

## Fonctions minishell

### readline : <stdio.h>

- ↳ lis une ligne du terminal et la renvoie, utilise prompt comme invite.
- ↳ ligne retournée alloué avec malloc
- ↳ capacité d'édition pendant qu'utilisateur entre dans la ligne

#### Valeur de retour

- ↳ renvoie texte de la ligne lue
- ↳ Si EOF rencontré, NULL renvoyé.

Readline est personnalisé en plaçant des commandes dans un fichier init.

- ↳ nom du fichier → **INPUTRC**
- ↳ programme démarre, fichier init lu, raccourcis claviers & variables définis.
  - ↳ ligne vide ignorée
  - ↳ # commentaire
  - ↳ \$ constructions conditionnelles

### Fichier inputrc

- ↳ nom d'une commande avec séquence de touches, elle doit être liée.
- ↳ nom & séquence de touche séparé par :

BASH permet afficher ou modifier raccourcis claviers actuels avec commande **BIND**

→ variables pouvant être utilisées pour personnaliser comportement

### Constructions conditionnelles

- ↳ **\$if** → effectue des liaisons en fonction du mode d'édition
  - ↳ mode = → teste si readline mode emacs ou vi
  - ↳ term = → inclure liaisons de touche spécifiques au terminal, teste par rapport au nom complet dans le terminal.
  - ↳ applicat° → inclure paramètres spécifiques à l'appli°
- ↳ **\$endif** → termine commande if
- ↳ **\$autre** → commandes de \$if exécutées si test échoue
- ↳ **\$include** → un seul nom de fichier comme argument & lit cmd, liaisons de

fichiers



## Recherche dans l'historique des commandes

↳ 2 modes de recherche : incrémentiel & non incrémentiel

### ↳ recherches incrémentielles :

- ↳ commencent avant que l'utilisateur fini de taper chaîne de recherche.
- ↳ au fur et à mesure, nouvelle entrée suivante de l'historique correspondant à la chaîne saisie
- ↳ nécessite nombre de caractères nécessaires pour trouver entrée historique souhaitée

### ↳ recherches non incrémentielles

- ↳ lit l'intégralité de la chaîne de recherche avant de commencer recherche lignes d'historique correspondantes.

↳ rl-clear-history : efface liste historique, supprime toutes entrées

↳ rl-replace-line : remplace contenu de buf

↳ rl-redisplay : modifie affichage, écran reflète contenu actuel buf

↳ add-history : enregistre ligne dans l'historique de ces lignes

signal :  $\langle \text{signal.h} \rangle$

↳ interface simplifiée

↳ signaux permet manipulation processus depuis extérieur domaine, manipuler ou copier

↳ 2 types de signaux

↳ provoque arrêt, résultat erreur irrécupérable

↳ provoque × arrêt

sigaction :  $\langle \text{signal.h} \rangle$

↳ modifier act° entreprise par un processus à receipt° d'un signal spécifique

sigemptyset :  $\langle \text{signal.h} \rangle$

↳ initialise ensemble signaux pointé vers paramètre fonct°, tous signaux de POSIX exclus.

sigaddset :  $\langle \text{signal.h} \rangle$

↳ ajoute signal individuel spécifié dans 2<sup>ème</sup> paramètre à l'ensemble des signaux pointé par 1<sup>er</sup> paramètre



kill :  $\langle \text{signal}, h \rangle$

- ↳ envoyer signal à processus
- ↳ si PID positif, signal envoyé avec id PID
- ↳ si PID = 0, sig envoyé à chaque processus du groupe de processus
- ↳ si " = -1, sig envoyé tous processus permis envoyer signaux
- ↳ si PID < -1, sig envoyé à chaque processus du groupe dont ID est -PID
- ↳ si PID vaut 0, x signal

getcwd :  $\langle \text{unistd}, h \rangle$

- ↳ copie chemin accès absolu du fichier courant de travail vers tableau pointé par paramètre buf, détaille de longueur, paramètre taille
- ↳ si longueur chemin accès dépasse taille octets, NULL est renvoyé.

chdir :  $\langle \text{unistd}, h \rangle$

- ↳ change répertoire de travail de l'appelant processus, dans répertoire spécifié dans PATH (paramètre fond<sup>o</sup>)

stat :  $\langle \text{sys} / \text{stat}, h \rangle$

- ↳ récupérer infos sur fichier pointé par "chemin d'accès" (paramètre fond<sup>o</sup>)
- ↳ lstat
  - ↳ identique à stat, sauf si path = lien symbolique, renvoie infos sur lien.

↳ fstat

- ↳ " , sauf si fichier est fichier fd

unlink :  $\langle \text{unistd}, h \rangle$

- ↳ supprime un nom système de fichiers

isatty :  $\langle \text{unistd}, h \rangle$

- ↳ teste si fd est un descripteur de fichier ouvert & referant à terminal

ttyname :  $\langle \text{unistd}, h \rangle$

- ↳ renvoie un pointeur vers chemin accès terminal ouvert sur fd.

lseek :  $\langle \text{unistd}, h \rangle$

- ↳ renvoie indice actuel entrée utilisateur fichier

ioctl :  $\langle \text{sys} / \text{ioctl}, h \rangle$

- ↳ manipule périphérique sous-jacent paramètres fichiers spécifiques.



getenv: < stdlib.h >

↳ renvoie pointeur vers valeur dans env

termios: < termios.h >

↳ fixe paramètres terminal en lisant structure termios pointée par termios-p précise quand changement lieu

tcgetattr: < termios.h >

↳ récupère paramètres associés à objet référence par fd et stocke dans structure termios

↳ peut être appelé par processus asservi plan

tcsetattr: < curses.h > < term.h >

↳ aide à la conversion

↳ interage capacités entrées pour terminfo compilée

↳ tcgetent

↳ charge entrée pour name

↳ 1 si succès, 0 si pas cet-entrée, -1 si base donnée introuvable

↳ tcgetflag

↳ obtient entrée binaire pour id

↳ tcgetnum

↳ obtient entrée numérique pour id

↳ tcgetstr

↳ renvoie entrée de chaîne pour id (tput affiche chaîne renvoyée)

↳ tgetch

↳ instancie paramètres dans capacité donnée (Sortie transmise à tput)

↳ tput

↳ récupère fonctionnalités par nom