Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'stdlib.h.0p' command

$ man stdlib.h.0p

stdlib.h(0P) POSIX Programmer's Manual stdlib.h(0P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

stdlib.h ? standard library definitions

SYNOPSIS

#include <stdlib.h>

DESCRIPTION

Some of the functionality described on this reference page extends the ISO C standard. Applications shall define the appropriate feature test macro (see the System Interfaces volume of POSIX.1?2017, Section 2.2, The Compilation Environment) to enable the visibility of these symbols in this header.

The <stdlib.h> header shall define the following macros which shall ex? pand to integer constant expressions:

EXIT_FAILURE

Unsuccessful termination for exit(); evaluates to a non-zero value.

EXIT_SUCCESS

Successful termination for exit(); evaluates to 0.

{RAND_MAX} Maximum value returned by rand(); at least 32767.

The <stdlib.h> header shall define the following macro which shall ex?

pand to a positive integer expression with type size_t:

{MB_CUR_MAX}

Maximum number of bytes in a character specified by the current locale (category LC_CTYPE).

In the POSIX locale the value of {MB_CUR_MAX} shall be 1.

The <stdlib.h> header shall define NULL as described in <stddef.h>.

The <stdlib.h> header shall define the following data types through

typedef:

div_t Structure type returned by the div() function.

ldiv_t Structure type returned by the ldiv() function.

lldiv_t Structure type returned by the lldiv() function.

size_t As described in <stddef.h>.

wchar_t As described in <stddef.h>.

In addition, the <stdlib.h> header shall define the following symbolic

constants and macros as described in <sys/wait.h>:

WEXITSTATUS

WIFEXITED

WIFSIGNALED

WIFSTOPPED

WNOHANG

WSTOPSIG

WTERMSIG

WUNTRACED

The following shall be declared as functions and may also be defined as

macros. Function prototypes shall be provided.

```
void        _Exit(int);
long        a64l(const char *);
void        abort(void);
int         abs(int);
int         atexit(void (*)(void));
double      atof(const char *);
```

```
int           atoi(const char *);
long          atol(const char *);
long long     atoll(const char *);
void         *bsearch(const void *, const void *, size_t, size_t,
                int (*)(const void *, const void *));
void         *calloc(size_t, size_t);
div_t         div(int, int);
double        drand48(void);
double        erand48(unsigned short [3]);
void          exit(int);
void          free(void *);
char         *getenv(const char *);
int           getsubopt(char **, char *const *, char **);
int           grantpt(int);
char         *initstate(unsigned, char *, size_t);
long          jrand48(unsigned short [3]);
char         *l64a(long);
long          labs(long);
void          lcong48(unsigned short [7]);
ldiv_t        ldiv(long, long);
long long     llabs(long long);
lldiv_t       lldiv(long long, long long);
long          lrand48(void);
void         *malloc(size_t);
int           mblen(const char *, size_t);
size_t        mbstowcs(wchar_t *restrict, const char *restrict, size_t);
int           mbtowc(wchar_t *restrict, const char *restrict, size_t);
char         *mkdtemp(char *);
int           mkstemp(char *);
long          mrand48(void);
long          nrand48(unsigned short [3]);
int           posix_memalign(void **, size_t, size_t);
int           posix_openpt(int);
```

```
char         *ptsname(int);
int          putenv(char *);
void         qsort(void *, size_t, size_t, int (*)(const void *,
                 const void *));
int          rand(void);
int          rand_r(unsigned *);
long         random(void);
void         *realloc(void *, size_t);
char         *realpath(const char *restrict, char *restrict);
unsigned short *seed48(unsigned short [3]);
int          setenv(const char *, const char *, int);
void         setkey(const char *);
char         *setstate(char *);
void         srand(unsigned);
void         srand48(long);
void         srandom(unsigned);
double       strtod(const char *restrict, char **restrict);
float        strtof(const char *restrict, char **restrict);
long         strtol(const char *restrict, char **restrict, int);
long double  strtold(const char *restrict, char **restrict);
long long    strtoll(const char *restrict, char **restrict, int);
unsigned long strtoul(const char *restrict, char **restrict, int);
unsigned long long
             strtoull(const char *restrict, char **restrict, int);
int          system(const char *);
int          unlockpt(int);
int          unsetenv(const char *);
size_t       wcstombs(char *restrict, const wchar_t *restrict, size_t);
int          wctomb(char *, wchar_t);
```

Inclusion of the <stdlib.h> header may also make visible all symbols from <stddef.h>, <limits.h>, <math.h>, and <sys/wait.h>.

The following sections are informative.

APPLICATION USAGE

None.

RATIONALE

None.

FUTURE DIRECTIONS

None.

SEE ALSO

<limits.h>, <math.h>, <stddef.h>, <sys_types.h>, <sys_wait.h>

The System Interfaces volume of POSIX.1?2017, Section 2.2, The Compila?

tion Environment, _Exit(), a64l(), abort(), abs(), atexit(), atof(),

atoi(), atol(), bsearch(), calloc(), div(), drand48(), exit(), free(),

getenv(), getsubopt(), grantpt(), initstate(), labs(), ldiv(), mal?

loc(), mblen(), mbstowcs(), mbtowc(), mkdtemp(), posix_memalign(),

posix_openpt(), ptsname(), putenv(), qsort(), rand(), realloc(), real?

path(), setenv(), setkey(), strtod(), strtol(), strtoul(), system(),

unlockpt(), unsetenv(), wcstombs(), wctomb()

COPYRIGHT