ssize_t readlink();
775 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__XPG4_2)... */
776 #if (!defined(__XOPEN_OR_POSIX) || (defined(__XPG3) && !defined(__XPG4))) || \
777     defined(__EXTENSIONS__)
778 extern int rename();
779 #endif /* (!defined(__XOPEN_OR_POSIX) || (defined(__XPG3)... */
780 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
781 extern int resolvepath();
782 /* per RFC 3542; This is also defined in netdb.h */
783 extern int rexec_af();
784 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
785 extern int rmdir();
786 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
787 /* per RFC 3542; This is also defined in netdb.h */

```

```

788 extern int rresvport_af();
789 #endif
790 #if !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2) && !defined(_XPG6)) || \
791     defined(__EXTENSIONS__)
792 extern void *sbrk();
793 #endif /* !defined(__XOPEN_OR_POSIX) || (defined(_XPG4_2)... */
794 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG6) || defined(__EXTENSIONS__)
795 extern int setegid();
796 extern int seteuid();
797 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG6) ... */
798 extern int setgid();
799 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
800 extern int setgroups();
801 extern int sethostname();
802 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
803 extern int setpgid();
804 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
805 extern pid_t setpgpr();
806 extern int setregid();
807 extern int setreuid();
808 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
809 extern pid_t setsid();
810 extern int setuid();
811 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
812 extern void setusershell();
813 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
814 extern unsigned sleep();
815 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
816 extern int stime();
817 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
818 #if defined(_XPG4)
819 /* __EXTENSIONS__ makes the SVID Third Edition prototype in stdlib.h visible */
820 extern void swab();
821 #endif /* defined(_XPG4) */
822 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
823 extern int symlink();
824 extern void sync();
825 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
826 #if defined(_XPG5)
827 #ifdef __PRAGMA_REDEFINE_EXTNAME
828 #pragma redefine_extname sysconf __sysconf_xpg5
829 extern long sysconf();
830 #else /* __PRAGMA_REDEFINE_EXTNAME */
831 extern long __sysconf_xpg5();
832 #define sysconf __sysconf_xpg5
833 #endif /* __PRAGMA_REDEFINE_EXTNAME */
834 #endif /* defined(_XPG5) */
835 extern pid_t tcgetpgrp();
836 extern int tcsetpgrp();
837 #if !defined(__XOPEN_OR_POSIX) || \
838     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
839     defined(__EXTENSIONS__)
840 extern off_t tell();
841 #endif /* !defined(__XOPEN_OR_POSIX)... */
842 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || \
843     (defined(_LARGEFILE_SOURCE) && _FILE_OFFSET_BITS == 64) || \
844     defined(__EXTENSIONS__)
845 extern int truncate();
846 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
847 extern char *ttyname();
848 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
849 extern useconds_t ualarm();
850 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
851 extern int unlink();
852 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
853 extern int usleep();

```

```

854 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
855 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
856 extern pid_t vfork();
857 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */
858 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
859 extern void vhangup();
860 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */
861 extern ssize_t write();
862 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
863 extern void yield();
864 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

866 #if !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE) || \
867     defined(__EXTENSIONS__)
868     /* || defined(_XPG7) */
869 extern int faccessat();
870 extern int fchownat();
871 extern int linkat();
872 extern ssize_t readlinkat();
873 extern int renameat();
874 extern int symlinkat();
875 extern int unlinkat();
876 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_ATFILE_SOURCE)... */
877 #if !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__)
878 extern int get_nprocs();
879 extern int get_nprocs_conf();
880 #endif /* !defined(__XOPEN_OR_POSIX) || defined(__EXTENSIONS__) */

882 /* transitional large file interface versions */
883 #if defined(_LARGEFILE64_SOURCE) && !((_FILE_OFFSET_BITS == 64) && \
884     !defined(__PRAGMA_REDEFINE_EXTNAME))
885 extern int ftruncate64();
886 extern off64_t lseek64();
887 extern ssize_t pread64();
888 extern ssize_t pwrite64();
889 extern off64_t tell64();
890 extern int truncate64();
891 extern int lockf64();
892 #endif /* _LARGEFILE64_SOURCE */

894 #endif /* _STDCL_ */

896 #if !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2) || defined(__EXTENSIONS__)
897 #pragma unknown_control_flow(vfork)
898 #endif /* !defined(__XOPEN_OR_POSIX) || defined(_XPG4_2)... */

900 /*
901 * getlogin_r() & ttyname_r() prototypes are defined here.
902 */

904 /*
905 * Previous releases of Solaris, starting at 2.3, provided definitions of
906 * various functions as specified in POSIX.1c, Draft 6. For some of these
907 * functions, the final POSIX 1003.1c standard had a different number of
908 * arguments and return values.
909 *
910 * The following segment of this header provides support for the standard
911 * interfaces while supporting applications written under earlier
912 * releases. The application defines appropriate values of the feature
913 * test macros _POSIX_C_SOURCE and _POSIX_PTHREAD_SEMANTICS to indicate
914 * whether it was written to expect the Draft 6 or standard versions of
915 * these interfaces, before including this header. This header then
916 * provides a mapping from the source version of the interface to an
917 * appropriate binary interface. Such mappings permit an application
918 * to be built from libraries and objects which have mixed expectations
919 * of the definitions of these functions.

```

```

920 *
921 * For applications using the Draft 6 definitions, the binary symbol is the
922 * same as the source symbol, and no explicit mapping is needed. For the
923 * standard interface, the function func() is mapped to the binary symbol
924 * __posix_func(). The preferred mechanism for the remapping is a compiler
925 * #pragma. If the compiler does not provide such a #pragma, the header file
926 * defines a static function func() which calls the __posix_func() version;
927 * this has to be done instead of #define since POSIX specifies that an
928 * application can #undef the symbol and still be bound to the correct
929 * implementation. Unfortunately, the statics confuse lint so we fallback to
930 * #define in that case.
931 *
932 * NOTE: Support for the Draft 6 definitions is provided for compatibility
933 * only. New applications/libraries should use the standard definitions.
934 */

936 #if defined(__EXTENSIONS__) || defined(REENTRANT) || \
937 !defined(__XOPEN_OR_POSIX) || (_POSIX_C_SOURCE - 0 >= 199506L) || \
938 defined(_POSIX_PTHREAD_SEMANTICS)

940 #if defined(__STDC__)

942 #if (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_POSIX_PTHREAD_SEMANTICS)

944 #ifdef __PRAGMA_REDEFINE_EXTNAME
945 #pragma redefine_extname getlogin_r __posix_getlogin_r
946 #pragma redefine_extname ttyname_r __posix_ttyname_r
947 extern int getlogin_r(char *, int);
948 extern int ttyname_r(int, char *, size_t);
949 #else /* __PRAGMA_REDEFINE_EXTNAME */

951 extern int __posix_getlogin_r(char *, int);
952 extern int __posix_ttyname_r(int, char *, size_t);

954 #ifdef __lint

956 #define getlogin_r __posix_getlogin_r
957 #define ttyname_r __posix_ttyname_r

959 #else /* !__lint */

961 static int
962 getlogin_r(char *__name, int __len)
963 {
964     return (__posix_getlogin_r(__name, __len));
965 }
966 static int
967 ttyname_r(int __fildes, char *__buf, size_t __size)
968 {
969     return (__posix_ttyname_r(__fildes, __buf, __size));
970 }

972 #endif /* !__lint */
973 #endif /* __PRAGMA_REDEFINE_EXTNAME */

975 #else /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

977 extern char *getlogin_r(char *, int);
978 extern char *ttyname_r(int, char *, int);

980 #endif /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

982 #else /* __STDC__ */

984 #if (_POSIX_C_SOURCE - 0 >= 199506L) || defined(_POSIX_PTHREAD_SEMANTICS)

```

```

986 #ifdef __PRAGMA_REDEFINE_EXTNAME
987 #pragma redefine_extname getlogin_r __posix_getlogin_r
988 #pragma redefine_extname ttyname_r __posix_ttyname_r
989 extern int getlogin_r();
990 extern int ttyname_r();
991 #else /* __PRAGMA_REDEFINE_EXTNAME */

993 extern int __posix_getlogin_r();
994 extern int __posix_ttyname_r();

996 #ifdef __lint

998 #define getlogin_r __posix_getlogin_r
999 #define ttyname_r __posix_ttyname_r

1001 #else /* !__lint */

1003 static int
1004 getlogin_r(__name, __len)
1005     char *__name;
1006     int __len;
1007 {
1008     return (__posix_getlogin_r(__name, __len));
1009 }
1010 static int
1011 ttyname_r(__fildes, __buf, __size)
1012     int __fildes;
1013     char *__buf;
1014     size_t __size;
1015 {
1016     return (__posix_ttyname_r(__fildes, __buf, __size));
1017 }
1018 #endif /* !__lint */
1019 #endif /* __PRAGMA_REDEFINE_EXTNAME */

1021 #else /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

1023 extern char *getlogin_r();
1024 extern char *ttyname_r();

1026 #endif /* (_POSIX_C_SOURCE - 0 >= 199506L) || ... */

1028 #endif /* __STDC__ */

1030 #endif /* defined(__EXTENSIONS__) || defined(REENTRANT)... */

1032 #ifdef __cplusplus
1033 }
1034 #endif

1036 #endif /* _UNISTD_H */

```