

Note de cours

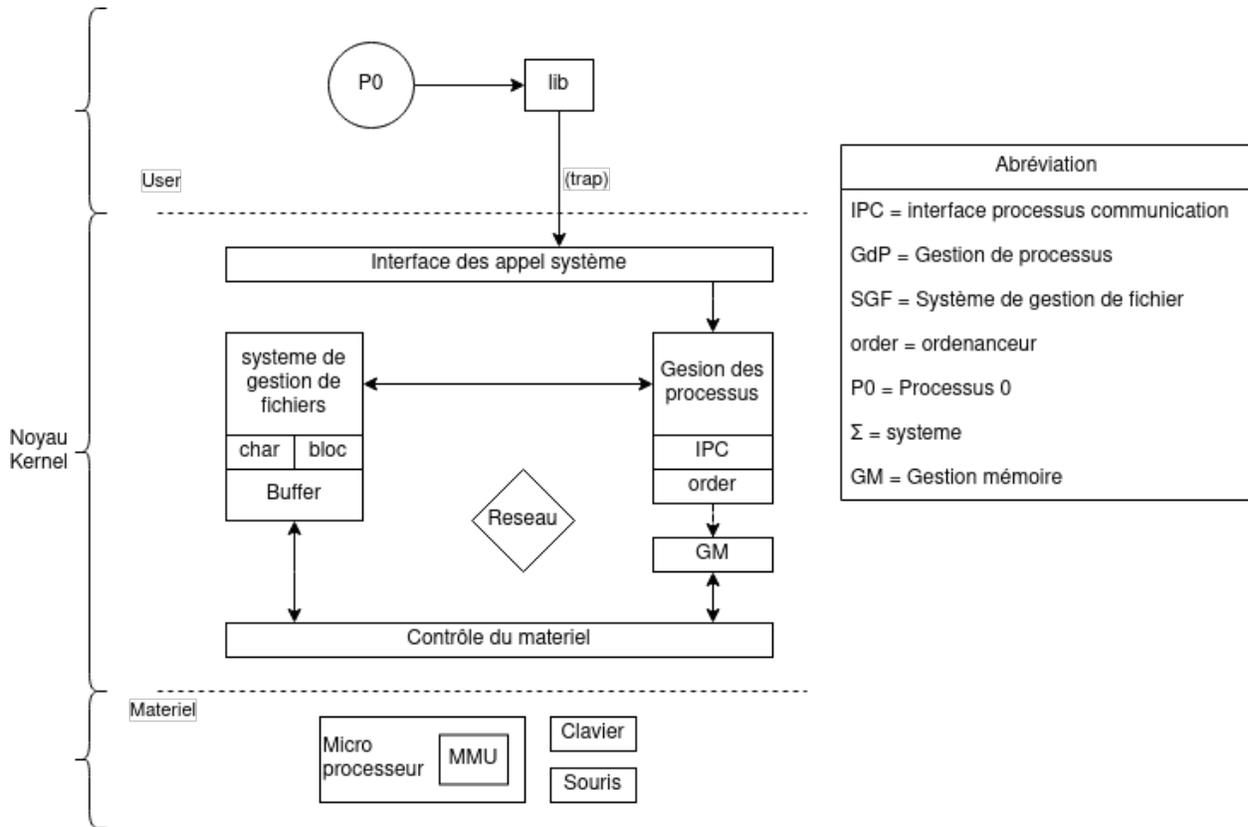


Figure 1: ../diagram_system.png

Partiel système - Février 2023

Document papiers et électroniques non autorisés - Durée 1h45

Debut 13h13

Exercice 1

Ajouter un cas (une instruction) au programme ci-dessous pour afficher l'égalisé des deux nombres que l'on doit saisir à partir de l'entrée standard.

```
#!/bin/bash
```

```
echo "Entrez un premier entier positif x = "  
read -r x  
if [ "$x" -le 1 ]; then  
    echo "x doit etre positif"  
    exit 1  
fi  
  
echo "Entrez un deuxieme entier positif y = "  
read -r y  
if [ "$y" -le 1 ]; then
```

```

    echo "y doit etre positif"
    exit 1
fi

# === Modification, -ge (greter or equal >=) vers -gt (greter than >)
if [ "$x" -gt "$y" ]
    echo "x est plus grand que y"
elif [ "$y" -gt "$x" ]
    echo "y est plus grand que x"
# ==== Ajout de l'instruction egale ==== #
elif [ "$x" -eq "$y" ]
    echo "x et y sont egale"
# ==== fin de l'ajout ==== #
fi

```

Exercice 2

Ecrire une fonction qui affiche le nombre d'argument qu'elle à reçu (lors de son appel)

```

#!/bin/bash

nb_argument(){
    echo $#
}

nb_argument 1 2 3

```

Exercice 3

Ecrire une fonction qui affiche le nombre d'argument et aussi les argument quelle à reçu

```

#!/bin/bash

echoArgs(){
    echo $#
    echo $@
}

echoArgs "Hello" "World" "!"

```

Exercice 4

Ecrire une fonction qui calcule la somme de ses deux premiers argument (type entier) et renvoie le resultat dans son troisieme

```

somme()
{
    result=$(( $1 + $2 ))
}

somme 5 5 $result
echo $result

```

Exercice 5

Ecrire une fonction qui reçoit un tableau d'un dizaine d'entiers et retourne comme résultat le plus petit élément de ce tableau. Les éléments du tableau sont initialisés à partir d'un clavier et dans la fonction appelante

```
plusPetitElement()
{
  arr=("$@")
  max=0
  for i in "${arr[@]}; do
    if [ $i -gt $max ]; then
      max=$i
    fi
  done

  echo $max
}

tableau=(37 100 42 69 5)
plusPetitElement "${tableau[@]}"
```

Exercice 6

- 1) Copier le fichier passwd de son repertoire d'origine (systeme) dans votre repertoire de travail sous le nom passwd.tmp

```
cp /etc/passwd ~/passwd.tmp
```

- 2) Extraire en une seule commande, du fichier passwd.tmp les noms de login et UID, en les triant en fonction des UID et en redirigeant le tout vers un nouveau fichier passwd.trie

```
cat passwd.tmp | awk -F':' '{print $1 "\t" $4}' | sort -n -k 2 > passwd.trie
```

Exercice 7

- 1) Extraire du fichier passwd.tmp les lignes contenant les champs login et repertoire de travail (home directory)

```
cat passwd.tmp | awk -F':' '{print $1 ":" $6}'
```

- 2) Mettre les résultats de l'opération précédente dans un fichier de nom passwd.res

```
cat passwd.tmp | awk -F':' '{print $1 ":" $6}' > passwd.res
```

- 3) Remplacer dans le fichier passwd.res les champs : par le caractère "!" et mettre le résultat dans le fichier passwd.fin

```
cat passwd.res | awk -F':' '{print $1 "!" $2}' > passwd.fin
```

- 4)

```
tail -n 15 passwd.fin
```

Exercice 8

- 1) Afficher par ordre décroissant, les 5 premiers fichiers du repertoire /etc ayant la plus grande taille.

```
ls -lS /etc | head -n 6
```

- 2) Trouver un fichier dans le répertoire /etc qui à été modifié il ya moins de 3 jours et dans la taille est inférieur à 1Ko.

```
find /etc -type f -mtime -3 -size -1k
```