

使用できる関数の一覧

プロトタイプ・リンク・Tipsなど。

getline

char *getline (prompt)

<https://nxmlnpg.lemoda.net/ja/3/readline>

- 端末から一行を読み込み、その値を返す。
- 入力処理で使用。

rl_clear_history

- readlineの履歴をすべて削除する
- freeもしてくれる？

rl_on_new_line

int rl_on_new_line (void)

https://tiswww.case.edu/php/chet/readline/readline.html#index-rl_005fon_005fnew_005fline

- 新しく改行したことを関数に伝える。

rl_replace_line

void rl_replace_line (const char *text, int clear_undo);

- rl_line_bufferの文字列をtextで置き換える
- clear_undoが0でない場合、現在行に紐づくundoリストが削除される。

rl_redisplay

void rl_redisplay (void)

<https://tiswww.case.edu/php/chet/readline/readline.html#Redisplay>

- rl_line_bufferの内容を再出力する。

add_history

- 文字列を履歴として追加する

printf

int printf(const char *format, ...)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/printf.3.html

malloc

void malloc(size_t size)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/malloc.3.html

free

void free(void *ptr)

<https://linuxjm.osdn.jp/html/procps/man1/free.1.html>

write

ssize_t write(int fd, const void *buf, size_t count)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/write.2.html

access

open

int open(const char *pathname, int flags)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/open.2.html

read

ssize_t read(int fd, void *buf, size_t count)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/read.2.html

close

int close(int fd)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/close.2.html

fork

pid_t fork(void)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/fork.2.html

- 子プロセスを作成する

wait

pid_t wait(int *wstatus)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/wait.2.html

- 子プロセスの終了を待機する。

waitpid

pid_t waitpid(pid_t pid, int *wstatus, int options)

- 特定の子プロセスの終了を待機する。

wait3

wait3(int *status, int options, struct rusage *rusage)

<https://nxmlnpg.lemoda.net/ja/2/wait3>

- 子プロセスの状態変更を待機する。

wait4

wait4(pid_t wpid, int *status, int options, struct rusage *rusage)

- 特定の子プロセスの状態変更を待機する。

signal

https://linuxjm.osdn.jp/html/LDP_man-pages/man7/signal.7.html

sigaction

- 割り込み処理

sigemptyset

- シグナルセットを空にする。

sigaddset

- シグナルマスクにシグナルを追加する。

kill

int kill(pid_t pid, int sig)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/kill.2.html

exit

void exit(int status)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/exit.3.html

getcwd

char *getcwd(char *buf, size_t size)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/getcwd.2.html

- カレントディレクトリを取得する。

chdir

int chdir(const char *path)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/chdir.2.html

- カレントディレクトリを変更する。

stat

int stat(const char *pathname, struct stat *buf)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/stat.2.html

- ファイルの状態を取得する。

lstat

int lstat(const char *pathname, struct stat *buf)

- ファイルの状態を取得する。

fstat

```
int fstat(int fd, struct stat *buf);
```

- ファイルの状態を取得する（ファイルはfdで指定する）。

unlink

- ファイルシステム上の名前を削除する。

execve

```
int execve(const char *pathname, char *const argv[],char *const envp[])
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/execve.2.html

dup

```
int dup(int oldfd)
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/dup.2.html,

- ファイルディスクリプターを複製する。
- forkと組み合わせてプロセス間通信に使う。

dup2

```
int dup2(int oldfd, int newfd)
```

- ファイルディスクリプターを複製する。
- forkと組み合わせてプロセス間通信に使う。

pipe

```
int pipe(int pipefd2)
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/pipe.2.html

- パイプを作成する。

opendir

```
DIR *opendir(const char *name)
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/opendir.3.html

- ディレクトリをオープンする。

readdir

```
struct dirent *readdir(DIR *dirp)
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/readdir.3.html

- オープンしたディレクトリからファイル一覧を取得する。

closedir

```
int closedir(DIR *dirp)
```

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/closedir.3.html

- ディレクトリをクローズする。

strerror

char *strerror(int errnum)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/strerror_r.3.html

perror

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/errno.3.html

- errnoに対応するエラーメッセージを出力する。

getenv

char *getenv(const char *name)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/getenv.3.html

- 環境変数のvalueを取得する。

(以下は旧要件の名残であり、基本的に使用しない)

isatty

int isatty(int fd)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/isatty.3.html

- ファイルディスクリプターが端末を参照しているか判定する。

ttyname

char *ttyname(int fd)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/ttyname.3.html

- ファイルディスクリプターが参照している端末のパスを取得する。

ttyslot

int ttyslot(void)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/ttyslot.3.html

- カレントユーザーの端末のスロットをファイルから探す

ioctl

int ioctl(int fd, unsigned long request, ...)

https://linuxjm.osdn.jp/html/LDP_man-pages/man2/ioctl.2.html

- スペシャルファイルを構成するデバイスのパラメーターを 操作する

tcsetattr

int tcsetattr(int fd, int optional_actions, const struct termios *termios_p)

https://linuxjm.osdn.jp/html/LDP_man-pages/man3/termios.3.html

- 端末の属性を設定する。

tcgetattr

int tcgetattr(int fd, struct termios *termios_p)

- 端末の属性を取得する。

tgetent

int tgetent(char *bp, const char *name)

<https://nxmlnpg.lemoda.net/ja/3/tgetent>

- 端末名の入力をbpに書き出す。

tgetflag

int tgetflag(const char *id)

- 指定した機能が端末に存在するかを判定する。

tgetnum

int tgetnum(const char *id)

- 指定した機能idが与えられていないか判定する。

tgetstr

char *tgetstr(const char *id, char **area)

tgoto

char *tgoto(const char *cm, int destcol, int destline)

tputs

int tputs(const char *cp, int affcnt, int (*outc)(int))