

TP17 - Introduction aux scripts shell : autoformation

> Création du script « bonjourbis.sh » :

```
#!/bin/bash
echo "Bonjour $USER"
echo -n "Nous sommes le " ; date
echo "Votre numéro d'utilisateur est" `grep "^$USER" /etc/passwd | cut -d: -f3`
~
```

> Accord du droit d'exécution et exécution du script bonjourbis.sh :

```
root@DS1: ~#chmod a+x bonjourbis.sh
root@DS1: ~#./bonjourbis.sh
Bonjour root
Nous sommes le jeu. 30 janv. 2025 14:39:48 CET
Votre numéro d'utilisateur est 0
root@DS1: ~#_
```

> Création du script bonjourter.sh :

```
#!/bin/bash
if [ $# = 2 ]
then
echo "Bonjour $2 $1 et bonne journée !"
else
echo "Syntaxe : $0 nom prenom"
fi
~
```

> Exécution du script bonjourter.sh :

```
root@DS1: ~#sh bonjourter.sh
Syntaxe : bonjourter.sh nom prenom
root@DS1: ~#sh bonjourter.sh Faudon Enzo
Bonjour Enzo Faudon et bonne journée !
root@DS1: ~#
```

> Tests avec la commande echo :

```
root@DS1: ~#echo "Bonjour à tous !"
Bonjour à tous !
root@DS1: ~#echo "Bonjour \n à tous !"
Bonjour \n à tous !
root@DS1: ~#echo -e "Bonjour \nà tous !"
Bonjour
à tous !
root@DS1: ~#echo -e "Bonjour \nà toutes \net à tous ! \c"
Bonjour
à toutes
et à tous ! root@DS1: ~#_
```

> Création du script saisie_clavier.sh :

```
#!/bin/bash
echo "Donnez votre prénom et votre nom : "
read prenom nom
echo "Bonjour $prenom $nom"
```

> Exécution saisie_clavier.sh :

```
root@DS1: ~#sh saisie_clavier.sh
Donnez votre prénom et votre nom :
Enzo Faudon
Bonjour Enzo Faudon
root@DS1: ~#_
```

> Exemple de substitution de variable :

```
root@DS1: ~#n=123
root@DS1: ~#echo "La variable \$n vaut $n"
La variable $n vaut 123
root@DS1: ~#salut="bonjour à toutes et à tous!"
root@DS1: ~#echo "Alors moi je dis : $salut"
Alors moi je dis : bonjour à toutes et à tous!
root@DS1: ~#echo "Alors moi je dis : $salut"
Alors moi je dis : bonjour à toutes et à tous!
root@DS1: ~#echo "Alors moi je dis : \"\$salut\""
Alors moi je dis : "bonjour à toutes et à tous!"
root@DS1: ~#readonly salut
root@DS1: ~#salut="bonjour à tous sauf à toto"
-bash: salut : variable en lecture seule
root@DS1: ~#echo "Alors moi je dis : $salut"
Alors moi je dis : bonjour à toutes et à tous!
root@DS1: ~#
```

> Exemple de l'opérateur {} :

```
root@DS1: ~#user="/home/stagiaire"
root@DS1: ~#echo $user
/home/stagiaire
root@DS1: ~#u1=$user1
root@DS1: ~#echo $u1

root@DS1: ~#u1=${user}1
root@DS1: ~#echo $u1
/home/stagiaire1
root@DS1: ~#_
```

> Affichage du résultat de la commande « env | more » :

```
HELL=/bin/bash
DG_SEAT=seat0
WD=/root
OGNAME=root
DG_SESSION_TYPE=tty
SYSTEMD_EXEC_PID=554
OTD_SHOWN=pam
OME=/root
LANG=fr_FR.UTF-8
NVOCATION_ID=4227b32e76b44cc59377760bbdff232e
DG_SESSION_CLASS=user
ERM=linux
SER=root
HLVL=1
DG_VTNR=1
DG_SESSION_ID=1
DG_RUNTIME_DIR=/run/user/0
USHLOGIN=FALSE
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/root
BUS_SESSION_BUS_ADDRESS=unix:path=/run/user/0/bus
MAIL=/var/mail/root
=/usr/bin/env
```

> Utilisation des commandes de test :

```
root@DS1: ~# [ -e ./fichier ]
root@DS1: ~# echo $?
1
root@DS1: ~# touch fichier
root@DS1: ~# [ -e ./fichier ]
root@DS1: ~# echo $?
0
root@DS1: ~# [ -S ./fichier ]
root@DS1: ~# echo $?
1
root@DS1: ~# date > fichier
root@DS1: ~# [ -s ./fichier ]
root@DS1: ~# echo $?
0
root@DS1: ~# [ -r "/etc/passwd" ]
root@DS1: ~# echo $?
0
root@DS1: ~# [ -r "/etc/shadow" ]
root@DS1: ~# echo $?
0
```

> Test des commandes sur la session « enzo » :

```
root@DS1: ~# su - enzo
enzo@DS1:~$ [ -r "/etc/shadow" ]
enzo@DS1:~$ echo $?
1
enzo@DS1:~$ [ -r "/etc/shadow" ] || echo "lecture du fichier interdite"
lecture du fichier interdite
enzo@DS1:~$ [ -r "/etc/passwd" ]
enzo@DS1:~$ echo $?
0
```

> Test des chaînes de caractère :

```

root@DS1: ~# [ -z "est-ce que la chaîne est vide ?" ]
root@DS1: ~#echo $?
1
root@DS1: ~#ch="Bonjour"
root@DS1: ~#[ "$ch" = "bonjour" ]
root@DS1: ~#echo $?
1
root@DS1: ~#[ $USER != "root" ] && echo "l'utilisateur n'est pas le \"root\" !"
root@DS1: ~#[ $USER != "root" ] && echo "l'utilisateur n'est pas le \"root\" !" || echo "l'utilisateur est le \"root\" !"
l'utilisateur est le "root" !
root@DS1: ~#su - enzo
enzo@DS1:~$ [ $USER != "root" ] && echo "l'utilisateur n'est pas le \"root\" !" || echo "l'utilisateur est le \"root\" !"
l'utilisateur n'est pas le "root" !
enzo@DS1:~$

```

> Test d'un nombre :

```

root@DS1: ~#a=15
root@DS1: ~#[ $a -lt 15 ]
root@DS1: ~#echo $?
1
root@DS1: ~#[ $a -le 15 ]
root@DS1: ~#echo $?
0
root@DS1: ~#

```

> Test des opérations :

```

root@DS1: ~#f="/root"
root@DS1: ~#[ -d $f -a -x $f ]
root@DS1: ~#echo $?
0
root@DS1: ~#ls -ld /root
drwx----- 5 root root 4096 30 janv. 15:19 /root
root@DS1: ~#_

```

```

root@DS1: ~#note=9
root@DS1: ~#[ $note -lt 8 -o $note -ge 10 ] && echo "tu n'est pas convoqué à l'oral"
root@DS1: ~#note=7
root@DS1: ~#[ $note -lt 8 -o $note -ge 10 ] && echo "tu n'est pas convoqué à l'oral"
tu n'est pas convoqué à l'oral
root@DS1: ~#

```

> Création du script compte.sh pour un exemple de structure conditionnelle simple :

```

#!/bin/bash
if grep "^enzo" /etc/passwd
then
echo "enzo_a déjà un compte"
fi
~

```

> Exécution du script compte.sh (La machine virtuelle appartenait à Nicolas, d'où le user « nicolas ») :

```

root@DS1: ~#sh compte.sh
enzo:x:1000:1000:nicolas,,,:/home/nicolas:/bin/bash
enzo a déjà un compte
root@DS1: ~#_

```

> Modification du script compte.sh :

```
#!/bin/bash
if grep "^enzo" /etc/passwd > /dev/null
then
echo "enzo a déjà un compte"
fi
~
```

> Seconde exécution du script compte.sh :

```
root@DS1: ~#sh compte.sh
enzo a déjà un compte
root@DS1: ~#
```

> Création du script note.sh (avec une modification supplémentaire) :

```
#!/bin/bash
echo "Saisissez votre note : "
read note
if [ $note -gt 16 ]
then echo "C'est très bien !"
fi
~
```

```
#!/bin/bash
echo "Saisissez votre note : "
read note
if [ $note -gt 16 ]
then echo "C'est très bien !"
else
echo "Peut toujours mieux faire!"
fi
~
```

> Exécution du script avec modification :

```
root@DS1: ~#sh note.sh
Saisissez votre note :
5
Peut toujours mieux faire!
root@DS1: ~#sh note.sh
Saisissez votre note :
17
C'est très bien !
root@DS1: ~#
```

> Création du script examen.sh pour un exemple de structure conditionnelle imbriquée :

```
#!/bin/bash
echo "Saisissez votre moyenne : "
read note
if [ $note -lt 8 ]
then
echo "Récalé!"
elif [ $note -lt 10 ]
then
echo " Vous êtes convoqué à l'oral de rattrapage"
else
echo "Bien joué, vous êtes admis"
fi
~
```

> Exécution du script examen.sh :

```
root@DS1: ~#sh examen.sh
Saisissez votre moyenne :
7
Récalé!
root@DS1: ~#sh examen.sh
Saisissez votre moyenne :
8
 Vous êtes convoqué à l'oral de rattrapage
root@DS1: ~#sh examen.sh
Saisissez votre moyenne :
10
Bien joué, vous êtes admis
root@DS1: ~#_
```

> Création du script devoir.sh :

```
fichier=/home/enzo/devoir1.txt
if [ -f $fichier -a -r $fichier ]
then
echo "je vais vérifier ton devoir..."
elif [ ! -e $fichier ]
then
echo "ton devoir n'existe pas!"
else
echo "je ne peux pas lire ton devoir !"
fi
~
```

> Première exécution du script `devoir.sh` :

```
root@DS1: ~#sh devoir.sh
ton devoir n'existe pas!
root@DS1: ~#su - enzo
enzo@DS1:~$ touch devoir1.txt
enzo@DS1:~$ exit_
```

> Seconde exécution après avoir créé le script « `devoir1.txt` » :

```
root@DS1: ~#sh devoir.sh
je vais vérifier ton devoir...
root@DS1: ~#_
```

> Création du script `argument.sh` :

```
#!/bin/bash
if [ $# = 0 ]
then
echo "Erreur, la commande exige deux arguments"
elif [ $# = 1 ]
then
echo "Donnez le second argument : "
read arg2
fi_
~
```

> Exécution du script `argument.sh` :

```
root@DS1: ~#sh argument.sh
Erreur, la commande exige deux arguments
root@DS1: ~#sh argument.sh arg1
Donnez le second argument :
arg2
root@DS1: ~#
```

> Création du script `cas.sh` pour un exemple de script à choix multiples :

```
#!/bin/bash
case $USER in
root)
echo "Mes respects Monsieur le_$USER" ;;
guest | enzo)
echo "Salut $USER" ;;
*)
echo "Bonjour $USER" ;;
esac
~
```

> Exécution du script cas.sh :

```
root@DS1: ~#cas.sh
Mes respects Monsieur le root
root@DS1: ~#su - enzo
enzo@DS1:~$ cas.sh
Salut enzo
enzo@DS1:~$ _
```

> Création du script poursuite.sh :

```
#!/bin/bash
echo "Voulez-vous vraiment exécuter le script ? (o/n) // (y/n)"
read reponse
case $reponse in
[nN]*)
    echo "Script interrompu"
    exit 0
    ;;
[yYo0]*)
    echo "Attention pour le décompte final"
    for i in 10 9 8 7 6 5 4 3 2 1
    do
        do
            echo "==>$i"
            sleep 1
        done
        echo "Allumage Vulcain"
        sleep 2
        echo "Allumage des 2 EAP"
        sleep 2
        echo "Décollage"
        sleep 2
        echo "Tous les paramètres à bord sont normaux"
        sleep 2
        echo "Enzo est en orbite"
    done
    ;;
esac
~
```

> Exécution du script poursuite.sh :

```
root@DS1: ~#sh poursuite.sh
Voulez-vous vraiment exécuter le script ? (o/n) // (y/n)
y
Attention pour le décompte final
===>10
===>9
===>8
===>7
===>6
===>5
===>4
===>3
===>2
===>1
Allumage Vulcain
Allumage des 2 EAP
Décollage
Tous les paramètres à bord sont normaux
Enzo est en orbite
root@DS1: ~#_
```

> Création du script liste.sh pour un exemple de structure itérative :

```
#!/bin/bash
for nom in Nicolas Axel Noah Yanis Nathan Nina Yanis Maxence_Enzo
do
echo "$nom, bonjour"
done
~
```

> Exécution du script liste.sh :

```
root@DS1: ~#sh liste.sh
Nicolas, bonjour
Axel, bonjour
Noah, bonjour
Yanis, bonjour
Nathan, bonjour
Nina, bonjour
Yanis, bonjour
Maxence, bonjour
Enzo, bonjour
root@DS1: ~#_
```

> Création du script copie.sh :

```
#!/bin/bash
if [ ! -e /tmp/nicolas ]
then
mkdir /tmp/nicolas
fi
for fich in /home/nicolas/*
do
cp $fich /tmp/nicolas
done
~
```

> Exécution du script copie.sh :

```
root@DS1: ~#sh copie.sh
root@DS1: ~#ls -l /tmp/nicolas
total 0
-rw-r--r-- 1 root root 0 30 janv. 16:33 devoir1.txt
-rw-r--r-- 1 root root 0 30 janv. 16:33 fichier1
-rw-r--r-- 1 root root 0 30 janv. 16:33 fichier2
root@DS1: ~#
```

> Création du script sanslisteprecise.sh :

```
#!/bin/bash
cd /tmp/nicolas
set *
for nom in $@
do echo $nom
done_
~
```

> Exécution du script sanslisteprecise.sh :

```
root@DS1: ~#sh sanslisteprecise.sh
devoir1.txt
fichier1
fichier2
root@DS1: ~#_
```

> QUESTION : Comparez les deux scripts ci-dessous. Qu'en déduisez-vous ?

> J'en déduis que le deuxième script est le mieux car les commandes sont toutes faites à l'intérieur de la fonction while ce qui permet de recommencer la saisie en boucle si le nom du fichier est mal écrit.

> Création du script true.sh :

```
#!/bin/bash
while true
do
echo "Bonjour Monsieur $USER"
sleep 1
done
~
```

> Exécution du script true.sh :

```
root@DS1: ~#sh true.sh
Bonjour Monsieur root
Bonjour Monsieur root
Bonjour Monsieur root
Bonjour Monsieur root
Bonjour Monsieur root
Bonjour Monsieur root
^C
root@DS1: ~#_
```

> Modification du script true.sh :

```
#!/bin/bash
fichier=/etc/passwd_
while read ligne
do
echo $ligne
done < $fichier
```

> Exécution du nouveau script true.sh :

```
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:997:997:systemd Time Synchronization:/:/usr/sbin/nologin
messagebus:x:100:107::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:101:109:Avahi autoip daemon,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
vboxadd:x:999:1::/var/run/vboxadd:/bin/false
sshd:x:102:65534::/run/sshd:/usr/sbin/nologin
guest:x:1001:1001:guest,,:/home/guest:/bin/bash
luke:x:1002:1002::/home/luke:/bin/bash
vador:x:1003:1002::/home/vador:/bin/sh
solo:x:1004:1003::/home/solo:/bin/sh
leia:x:1005:1005::/home/leia:/bin/bash
toor:x:0:1006::/root:/bin/bash
palpatine:x:1007:1007:,,:/home/palpatine:/bin/bash
tcpdump:x:103:111::/nonexistent:/usr/sbin/nologin
bind:x:104:112::/var/cache/bind:/usr/sbin/nologin
enzo:x:1000:1000:nicolas,,:/home/nicolas:/bin/bash
```

> Création du script sortie.sh :

```
#!/bin/bash
fich="/etc/passwd"
grep "^enzo" $fich | while true
do
    read ligne
    if [ "$ligne" = "" ]
    then
        break
    fi
    echo $ligne
done
~
```

> Exécution du script sortie.sh :

```
root@DS1: ~#sh sortie.sh
enzo:x:1000:1000:nicolas,,:/home/nicolas:/bin/bash
root@DS1: ~#
```

> Commande diverse de remplacement de caractère :

```
root@DS1: ~#chaine="Bonjour, comment allez-VOUS aujourd'hui ?"
root@DS1: ~#echo $chaine | tr 'A-Z' 'a-z'
bonjour, comment allez-vous aujourd'hui ?
root@DS1: ~#
```

> Commande diverse qui remplace les deux points par des espaces :

```
root@DS1: ~#cat /etc/passwd | tr ":" " " > passwd.txt
root@DS1: ~#head passwd.txt
root x 0 0 root /root /bin/bash
daemon x 1 1 daemon /usr/sbin /usr/sbin/nologin
bin x 2 2 bin /bin /usr/sbin/nologin
sys x 3 3 sys /dev /usr/sbin/nologin
sync x 4 65534 sync /bin /bin/sync
games x 5 60 games /usr/games /usr/sbin/nologin
man x 6 12 man /var/cache/man /usr/sbin/nologin
lp x 7 7 lp /var/spool/lpd /usr/sbin/nologin
mail x 8 8 mail /var/mail /usr/sbin/nologin
news x 9 9 news /var/spool/news /usr/sbin/nologin
root@DS1: ~#
```

> Création du script read_set_tr.sh :

```
#!/bin/bash
fichier="/etc/passwd"
cat $fichier | head | tr ":" " " | while true
do
    read ligne
    if [ "$ligne" = "" ]
    then
        break
    fi
    set $ligne
    echo $1
done
~
```

> Exécution du fichier read_set_tr.sh :

```
root@DS1: ~#sh read_set_tr.sh
root
daemon
bin
sys
sync
games
man
lp
mail
news
root@DS1: ~#
```