

Fonctions autorisées - Minishell

getline(prompt)

Lit une ligne avec un prompt (gère l'historique, flèches, etc.).

rl_clear_history()

Efface l'historique de readline.

rl_on_new_line()

Indique qu'on est sur une nouvelle ligne.

rl_replace_line(line, clear_undo)

Remplace la ligne actuelle dans le buffer.

rl_redisplay()

Réaffiche la ligne actuelle.

add_history(line)

Ajoute une ligne à l'historique de readline.

printf

Affiche du texte formaté.

malloc(size)

Alloue dynamiquement de la mémoire.

free(ptr)

Libère de la mémoire allouée avec malloc.

write(fd, buf, size)

Écrit des données dans un fichier ou terminal.

access(path, mode)

Vérifie les droits d'accès à un fichier.

open(path, flags)

Ouvre un fichier.

read(fd, buf, size)

Lit des données depuis un fichier.

close(fd)

Ferme un descripteur de fichier.

fork()

Crée un processus enfant.

wait(pid)

Attend la fin d'un processus enfant.

waitpid(pid, &status, options)

Attend un processus spécifique.

wait3

Version avancée de wait avec infos CPU.

wait4

Comme wait3 avec plus de détails.

signal(sig, handler)

Définit une fonction pour un signal.

sigaction(sig, &act, &oldact)

Version plus sûre de signal.

sigemptyset(set)

Crée un set de signaux vide.

sigaddset(set, sig)

Ajoute un signal dans un set.

kill(pid, sig)

Envoie un signal à un processus.

exit(status)

Quitte le programme avec un code.

getcwd(buf, size)

Retourne le chemin actuel.

chdir(path)

Change le répertoire courant.

stat(path, &buf)

Infos sur un fichier.

lstat(path, &buf)

Comme stat mais ne suit pas les liens symboliques.

fstat(fd, &buf)

Infos fichier via descripteur.

unlink(path)

Supprime un fichier.

execve(path, argv, envp)

Remplace le processus courant.

dup(fd)

Duplique un descripteur.

dup2(fd, fd2)

Redirige un descripteur vers un autre.

pipe(fd[2])

Crée un pipe entre deux processus.

opendir(path)

Ouvre un répertoire.

readdir(dir)

Lit une entrée de répertoire.

closedir(dir)

Ferme un répertoire.

strerror(errno)

Retourne un message d'erreur.

perror(msg)

Affiche un message d'erreur.

isatty(fd)

Vérifie si un descripteur est un terminal.

ttyname(fd)

Nom du terminal pour un descripteur.

ttyslot()

Slot de terminal du processus.

ioctl(fd, request, ...)

Contrôle des périphériques.

getenv(name)

Lit une variable d'environnement.

tcsetattr(fd, &termios)

Change les attributs du terminal.

tcgetattr(fd, &termios)

Récupère les attributs du terminal.

tgetent(buf, termtype)

Initialise termcap.

tgetflag(id)

Vérifie une capacité termcap.

tgetnum(id)

Lit une valeur numérique termcap.

tgetstr(id, &area)

Lit une chaîne termcap.

tgoto(cap, col, row)

Positionne le curseur.

tputs(str, affcnt, putc)

Affiche une chaîne termcap.