

Annexe A

Opérateurs et variables shell

test et []

Opérateur	Description
! expression	True si l'expression est fausse. Elle peut être utilisée pour inverser n'importe quel autre test
-b fichier	True si le fichier est un périphérique bloc
-c fichier	True si le fichier est un périphérique caractère
-d fichier	True si le fichier est un répertoire
-e fichier	True si le fichier existe
-f fichier	True si le fichier est un fichier normal
-p fichier	True si le fichier est un FIFO
-L fichier	True si le fichier est un lien symbolique
-S fichier	True si le fichier est un socket Unix
-G fichier	True si le fichier appartient au même groupe que le GID effectif du processus courant
-n chaîne	True si la longueur de la chaîne $\neq 0$
-z chaîne	True si la longueur de la chaîne = 0
-O fichier	True si le fichier appartient à la même personne que l'UID effectif du processus courant
-g fichier	True si le fichier a le bit setgid armé
-r fichier	True si les permissions de l'utilisateur sont suffisantes pour lire le fichier
-k fichier	True si le fichier a le bit sticky armé
-t [fd]	True si le descripteur de fichier correspond à un terminal réel. fd=1 par défaut (sortie standard)

-u fichier	True si le fichier a le bit setuid armé
-w fichier	True si les permissions de l'utilisateur courant sont suffisantes pour écrire dans le fichier spécifié
-x fichier	True si les permissions de l'utilisateur courant sont suffisantes pour exécuter le fichier spécifié
Expression1 -a expression2	True si les deux expressions spécifiées sont vraies
Expression1 -o expression2	True si au moins une des expressions spécifiées est vraie
fichier1 -ef fichier2	True si les deux fichiers correspondent au même numéro d'inode sur le même périphérique
fichier1 -nt fichier2	True si la date de dernière modification du fichier1 est plus récente que celle du fichier2
fichier1 -ot fichier2	True si la date de dernière modification du premier fichier est plus ancienne que celle du second
nombre1 -eq nombre2	True si nombre1 = nombre2
nombre1 -ne nombre2	True si nombre1 \neq nombre2
nombre1 -le nombre2	True si nombre1 \leq nombre2
nombre1 -lt nombre2	True si nombre1 < nombre2
nombre1 -ge nombre2	True si nombre1 \geq nombre2
nombre1 -gt nombre2	True si nombre > nombre2
chaine1 = chaine2	True si les deux chaînes sont égales
chaine1 != chaine2	True si les chaînes ne sont pas égales

Variables bash spéciales

W = écriture ; R = lecture

Variable	Accès	Description
!	R	ID du processus en arrière-plan le plus récent
@	R	Contient tous les paramètres du contexte courant Lorsqu'elle est utilisée entre des délimiteurs doubles, elle est évaluée comme des valeurs délimitées séparées, une pour chaque paramètre passé au contexte courant
#	R	Nombre de paramètres du contexte courant
*	R	Tous les paramètres du contexte courant Si elle est utilisée entre des délimiteurs doubles, le résultat est un unique paramètre contenant tous les paramètres passés, séparés par des espaces
\$	R	ID de processus bash actuel
-	R	Liste des flags d'option courants (de la commande), une lettre pour chacun, sans séparation
_	R	Nom de chemin complet du processus courant au cours de l'initialisation. En recherche de mail, contient le nom du fichier de mail courant Dans les autres cas, contient l'argument final de la commande précédente
0	R	Nom du processus ou script en cours
1 à 9	R	9 premiers paramètres du script (ou de la fonction) courant
BASH	R	Chemin complet du shell courant
BASH_VERSION	R	Numéro de version de votre bash
BASH_VERSINFO	R	Informations sur la version actuel de bash
DISPLAY	W	Nom et le numéro de la console de la machine sur laquelle les clients X affichent leurs interfaces
EUID	R	ID utilisateur effectif numérique du processus shell courant
HISTCMD	R	Index numérique de la commande actuelle dans l'historique des commandes

HISTSIZE	W	Taille maximale de l'historique des commandes
HOME	R	Chemin du répertoire de l'utilisateur
HOSTNAME	R	Nom de la machine
HOSTTYPE	R	Architecture de la machine
IFS	W	Valeur du séparateur de champs interne. Cette valeur est utilisée pour découper les commandes en leurs composants
LANG	W	Indique la localisation courante (ou préférée) aux programmes qui supportent l'internationalisation Linux. En tant que telle, elle est souvent utilisée comme variable d'environnement
LD_LIBRARY_PATH	W	Spécifie les emplacements additionnels (séparés par :) dans lesquels doivent être cherchées les bibliothèques partagées pour les exécutables liés dynamiquement
LD_PRELOAD	W	Spécifie une liste de bibliothèques spécifiques (séparées par des espaces) à charger dans les autres, y compris celles spécifiées par le programme lui-même. Pour des raisons de sécurité, cette spécification peut être incompatible avec setuid et setgid
LINENO	R	Lorsqu'elle est utilisée dans un shell ou une fonction, contient l'offset dans les lignes depuis le démarrage de ce shell ou cette fonction
MACHTYPE	R	Identificateur GNU du type de machine
MAIL	W	Vous informe de l'arrivée d'un nouveau mail dans une boîte de style UNIX mbox. Si vous voulez que bash vous informe automatiquement, spécifiez l'emplacement de cette boîte
MAILCHECK	W	Intervalle (en secondes) qui indique la fréquence de contrôle d'arrivée de nouveau mail dans la boîte spécifiée
OLDPWD	R	Nom du répertoire de travail précédent
OSTYPE	R	Nom du système d'exploitation courant
PATH	W	Liste de répertoires séparés par : dans laquelle on cherche les binaires lors de l'exécution de programmes. C'est une variable d'environnement

PPID	R	ID du père du processus en cours
PS1	W	Forme de générer l'invite de bash
PWD	R	Répertoire de travail courant
RANDOM	R	Valeur aléatoire différente. Dépendant de l'implantation
REPLY	R	Valeur des données lues sur l'entrée standard lorsqu'on y accède après la commande read, à moins qu'une variable différente ne soit spécifiée à read
SECONDS	R	Nombre de secondes écoulées depuis le démarrage du processus shell courant
SHELLOPTS	R	Liste des options du shell courant
UID	R	ID utilisateur numérique réel du propriétaire du processus shell courant

Annexe B

Appels système

<p><code>fork()</code> : Création de nouveaux processus</p>
<p>Prototype :</p> <pre><sys/types.h> <unistd.h> pid_t fork()</pre> <p><code>pid_t</code> est un type entier défini dans <code><sys/types.h></code></p> <p>Valeurs retournées :</p> <ul style="list-style-type: none">-1 en cas d'erreur. Nombre de processus usager ou système excédé. La variable <code>error</code> contient le numéro d'erreur <code>EAGAIN</code>.>0 identifiant le processus fils.0 retourné au processus fils.
<p><code>execv()</code> : Charger un programme dans l'espace adresse d'un processus</p>
<p>Prototype :</p> <pre><unistd.h> int execv(const char *path, char *const argv[]);</pre> <p><code>path</code> spécifie le programme à charger.</p> <p><code>argv</code> spécifie les arguments à passer au programme chargé.</p> <p><code>argv[0]</code> contient le nom du programme.</p> <p><code>argv[i]</code> contient les arguments.</p> <p>Valeurs retournées :</p> <p>Si <code>execv</code> retourne alors l'appel a échoué.</p> <p>Appel de <code>perror("execv n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>execvp()</code> : Charger un programme dans l'espace adresse d'un</p>

<p>processus</p> <p>Prototype : <code><unistd.h></code> <code>int execvp(const char *file, char *const argv[]);</code> file spécifie le programme à charger. Répertoire de recherche du programme sauvegardé dans la variable PATH <code>%PATH = /bin :/usr/bin :/usr/jmtorres %export PATH</code> <code>argv[0]</code> contient le nom du programme. <code>argv[i]</code> contient les arguments. Valeurs retournées : Si <code>execvp</code> retourne alors l'appel a échoué. Appel de <code>perror("execvp n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>execve()</code> : Charger un programme dans l'espace adresse d'un processus</p> <p>Prototype : <code><unistd.h></code> <code>int execve(const char *path, char *const argv[], char *const envp[]);</code> <code>path</code> spécifie le programme à charger. <code>argv[0]</code> contient le nom du programme. <code>argv[i]</code> contient les arguments. <code>envp</code> spécifie un environnement à substituer à l'environnement par défaut USER, PATH, HOME, LOGNAME, SHELL, TERM, etc. Valeurs retournées : Si <code>execve</code> retourne alors l'appel a échoué. Appel de <code>perror("execve n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>execl()</code> : Charger un programme dans l'espace adresse d'un processus</p> <p>Prototype : <code><unistd.h></code> <code>int execl(const char *path, char *const arg*0, ..., const char *argn, (char *)0);</code> <code>path</code> spécifie le programme à charger. <code>argi</code> ($i \geq 0$) spécifie une liste d'arguments à passer au programme à charger. Liste d'arguments terminée par NULL. <code>arg0</code> contient le nom du programme, <code>arg1, ..., argn</code> contient les arguments.</p>

<p>Valeurs retournées :</p> <p>Si <code>execl</code> retourne alors l'appel a échoué.</p> <p>Appel de <code>perror("execl n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>execlp()</code> : Charger un programme dans l'espace adresse d'un processus</p> <p>Prototype :</p> <pre><unistd.h> int execlp(const char *file, char *const arg*0, ..., const char *argn, (char *)0);</pre> <p><code>file</code> spécifie le programme à charger.</p> <p><code>argi</code> ($i \geq 0$) spécifie une liste d'arguments à passer au programme à charger. Liste d'arguments terminée par <code>NULL</code>.</p> <p><code>arg0</code> contient le nom du programme, <code>arg1, ..., argn</code> contient les arguments.</p> <p>Valeurs retournées :</p> <p>Si <code>execlp</code> retourne alors l'appel a échoué.</p> <p>Appel de <code>perror("execlp n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>execle()</code> : Charger un programme dans l'espace adresse d'un processus</p> <p>Prototype :</p> <pre><unistd.h> int execle(const char *path, char *const arg*0, ..., const char *argn, (char *)0, char *const envp[]);</pre> <p><code>path</code> spécifie le programme à charger.</p> <p><code>argi</code> ($i \geq 0$) spécifie une liste d'arguments à passer au programme à charger. Liste d'arguments terminée par <code>NULL</code>.</p> <p><code>arg0</code> contient le nom du programme, <code>arg1, ..., argn</code> contient les arguments.</p> <p>Valeurs retournées :</p> <p>Si <code>execle</code> retourne alors l'appel a échoué.</p> <p>Appel de <code>perror("execle n'a pas pu exécuter le programme")</code> pour afficher un message d'erreur.</p>
<p><code>exit()</code> : Terminaison et nettoyage d'un processus</p> <p>Prototype :</p> <pre><stdlib.h> int atexit(void (*func) (void));</pre> <p><code>atexit</code> enregistre la foction pointée par <code>func</code>.</p>

<p>func ne doit pas avoir d'arguments. Fonctions exécutées en ordre inversé d'enregistrement. Valeurs retournées : Indication du succès de l'enregistrement de la fonction.</p>
<p>atexit() : Définition par l'utilisateur d'opérations de terminaison</p> <p>Prototype : <stdlib.h> int atexit(void (*func) (void)) ; atexit enregistre la fonction pointée par func. func ne doit pas avoir d'arguments. Fonctions exécutées en ordre inversé d'enregistrement. Valeurs retournées : Indication du succès de l'enregistrement de la fonction.</p>
<p>wait() : Synchronisation d'un processus parent avec ses processus fils.</p> <p>Prototype : <sys/types.h> <sys/wait.h> pid_t wait(int *status) ; status contient le code de statut de terminaison du processus fils. Valeurs retournées : Numéro de processus (pid) du premier processus fils terminé. -1 si le processus n'a aucun processus fils. La variable error contient le code d'erreur ECHILD.</p>
<p>waitpid() : Synchronisation d'un processus parent avec un processus fils spécifique.</p> <p>Prototype : <sys/types.h> <sys/wait.h> pid_t waitpid(pid_t pid, int *status, int options) ; pid spécifie le numéro du processus fils. status contient le code de statut de terminaison du processus fils. options est habituellement initialisé à WNOHANG. Processus parent ne bloque pas durant l'exécution du processus fils, et waitpid retourne 0. Si pid = -1 et options = 0 alors waitpid a le même comportement que wait. Valeurs retournées :</p>

<p>Numéro de processus fils ayant terminé. -1 si le processus n'a aucun processus fils. La variable <code>error</code> contient le code d'erreur <code>ECHILD</code>.</p>
<p><code>getpid()</code> : Détermination du numéro d'un processus.</p>
<p>Prototype : <sys/types.h> <unistd.h> pid_t getpid(void); Valeurs retournées : Numéro du processus appelant.</p>
<p><code>getppid()</code> : Détermination du numéro du parent d'un processus.</p>
<p>Prototype : <sys/types.h> <unistd.h> pid_t getppid(void); Valeurs retournées : Numéro du parent du processus appelant.</p>
<p><code>pthread_create()</code> : Création d'un thread dans l'espace adresse d'un processus</p>
<p>Prototype : <pthread.h> int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_routine)(void*), void *arg); thread pointe vers une adresse contenant le ID du nouveau thread. attr spécifie les attributs du thread. Si attr est NULL alors les attributs par défaut sont utilisés. Valeurs retournées : 0 en cas de succès de création du thread. Un code d'erreur en cas d'échec de création du thread. Nombre exécutaire de threads, attribut invalide, permissions insuffisantes, etc.</p>
<p><code>pthread_exit()</code> : Terminaison d'un thread.</p>
<p>Prototype : <pthread.h> void pthread_exit(void *value_ptr); value_ptr est disponible à tout thread se synchronisant avec le</p>

<p>thread terminé. Valeurs retournées : Aucune.</p>
<p><code>pthread_join()</code> : Synchronisation du thread appelant avec un autre thread.</p>
<p>Prototype : <pthread.h> <code>int pthread_join(pthread_t thread, void **value_ptr);</code> thread est un pointeur vers l'ID du thread dont la terminaison est attendue. value_ptr est passé à <code>pthread_exit()</code> Valeurs retournées : 0 en cas de succès de synchronisation avec un thread. Un code d'erreur en cas d'échec de synchronisation avec un thread. Pas de synchronisation possible avec le thread spécifié, ID inexistant du thread spécifié, etc.</p>
<p><code>pthread_detach()</code> : Indication au système que l'espace adresse du thread spécifié peut être déalloué à sa terminaison.</p>
<p>Prototype : <pthread.h> <code>int pthread_detach(pthread_t thread);</code> thread est un pointeur vers l'ID du thread. Valeurs retournées : 0 en cas de succès. Un code d'erreur en cas d'échec de synchronisation avec un thread. Pas de synchronisation possible avec le thread spécifié, ID inexistant du thread spécifié, etc.</p>
<p><code>pthread_attr_init()</code> : Initialisation des attributs par défaut d'un thread.</p>
<p>Prototype : <pthread.h> <code>int pthread_attr_init(pthread_attr_t *attr);</code> attr est initialisé avec tous les attributs par défaut spécifiés pour un système donné. Valeurs retournées : 0 en cas de succès d'initialisation de l'objet attr. Un code d'erreur en cas d'échec d'initialisation de l'objet attr.</p>

Insuffisance de mémoire.

`pthread_attr_destroy()` `destroy()` : Effacement des attributs par défaut d'un thread.

Prototype :

`<pthread.h>`

```
int pthread_attr_destroy(pthread_attr_t *attr);
```

`attr` est effacé.

Valeurs retournées :

0 en cas de succès d'initialisation de l'objet `attr`.

Un code d'erreur en cas d'échec d'initialisation de l'objet `attr`.

`sleep()` : Dormir pendant quelque temps.

Pause de `NUMBER` secondes. `SUFFIX` peut-être en `s` secondes, `m` minutes, `h` heures ou `d` journées.

Prototype :

```
sleep [OPTION]...NUMBER[SUFFIX]
```


Annexe C

Signaux

Signaux Linux

Numéro	Nom	Action	Description
1	SIGHUP	exit	Terminaison de la liason avec le terminal
2	SIGINT	exit	Interruption en provenance du terminal (<ctrl C>)
3	SIGQUIT	core	Quitter en provenance du terminal (<ctrl \>)
4	SIGILL	core	Instruction illégale
5	SIGTRAP	core	Trace ou trap de point de débogage
6	SIGIOT		
7	SIGBUS	core	Erreur de bus (adresse mal alignée)
8	SIGFPE	core	Exception d'arithmétique point flottant
9	SIGKILL	exit	Élimination (pas d'attrapage et masquage)
10	SIGUSR1		Signal1 défini par l'utilisateur
11	SIGSEGV	core	Faute de segmentation (référence mémoire)
12	SIGUSR2	exit	Signal défini par l'utilisateur
13	SIGPIPE	exit	Tube sans processus récepteur
14	SIGALRM	exit	Déclenchement de l'alarme (setitimer, alarm)
15	SIGTERM	exit	Terminaison envoyée par la commande kill
17	SIGCHLD	ignore	Chargement d'état du processus enfant
18	SIGCONT		Reprise
19	SIGSTOP	stop	Arrêt (pas d'attrapage et masquage)
20	SIGTSTP	stop	Arrêt (<ctrl Z>)
21	SIGTTIN	stop	Lecture au terminal par processus en arrière plan (arrêt)

22	SIGTTOU	stop	Ecriture au terminal par un processus en arrière plan (arrêt)
23	SIGURG	ignore	Données réseau urgentes
24	SIGXCPU	core	Expiration du temps UCT (getrlimit)
25	SIGXFSZ	core	Dépassement de taille de fichier (getrlimit)
26	SIGVTALRM	exit	Déclenchement d'un temporisateur virtuel (setitimer)
27	SIGPROF	exit	Déclenchement d'un temporisateur profilé (setitimer)
28	SIGWINCH	ignore	Changement de taille de la fenêtre
29	SIGIO		
30	SIGPWR	ignore	Panne d'alimentation ou réinitialisation

Signaux Unix

Numéro	Nom	Action	Description
1	SIGHUP	exit	Terminaison de la liaison avec le terminal
2	SIGINT	exit	Interruption en provenance du terminal (<ctrl C>)
3	SIGQUIT	core	Quitter en provenance du terminal (<ctrl \>)
4	SIGILL	core	Instruction illégale
5	SIGTRAP	core	Trace ou trap de point de débogage
6	SIGABRT	core	Annulation par un appel à abort (...)
7	SIGEMT	core	Trap d'émulation
8	SIGFPE	core	Exception d'arithmétique point flottant
9	SIGKILL	exit	Élimination (pas d'attrapage et masquage)
10	SIGBUS	core	Erreur de bus (adresse mal alignée)
11	SIGSEGV	core	Faute de segmentation (référence mémoire)
12	SIGSYS	core	Appel système erroné
13	SIGPIPE	exit	Tube sans processus récepteur
14	SIGALRM	exit	Déclenchement de l'alarme (setitimer, alarm)
15	SIGTERM	exit	Terminaison envoyée par la commande kill
16	SIGUSR1	exit	Signal défini par l'utilisateur
17	SIGUSR2	exit	Signal défini par l'utilisateur
18	SIGCHLD	ignore	Chargement d'état du processus enfant
19	SIGPWR	ignore	Panne d'alimentation ou réinitialisation

20	SIGWINCH	ignore	Changement de taille de la fenêtre
21	SIGURG	ignore	Données réseau urgentes
22	SIGPOLL	exit	Événement sur un périphérique en polling
23	SIGSTOP	stop	Arrêt (pas d'attrapage et masquage)
24	SIGTSTP	stop	Arrêt (<ctrl Z>)
25	SIGCONT	ignore	Continuation (<ctrl Q>)
26	SIGTTIN	stop	Lecture au terminal par processus en arrière plan (arrêt)
27	SIGTTOU	stop	Écriture au terminal par un processus en arrière plan (arrêt)
28	SIGVTALRM	exit	Déclenchement d'un temporisateur virtuel (setitimer)
29	SIGPROF	exit	Déclenchement d'un temporisateur profilé (setitimer)
30	SIGXCPU	core	Expiration du temps UCT (getrlimit)
31	SIGXFSZ	core	Dépassement de taille de fichier (getrlimit)

Annexe D

Sites Internet

- Site du cours à l'École Polytechnique de Montréal
 - <http://www.cours.polymtl.ca/inf3600/>
- Site Internet de GNU
 - <http://www.gnu.org>
- Sites pour les développeurs Linux
 - <http://developer.kde.org>
 - <http://www.linuxprogramming.com>
 - <http://www.blackdown.org>
 - <http://developer.gnome.org>
 - <http://www.perl.com>
 - <http://java.sun.com>
- Site du serveur Web Apache pour Linux/Unix
 - <http://www.apache.org>
- Sites des distributions et des noyaux Linux
 - <http://www.redhat.com>
 - <http://www.calderasystems.com>
 - <http://www.suse.com>
 - <http://www.infomagic.com>
 - <http://www.linuxppc.com>
 - <http://www.slackware.com>
 - <http://www.linux-mandrake.com>
 - <http://www.java.sun.com>
 - <http://www.kernel.org>
 - <http://www.debian.org>
 - <http://www.linux.org>
- Sites des bureaux et des gestionnaires de fenêtres Linux

- <http://www.gnome.org>
- <http://www.kde.org>
- <http://www.x11.org>
- <http://www.fvwm.org>
- <http://www.windowmaker.org>
- <http://www.enlightenment.org>
- <http://www.xfree86.org>
- <http://www.themes.org>
- Site sur les appels système Unix
 - <http://www.opennc.org/onlinepubs/007908799/>
- Sites sur des systèmes d'exploitation divers
 - <http://www.apple.com/macosx>
 - <http://www.microsoft.com>
 - <http://www.qnx.org>
 - <http://www.freebsd.org>
 - <http://www.novell.com>
 - <http://www.palmos.com>
 - <http://www.sun.com>
 - <http://www.samba.com>
 - <http://www.beos.com>
 - <http://www.ibm.com>
 - <http://www.free-vms.org>
- Sites sur le simulateur Nachos
 - <http://www.cs.duke.edu/narten/110/nachos/main/main.html>
 - <http://www.cs.washington.edu/homes/tom/nachos/>