A Corrigés des exercices

A.1 Les commandes du SGF

```
1) date
2) who
3) man ls
4) pwd
5) ls
6) cat > exemple
   ceci est un
   exemple de texte
   pour répondre a l'exercice 1
   question numéro 6
  ^D
7) cp exemple exemple.txt
8) mkdir TP_UNIX ; mv exemple.txt TP_UNIX
  ou
   mkdir TP_UNIX && mv exemple.txt TP_UNIX
9) ls -F
10) ls -l TP_UNIX
11) cd TP_UNIX
12) pwd ; ls
13) cat exemple.txt
14) cd
15) cat > fic1.ex
    ceci est le texte
    du premier fichier exemple
    appele fic1.ex
   ^D

    cat > fic2.ex
    ceci est le texte du second
    fichier exemple de la question 15
   ^D

16) ls *ex*
17) ls *.ex
18) rm exemple
19) mkdir EXEMPLE
20) mv fic1.ex EXEMPLE
21) mv fic2.ex EXEMPLE/fichier.ex
22) cd EXEMPLE ; ls
23) ls ../TP_UNIX
24) cd / ; ls
25) cd
    ou cd ~
```

A.2 Les commandes externes du Shell

```
Redirection
-----------
1) cat > fictrav
   premiere ligne
  ^D

2) cat >> fictrav
```

```
        deuxieme ligne
        troisieme ligne
       ^D

3) echo quatrieme ligne >> fictrav

4) ls -l > contenu_rep
5) mail adresse1 < contenu_rep

Droits d'accès
--------------
1) chmod g+r fictrav
2) cat ~nom_user/fictrav        -> problème d'accès au répertoire
                                           et d'accès en écriture
3) chmod go+rwx fictrav
4) chmod o-rwx *

Traitement sur les fichiers
---------------------------
1) wc -l contenu_rep
2) cat > fic1.ex
    ceci est le texte
    du premier fichier exemple
    appele fic1.ex
   ^D

    cat > fic2.ex
    ceci est le texte du second
    fichier exemple de la question 15
   ^D

3) head -3 fic1 > temp ; tail -5 fic2 >> temp
4) cat fic1 fic2 > fic3

Recherche d'un motif
--------------------
1) cd TP_UNIX
   puis grep "\<fichtrav\>" ../*
(c'est le fichier contenu_rep qui contient la chaîne fictrav)

2) grep -i "\<fictrav\>" ../*
   grep -n "\<fictrav\>" ../*

3) cd
   echo > ~/coordonnees
   echo DUPOND Jean 61 55 66 11 >> ~/coordonnees

4) cat >> ~/coordonnees
   ...
   ^D

5) grep "DUPOND" ~/coordonnees

Copie, déplacement et suppression de fichiers
---------------------------------------------
1) mkdir ~/REP1 ; mkdir ~/REP2 ; mkdir ~/REP3 ; cd ~/REP1
```

```
2) cp ~/coordonnees coord_bis
3) mv coord_bis ../REP2
   mv coord_bis ~/REP2
4) mv ../coordonnees ../REP3/coord_ter
   mv ~/coordonnees ~/REP3/coor_ter
5) rm ~/REP2/*


Affichage trié du contenu d'un fichier
--------------------------------------
1) cat > etudiant
   n1:p1:age1
   ...
   ^D

2) sort etudiant
3) sort -t: +2n etudiant -o etud_age
   ou
   sort -t: +2n etudiant > etud_age
4) asedit etud_age ...
5) cut -c 1-2, 7-10 etudiant  --> le nom est sur 2 caractères (numéro 1 et 2)
                                  le prénom sur 2 caractères (numéro 4 et 5)
                                  l'âge sur 4 caractères (numéros 7 à 10)
6) cut -c 1-5 < etudiant
7) cut -c 1-5 etudiant > liste_etud

Processus concurrents = communication de processus
--------------------------------------------------
1) ls -l | mail adresse1
2) ls | wc -w  ou ls -a | wc -w
3) ls -l| grep "^[^d]"
4) ls -l| grep "^[^d]..x" > info_exec
5) cut -c 7-10 < etudiant | sort
6) cut -c 7-10 < etudiant | sort > liste_etud2
7) sort -t: +2n -r etudiant | head -3

Contrôle de tâches
------------------
1) gedit
2) jobs (on n'a pas la main)
3) on récupère la main la commande jobs s'exécute
4) gedit & puis  jobs
(la commande s'exécute car on a récupéré la main)

5) et 6)
   ps avec différentes options
   jobs -l

7) mkdir ~/TP_C   puis   cd ~/TP_C    puis    gcc boucle.c
8) ls   puis  a.out &
9) ps
10) kill -9 num_process
11) a.out
12) ^Z
13) jobs -l  ---> le processus est marqué suspendu
14) fg %num_tache
```

```
    ^Z
    jobs -l  ---> le processus est marqué suspendu
15) bg %num_tache
    jobs -l  ---> le processus est marqué running
16) kill -9 %num_tache
    jobs -l  ---> le processus n'apparaît plus
```

A.3 Scripts shell

```
Expressions arithmétiques et scripts shell
------------------------------------------
1) directement saisies au clavier
set X = 5
set Y = 3
@ Z1 =  $X + $Y
@ Z2 = $X * $Y
echo $Z1
echo $Z2

2)Saisir le texte suivant sous un éditeur et nommer le fichier script1.csh

#!/bin/csh
# script1.csh
set X = 5
set Y = 3
@ Z1 =  $X + $Y
@ Z2 = $X * $Y
echo $Z1
echo $Z2

puis exécuter la commande
   csh script1.csh
  ou script1.csh si les droits en exécution ont été mis sur ce fichier

3) Version 1

#!/bin/csh
# script2.csh
echo "Donner la valeur de X"
set X = $<
echo "Donner la valeur de Y"
set Y = $<
@ Z1 =  $X + $Y
@ Z2 = $X * $Y
echo $Z1
echo $Z2

puis exécuter la commande
   csh script2.csh
  ou script2.csh si les droits en exécution ont été mis sur ce fichier
    saisir 2 valeurs a la demande du programme

Version 2

#!/bin/csh
# script3.csh
```

```
set X = $1               # valeur du premier paramètre
set Y = $2               # valeur du second paramètre
@ Z1 =  $X + $Y          # on pourrait faire directement $1 + $2
@ Z2 = $X * $Y           # on pourrait faire directement $1 * $2
echo $Z1
echo $Z2
```

ou bien

```
#!/bin/csh
# script3.csh
set X = $argv[1]         # valeur du premier paramètre
set Y = $argv[2]         # valeur du second paramètre
@ Z1 = $X + $Y           # on pourrait faire directement $argv[1] + $argv[2]
@ Z2 = $X * $Y           # on pourrait faire directement $argv[1] * $argv[2]
echo $Z1
echo $Z2
```

puis exécuter la commande
   csh script3.csh valeur_de_X valeur_de_Y
  ou script3.csh valeur_de_X valeur_de_Y
si les droits en exécution ont été mis sur ce fichier (exemple script3.csh 3 5)

Conversion de la taille d'un fichier

Version 1

```
#!/bin/csh
# conv_taille

if ( !(-e $1) ) then            # test sur l'existence du fichier
  echo "Le fichier $1 n'existe pas"
else if (-z $1) then            # test sur le contenu du fichier
  echo "$1 est vide"
else if (-f $1) then            # test sur la nature du fichier
                                # ---> il s'agit d'un fichier non vide
                                #      et qui n'est pas un repertoire
  set liste = `ls -l $1`
  set taille = $liste[4]        # récupération de la taille du fichier

  echo "----- Taille du fichier $1 ($taille)"
                                # conversion de la taille
  @ g = $taille / (1024 * 1024 * 1024)
  @ taille %= (1024 * 1024 * 1024)
  @ m = $taille / (1024 * 1024)
  @ taille %= (1024 * 1024)
  @ k = $taille / 1024
  @ o = $taille % 1024
                                # affichage des différents elements
  if ($g > 0) then
              echo "soit : $g Gio, $m Mio, $k Kio et $o octets"
  else  if ($m > 0) then
                    echo "soit : $m Mio, $k Kio et $o octets"
        else if ($k > 0) then
```

```
                        echo "soit : $Kio et $o octets"
             else
             echo "$o octets"
  endif
else    #---> il s'agit d'un répertoire

echo "$1 est un repertoire"
endif


# 1234567890 => 1 Gio, 153 Mio, 384 Kio et 722 octets
# 1234567 => 1 Mio, 181 Kio et 647 octets
# 1234 => 1 Kio et 210 octets
# 123 =>  123 octets


Version 2

#!/bin/csh
# conv_taille_bis

if ( !(-e $1) ) then            #test sur l'existence du fichier
  echo "Le fichier $1 n'existe pas"
else if (-z $1) then            #test sur le contenu du fichier
  echo "$1 est vide"
else if (-f $1) then            # test sur la nature du fichier
                                # ---> il s'agit d'un fichier non vide
                                #      et qui n'est pas un repertoire
        set liste = `ls -l $1`
        set taille = $liste[4]  # récupération de la taille du fichier
        echo "----- Taille du fichier $1 ($taille)"
        @ g = $taille / (1024 * 1024 * 1024)  # conversion
        @ taille %= (1024 * 1024 * 1024)
        @ m = $taille / (1024 * 1024)
        @ taille %= (1024 * 1024)
        @ k = $taille / 1024
        @ o = $taille % 1024
        if ($g > 0) then        #affichage du résultat
                 echo "soit : $g Gio, $m Mio, $k Kio et $o octets"
        else if ($m > 0) then
                        echo "soit : $m Mio, $k Kio et $o octets"
             else if ($k > 0) then
                        echo "soit : $k Kio et $o octets"
                  else
                        echo "$o octets"
        endif
     else                 # ---> il s'agit d'un repertoire
                          # calcul de la taille réelle d'un repertoire
        set taille = 0
        set liste_fic = `ls $1` # récupération de la liste des fichiers
        echo "$liste_fic"
        foreach fic ($liste_fic)        # cumul de la taille de chaque fichier
                set info = `ls -l $1/$fic` # récupération de la taille
                set taille_fic = $info[4]
                @ taille += $taille_fic
        end
```

```
        echo "Taille du repertoire $taille"
     endif


# 1234567890 => 1 Gio, 153 Mio, 384 Kio et 722 octets
# 1234567 => 1 Mio, 181 Kio et 647 octets
# 1234 => 1 Kio et 210 octets
# 123 =>  123 octets
```