

UNIX

3il

Historique :

1971 : ancêtre Multics des laboratoires bell (filiale de ATT), création d'UNIX en assembleur

1973 : portage d'UNIX en C (90 % du noyau)

Présentation des créateurs d'Unix : Ken thompson et Dennis Ritchie (inventeur du C)

1991 : Linus Torvalds annonce Linux (vient de minix)

Présentation des différents types d'Unix et normes (BSD(Berkeley), SYSTEM 5, POSIX)

Historique LINUX (vient de Minix de Andrew tanebaum, puis licence GPL General Public Licence)

Unix et Linux (SOLARIS(SUN), OpenServer (SCO), AIX(IBM), MANDRIVA, REDHAT,UBUNTU...)

Le standard : outils communs sur tous les unix/linux : exemple vi

Architecture processeur :

SPARC M8 : Gravée en 20 nanomètres (nm), la puce dispose de 32 cœurs V9 avec une fréquence qui peut monter à 5 GHz, soit 256 threads (8 flux d'instructions par cœur), 64 Mo de cache L3 par processeur et 128 Ko de cache L2 par cœur.

Power9 : 24 coeurs . Il bénéficie de la technologie de multithreading SMT8 avec 8 threads par coeur soit un total de 96 threads, 120mo de cache L3.

Le power 10 arrive en 7nm (debut 2021)

Cat /proc/cpuinfo

Linux aujourd'hui : part de marché qui grimpe (voir administration par exemple) + top500 (100 % des systèmes sont maintenant sous linux)

linux=ligne de commande, interface graphique independant (non natif)

linux = partout (steambox, android, tv, gros systeme ...)

BASH intégré dans windows 10

Tester la commande man

uname -a : caractéristiques du système

Architecture

Unix : système multi-taches multi-utilisateurs (schema)

Linux : système multi-taches mono-utilisateur (schema)

Types processeurs supportés (UNIX = RISC et LINUX = CISC et RISC)

Systèmes de fichiers supportés (Ext3fs, jfs, = fragmentation (gestion table des fragments libre... alors que ntfs fragmente)

Le système de fichier est propriétaire (exemple : AIX(IBM) est jfs (journalized file system)

LINUX est ext4fs, UFS (Unix File System) pour solaris) et donc non standard

/ : Répertoire racine. Point de départ de toute la hiérarchie du système de fichiers.

/boot/ : Répertoire contenant le noyau de Linux et ses informations de symboles.

/dev/ : Répertoire contenant tous les fichiers spéciaux permettant d'accéder aux périphériques.

/sbin/ : Répertoire contenant les commandes systèmes nécessaires à l'amorçage et réservées à l'administrateur.

/bin/ : Répertoire contenant les commandes systèmes générales nécessaires à l'amorçage.

/lib/ : Répertoire contenant les bibliothèques partagées

/lib64/ : architecture x86_64

/etc/ : Répertoire contenant tous les fichiers de configuration du système.

/tmp/ : Répertoire permettant de stocker des données temporaires.

/usr/ : Répertoire contenant les fichiers du système

/var/ : Répertoire contenant toutes les données variables du système.

/var/log/messages, /var/log/xorg.log, /var/log/dmesg

/home/ (/users/ sous unix) : Répertoire contenant les répertoires personnels des utilisateurs.

/root/ : Répertoire contenant le répertoire personnel de l'administrateur.

/media/ : Répertoire réservé au montage des systèmes de fichiers sur périphériques amovibles (CD-ROM, disquettes, etc.).

/mnt/ : Répertoire réservé à l'administrateur pour le montage temporaire des systèmes de fichiers.

/lost+found/ : Répertoire contenant les données récupérées lors de la réparation d'un système de fichiers endommagé.

/proc/ : Répertoire contenant le pseudo système de fichiers du noyau.

[Cd , cd .., cd-, cd / : se déplacer dans les répertoires](#)

Interface graphique

Le système Linux est différent du système Windows. Linux est un système qui fonctionne en ligne de commande.

Windows : interface graphique unique et natif

Linux : interface graphique multiple (GNOME, KDE, XFCE, Unity, compiz ..) et non natif

Lancement de l'interface graphique : startx

[startx : lancer le mode graphique](#)

Shutdown de l'interface graphique : CTRL+ALT+BACKSPACE (à réactiver sur les dernières versions).

Quelques outils spécifiques : kedit (éditeur kde), gedit (éditeur gnome)
mais on trouve des outils spécifiques qui fonctionnent sur tous les environnements (kaffeine)

Outils couplé au système UNIX

Tuxedo (serveur transactionnel UNIX), apache (serveur web), samba (Contrôleur principal de domaine), ORACLE (SGBDR)
GIMP (alternative à PHOTOSHOP), OpenOffice (Libre Office), IpTable (firewall ->ipcop), ECLIPSE (java), Firefox (navigateur), Thunderbird (messagerie),
LAMP (Linux Apache Mysql Php), moniteur système (gnome-system-monitor)

Caractéristiques d'un utilisateur UNIX :

Login (nom de connexion), password (mot de passe), UID (User Identifier), GID (Group Identifier), Home (répertoire de travail), SHELL (interpréteur de commande par défaut)
On retrouve ces caractéristiques dans le fichier /etc/passwd

cat /etc/passwd

Forme generale d'une commande unix : commande -[argument1] .. -[argument n]
Reaction aux commandes : l'interpreteur cherche la commande dans le systeme, si il la trouve ul l'execute sinon il renvoie : command not found

Commande pour entrer/sortir du système

login : entrer dans la système
logout : sortir du système
who : qui est connecté sur le système

Afficher les utilisateurs connectés sur le système

.bash_profile, .bash_history, .bash_rc : fichiers caches, demarrage session

Ajouter une commande echo dans le .bash_profile et dans le .bash_rc

Commandes du systeme de fichiers

find : cherche quoi,ou,comment,qu'en faire
echo : affiche à l'ecran (ou redirection dans un fichier)
cat : affiche le contenu d'un fichier
pwd : indique le chemin en cours
cd : changement de repertoire (cd, cd-,cd /)
mkdir : creation de repertoire

rmmdir : suppression de répertoire
rm : suppression de fichier (rm -i, rm -r, rm -f, rm -rf)
ls : lister les fichiers (-al)
cp : copie de fichier (-p , -r)
mv : renommer un fichier
file : indique le type du fichier
ln : creation de lien sur un fichier
uname: informations systeme
find : recherche de fichier personnel (-perm, -size, -mtime ...)
whereis: recherche de fichier systeme

Options spéciales : -r -f

Rappeler les commandes : flèche directionnelles

Completion des commandes : touche tab

ls -l |more (pipe)

Lister les fichiers avec et sans la couleur

Lister les fichiers par taille

(debian) installation via paquet : dpkg -i,-r,-u ou apt-get [install,remove,update] [paquet .deb]
via le réseau

apt-get install sl cow, supertux

(redhat) installation via paquet : rpm -i -e -U [paquet] ou dnf (yum deprecated) [install,
remove, via le reseau
Via le reseau , les dependances sont résolues.

*copier un fichier, renommer un fichier, déplacer un fichier, supprimer un fichier, creer un
répertoire, supprimer un répertoire vide, supprimer un répertoire non vide*

Rechercher le fichier resolv.conf,

Rechercher les fichiers sous /etc > 30k

Rechercher dans quel fichier se situe la chaine de caractère _XOPEN

Les fichiers

Sous unix , tout est fichier

La longueur d'un nom de fichier sous UNIX peut aller jusqu'à 255 caractères

Notion de chemin relatif / Chemin Absolu

2 types de fichiers : les fichiers normaux, les fichiers speciaux (/dev)

L'extension n'est pas significative

60

EXT4FS : Gestion des volumes jusqu'à un 1000 tera (2⁶⁰ octets), taille maximale d'un fichier
= 16to, allocation par extent qui permettent la pré-allocation d'une zone contiguë pour un
fichier, pour minimiser la fragmentation. Basé sur les listes (doublement) chaînées, la
journalisation, les droits, les inodes (numéro unique comme UID ou PID)

ls -al

Les informations des inodes

- Type de fichier : -,d, l, c, p, b.
- Droits d'accès : par exemple : rwxr-x---
- Nombre de liens (physiques) : correspond au nombre de références c'est à dire au nombre de noms.
- [UID](#) : effectif du processus créateur ou affecté par chown.
- [GID](#) : effectif du processus créateur, affecté par chgrp ou hérité du répertoire.
- Taille du fichier.
- atime :date de la dernière lecture.
- mtime :date de la dernière modification.
- ctime :date de la dernière connexion.
- Adresse du fichier.

Le superbloc contient des informations sur l'espace utilisé dans la partition, la liste des inodes. Il est dupliqué plusieurs fois.

Editeurs

Vi : éditeur standard

Emacs, gedit, kedit,nano : éditeurs non standards

Mode insertion de vi : i,a,o,A,O

Mode commande : touche ESC (escape)

! : sortir dans le shell en restant dans vi

q : sortir

w : sauvegarde

12 : aller à la ligne 12

\$: fin de fichier

1 : début de fichier

u : annule la dernière action

r : remplace un caractère

yy,p : copier coller

dd : supprimer

12 dd : supprimer 12 lignes (les lignes sont copiees avant suppression)

set nu : numérotation des lignes

set nonu : supprime la numérotation

set list : affiche les caractères ACSII

set nolist : supprime l'affichage des caractères ASCII

/ pour la recherche avec n pour le suivant et N pour le precedent

x pour suppression de caractere

dd pour suppression de lignes + indice pour le nombre de lignes

substitutuion : :%s/chaine1/chaine2/g

affichage hexadecimal : :%!xxd

/temp : système de fichier ou vi trouve la place de l'édition

Créer un fichier avec vi, de la forme :

nom prénom

date de naissance

adresse

téléphone

puis :

sauvegarder, quitter, revenir dans le fichier, remplacer une lettre de l'adresse par une autre puis supprimer la. Copier la ligne du téléphone, coller la, copier l'ensemble des lignes puis collez, déplacez-vous dans le fichier (début-fin), supprimer les lignes en double, rechercher le dernier numéro du téléphone, remplacer tous les e par des E puis afficher les numéros de lignes et les caractères non-affichables.

Caractères génériques

? : remplace un caractère quelconque

*: remplace une chaîne de caractères quelconques

[] : définit un intervalle ou un ensemble de caractères

! : indique la négation

« « : protège le contenu

` : non évaluation de la chaîne

` : protège la commande

^ : début de ligne

\$: fin de ligne

& : lancement batch

: commentaires

| : concaténation de commande (explication de more et tube)

\ : déspecialise la commande

Afficher la liste des périphériques de type tty(ou ram) de 3 a 5

Afficher la liste des tty de 20 à 30

Affichez la liste des périphériques de type tty(ou ram) de 0 a 9 (autre méthode)

Droits d'accès

Tout fichier dispose de :

- types d'accès
- droits d'accès

Les droits d'accès :

USER (le propriétaire), GROUP (le groupe), OTHERS (les autres) -> UGO

Les types d'accès :

READ (lecture), WRITE (écriture), EXECUTE (exécution)

Au niveau du repertoire, les droits d'accès se traduisent par :

READ (lecture) : lister le repertoire

WRITE (écriture) : creation/destruction d'un fichier dans ce repertoire

EXECUTE (exécution) : droit de traverser le repertoire

SUID, SGID, et sticky bit

SUID : execute sous l'identité du propriétaire du fichier, le x de execute est remplacé par un s, un S (majuscule) indique que le droit x masqué n'est pas activé.

SGID : execute sous l'identité du groupe, le x est remplacé par un s mais au niveau du groupe, S idem

Sticky : Image binaire reste en mémoire, un t apparaît a la place du x, T idem

Voir l'exécutable passwd sous /usr/bin/passwd pour explications.

Ces trois droits spéciaux ne s'appliquent qu'aux executables et non aux fichiers normaux et scripts

Modification des droits d'accès

Prototype : Chmod [mode] [fichier]

+ pour ajouter

- pour enlever

droits d'accès par défaut :

umask

changement de propriétaire :

prototype : chown [owner] [fichier]

changement de groupe :

prototype : chgrp [group] [fichier]

Seuls root et le propriétaire possède les droits de manipuler ces commandes.

Modifiez les droits d'accès de votre .bash_profile pour ne garder que des droits du user (le créer s'il n'existe pas)

Chercher les fichiers disposant des droits pour tous le monde

Créer un répertoire nommé "Secret", dont le contenu est visible uniquement par vous même

Les variables d'environnement

Deux types de famille : les variables modifiables et les non-modifiables (système et utilisateurs)

Les variables d'environnement permettent de définir l'espace de travail.

.profile : fichier chargé pour l'utilisateur lors d'un login (.bash_profile sous linux)

/etc/profile : fichier pour l'ensemble des utilisateurs

set, env : commandes permettant de lister les variables d'environnement

export [variable] permet de positionner une variable d'environnement

variables : TERM,PATH,PS1,USER, LOGNAME,SHELL,\$BASH ...

echo \$BASH pour voir le type de shell utilisé

variable TIMEOUT : deconnexion au bout de timeout, attention readonly pour le root

Afficher le type de : terminal, shell,user

Positionnez une nouvelle variable d'env nommée ECOLE et ayant pour valeur 3il, afficher

Redirection des sorties standards

>[fichier] : si le fichier existe il est supprimé et crée

>> [fichier] : la Redirection écrit à la fin du fichier

Comment copier un fichier sans la commande cp

Ajouter une ligne à la fin d'un fichier existant utilisant les redirections

les descripteurs de fichier (pointeur vers un fichier)

0 : entrée standard (clavier)

1 : sortie standard (écran)

2 : standard erreurs (écran)

Recherchez le fichier services sans aucune erreur affichée

code de retour et paramètres

\$# : nombre d'arguments

\$* : liste des arguments

\$0 : nom de la commande

\$1 : premier argument
\$2 : deuxième argument ...
\$\$: PID du dernier processus appelant
\$? code de retour (appel systeme, fonction)

Créer un script qui prend comme argument un fichier, le script doit afficher le contenu du fichier, afficher le nombre d'arguments et la liste

Linux peut lire/écrire sur des partitions NTFS via Samba

exemple de montage d'une clef usb :

```
mount -t vfat ' /dev/sd[ ??] /media/usb ( après avoir créer le répertoire /media/usb )
```

exemple de montage de type samba :

```
mount -t cifs //[NomPC]//[NOMDEPARTAGE WINODWS] /[REPERTOIRE OU ON MONTE LE FS] -o username=[DOMAINE]/[USER],uid=[UID],gid=[GID],rw ( rw pour lecture/écriture )
```

L'utilisateur doit exister à la fois sur windows et sur linux, même compte et même mot de passe

vfat pour les clefs de petite capacité

exfat pour les clefs de grande capacité

Monter un système de fichier windows depuis le système linux

Monter une clef usb

Modifier le prompt afin qu'il affiche nomdeloginencours_cheminencours>

Les processus

Programme : suite d'instructions statiques

Processus : entité évoluant dynamiquement dans le temps

mode noyau, mode utilisateur

Notions sur les pages : pas d'accès aux pages réelles (mémoire réelle), seulement virtuelles

DAT : Dynamic Address Translation, circuit qui effectue la conversion adresse réelle/virtuelle

TLB : Translation Lookaside Buffer, une table de correspondance numéro de page réelle/virtuelle

Différence entre processus et threads : processus = fork + exec alors que thread déjà chargé dans environnement du processus.

Notions sur les threads

Liste des processus : ps (table des processus)

Caractéristiques d'un processus :

PID : Process Identifier

PPID : Parent Process Identifier

PGID : Process Group Identifier

SID : Session Identifier

TTY : indique le terminal de lancement du processus

TIME : temps CPU consommé

Etat d'un processus

R : prêt à s'exécuter

T : stoppé

P : en attente de page

D : en attente non interruptible

S : dormant depuis - de 20s

I : dormant depuis + de 20s

Z : zombie

Commande pour envoyer un signal à un processus : kill

Liste des signaux : kill -l

Exemple de signal : SIGINT : CTL C, SIGKILL : terminaison non masquable, SIGCHLD = fin d'un fils

Quel est le numéro du signal SIGPWR

Un processus afin d'être ordonnancé possède une priorité. Plus la priorité est grande moins le processus s'exécute vite. ($P = P + \text{temps CPU} / 2$)

Lister les processus du terminal tty1 seulement, tuer bash

Modification de la priorité d'un processus : nice

Ps gv : activité du système vu des processus

Vmstat :activité complète du système

Performances du système : 8 analyses toutes les 3s

Les démons (ou services) : processus tournant continuellement sur le système , depuis le démarrage et qui ferme les descripteurs de fichiers 0,1,2. Noms se terminant par d (crond, httpd ...)

Voir pstree sous linux pour l'arborescence des processus et donc INIT, le premier

Le démon CRON

CRON est prévu pour planifier des commandes à une date et une heure donnée.

Signification des champs de CRON :

(à présenter en ligne pour la compréhension)

Minutes : 0-59

Heures : 0-23

Jour du mois : 1-31

Mois : 1-12

Jour de la semaine : 0-6 (avec 0 pour dimanche)

Commande à planifier

Soumission du fichier au démon : crontab

Liste des commandes soumises à CRON : crontab -l

Suppression des commandes soumises à CRON : crontab -r

La nécessité d'un interface graphique doit passer par xhost : exemple xhost + local:root

Traitement en mode batch

Deux modes de lancement d'un exécutable : mode foreground, mode background

Mode foreground : c'est le mode interactif

Mode background : on lance le processus en arrière plan

Créer un script qui dort 10s puis écrit la date, format ZULU, de la veille au format américain (date +%Y%m%d --date='-1day', Date=TZ='Zulu';date +"%Y-%m-%dT%H:%M:%SZ") dans un fichier, lancer le en avant/arrière plan

Prévoir l'exécution du script dans cron

& est le caractère permettant d'utiliser le mode background

jobs est la commande permettant de lister les processus en arrière plan

commande nohup (signal SIGHUP) pour déconnexion sans quitter le processus

CTRL Z pour tuer

Déroutement des signaux

Trap est la commande permettant de dérouter un signal

Prototype : trap ' ' signal

Reprendre le script, empêcher son interruption grâce au CTRL C (sans affichage de message)

Même chose que la question précédente mais en affichant un message

commandes à voir : du,df

TAR et GZIP

TAR : Tape Archive

Création d'archives pour regrouper des fichiers de même type et/ou ensemble

Mode création : tar -cvf toto.tar *.c

-c :création

-v : mode verbose

-f : attente d'un fichier

Mode extraction : tar -xvf toto.tar

-x : extraction

GZIP : compression de données

Gzip [fichier] -> fichier .gz

Gzip -d : décompression (supprime l'extension)

Archive compressée : .tar.gz ou .tgz

Nouveau compresseur linux : bz2

Créer une archive de l'ensemble des fichiers .conf de /etc appelée conf.rar,
compressez la puis extraire le tout sous /home/eleve/conf

Cpio : copy input output

Exemple : find /usr -cpio /dev/rmt0 -> copie le système de fichiers /usr sur le lecteur de bande rmt0

dd : copie en mode image

Mksysb : make system bootable -> permet de copier un système bootable sur bande
voir g4l pour linux (équivalent de clonezilla)

DLT et SDLT : sauvegarde sur bande

DLT : Digital Linear Tape

DLT-S4 : 800 Go de sauvegarde

IPC

IPC : Inter Process Communication

Les IPC permettent la communication entre les processus. Il existe 4 types d'IPC.

- les segments de mémoire partagée
- les sémaphores
- les files de messages
- les pipes

La commande `ipcs` permet de visionner les trois premières

La commande `ipcrm` permet de les supprimer

Lister les ipc sur le système

`bash` pour linux : bourne again shell, mais aussi

`ksh` : korn shell unix

`ssh` : secure shell

`ssh -l username hostname [commandname]` qui établit une communication sécurisée vers le compte Unix "username" ouvert sur la machine dont l'adresse réseau est "hostname". Si l'on introduit le paramètre optionnel "commandname", le système SSH se contentera d'y exécuter la commande demandée. Dans le cas contraire, il ouvrira sur la machine distante une session de type rlogin. Ce protocole est crypté.

`rssh` : restricted secure shell

`scp` : copie de fichier sur systemes distants : `user@]host1:file1 user@host2:file2`

Notion de réseau

TCP/IP : Transport Control Protocol / Internet Protocol

Protocoles Unix :

FTP : File Transfert Protocol

(sftp existe aussi)

automatisation de transfert FTP

dans un fichier `.netrc` ayant les droits suivants : `rw-----` car mot de passe dedans

```
machine mymachine
login myusername
password mypassword
```

```
macdef mydef
```

```
cd /pub/tests
bin
put filename.tar.gz
quit
```

ftp mymachine va déclencher la recherche d'un fichier .netrc et si le nom de machine est correct lancé le script de transfert.

TELNET : Terminal Network Protocol (accéder à des switches, routeur ou autre système)

NFS : Network File System

/etc/exports est indispensable, mais à moins de vouloir faire de votre réseau une vraie passerelle, il faut aussi éditer deux autres fichiers (dont l'intérêt est évident) :

/etc/hosts.allow (@IP autorisées) et /etc/hosts.deny (à IP non autorisées)

/etc/exports : /toto machine1(rw) machine2(ro) permet de partager le répertoire /toto pour la machine machine1 en lecture écriture et à la machine machine 2 en lecture seule.

Prise en compte du fichier /etc/exports: lancer un examen de /etc/exports par nfs, /sbin/service nfs reload par exemple

[activer/désactiver la carte réseau](#)

[Exécuter la commande eject -t sur un système distant](#)

[Copier le fichier /etc/services sur une machine distante](#)

[Ouvrir /etc/passwd sur un système distant en mode graphique](#)

Fichiers de configuration

/etc/hosts : résolution de noms / adresses IP

/etc/resolv.conf : résolution de noms via DNS

/etc/networks : description des réseaux disponibles

/etc/services : description des services réseaux

/etc/protocols : description des protocoles

affichage des éléments du réseau : ifconfig (ou /sbin/ifconfig pour ceux qui n'ont pas les droits)

ifconfig up : activation de l'interface réseau, en cours de dépréciation (pas sur BSD par exemple mais par sur Centos), remplacé par ifup/ifdown

ifconfig down : désactivation de l'interface réseau

Expressions régulières

Grep -v, -w

head, tail -n

cut -c, -d

wc -c, -l

sort

[Afficher l'UID de l'utilisateur IRC uniquement](#)

Afficher l'avant dernière ligne de /etc/passwd seulement

Afficher par ordre alphabétique les utilisateurs définis dans le fichier

/etc/passwd

Combien de lignes, de mots et de caractères comportent le fichier

/etc/passwd

Langage script

Initialisation de variable : a = 1

Echo \$a

le shell ne peut voir QUE des caracteres, jamais des entiers ou autre chose.

Man test et non man if

&& et || sont des operateurs bianires

deux types de test

- test sur les fichiers : -s (fichier existe et non vide), -f (fichier existe et est regulier) ...
- test sur les variables : -eq (equal ...)

le then doit etre sur une nouvelle ligne (sinon ;)

Commande test : if

If [condition]

Then liste de commandes

Else liste de commandes

Fi

ou encore

if [condition]

then liste de commandes

elif [condition]

then liste de commandes

else liste de commandes

fi

deuxieme maniere de noter les if :

```
if test $PWD = $HOME
```

```
    then echo "Vous êtes dans votre répertoire d'accueil."
```

```
elif test $PWD = $HOME/bin
```

```
    then echo "Vous êtes dans votre répertoire d'exécutables."
```

```
elif test $PWD = $HOME/bin/solaris
```

```
    then echo "Vous êtes dans votre répertoire d'exécutables  
pour Solaris."
```

```
else echo 'Vous êtes dans un répertoire quelconque, qui n'est
ni $HOME,'
    echo 'ni $HOME/bin, ni $HOME/bin/solaris. '"Vous êtes dans
$PWD."
fi
```

Tester l'existence et la non nullité d'un fichier passé en paramètre

Initialiser une variable et tester si cette valeur est inférieur ou égale à 12

Tester existence du repertoire `/usr/lib64/`

man test pour l'aide

Expr : expression permettant de manipuler des entiers

`A=`expr $a + 1``

ou alors

`A=$((a+b))` mais mauvaise interpretation en cas d'erreur de saisie

Saisir une valeur pour une variable (read) et effectuer des opérations

Tester la multiplication

While : boucle

While [condition]

Do

Liste de commandes

done

For : boucle

For [variable] in [liste de valeurs]

Do

Liste de commandes

done

`a=0`, tant que cette valeur est inférieure à 7, afficher la

exemple : `for i in `ls``

`do`

`echo $i`

`done`

Tester la boucle sur les fichiers

ou alors

`For [variable] in $(seq [valeur de fin de boucle])`

do
Liste de commandes
done

Tester une boucle avec une valeur finie (exemple : 48)

Until : boucle
Même chose que While à l'itération prêt

Même chose que le switch du langage C

```
case chaîne in
    motif) commandes ;;
    motif) commandes ;;
esac
```

exemple :

```
case "$PWD" in
    "$HOME") echo "Vous êtes dans votre répertoire d'accueil.>";;
    "$HOME/bin") echo "Vous êtes dans votre répertoire
d'exécutables.>";;
    "$HOME/bin/solaris") echo "Vous êtes dans votre répertoire
d'exécutables pour Solaris.>";;
    "$HOME/prive") echo "Vous êtes dans votre répertoire privé.>";;
    "**") echo 'Vous êtes dans un répertoire quelconque, qui n'est
ni $HOME, '
        echo 'ni $HOME/bin, ni $HOME/bin/solaris. '"Vous êtes dans
$PWD.>";;
esac
```

On peut utiliser des caractères génériques, * représentant le défaut du langage C

Une fonction est déclarée en début de script

Déclaration d'une fonction :

```
nom()
{
}
```

passage de paramètre à une fonction : toto 3 (on passe le paramètre de valeur 3 à la fonction toto)
on le récupère avec \$1

On peut utiliser return n avec n un entier pour tester les codes de retour

Le programme doit commencer par la ligne suivante (première ligne du fichier) :

```
#!/bin/bash ( sous linux ) ou /bin/ksh ( sous unix )
```

Les caractères `#!` indiquent au système que l'argument suivant est le programme à utiliser pour exécuter ce fichier. Ici, nous utilisons le shell `/bin/bash`

commentaires :

date, nom de l'auteur, objet du script au minimum