

Devoir sur Table : Programmation Système

Enseignant : Anthony Busson

Exercice 1 :

Nous considérons le code ci-dessous. Donnez le résultat de l'affichage de ce programme et expliquez.

```
int main()
{
    int tab[2], nbOctets, retourFork;
    char string1[100], string2[100], buffer[100];

    strcpy(string1,"Chaine 1\n");
    strcpy(string2,"Chaine 2\n");

    if((retourFork=fork())<0) {
        perror("Erreur fork()");
        exit(1);
    }

    if(pipe(tab)<0) {
        perror("Erreur pipe"); exit(1);
    }

    if(write(tab[1],string1,strlen(string1))<0) perror("Erreur write");
    if(write(tab[1],string2,strlen(string2))<0) perror("Erreur write");

    if(retourFork>0) {
        close(tab[0]); close(tab[1]);
        exit(0);
    }
    while((nbOctets=read(tab[0],buffer,100))>0)
    {
        buffer[nbOctets-1]='\0';//Rajout du caractere de fin de chaine
        printf("%s\n",buffer);
    }

    close(tab[0]); close(tab[1]);
    return(0);
} //fin du main
```

Exercice 2 : Signaux

Nous considérons le code ci-dessous. Donnez le résultat de l'affichage de ce programme et expliquez.

```
void myHandler(int sig)
{
    printf("Signal SIGUSR1 reçu\n");
    kill(getpid(),SIGUSR1);
}

int main()
{
    signal(SIGUSR1,myHandler);
    kill(getpid(),SIGUSR1);

    return(0);
}
```

Exercice 3 :

Nous considérons le bout de code ci-dessous. Vous devrez décrire le contenu du tableau des fichiers ouverts (voir plus bas).

```
int main()
{
    int fd1, fd2, tab[2] ;
    if( (fd1=open(« fichier1.txt », O_WRONLY|O_CREAT,0666))<0)
        perror("Erreur open fichier1.txt");
    if( (fd2=open(« fichier2.txt », O_WRONLY|O_CREAT,0666))<0)
        perror("Erreur open fichier2.txt");

    if(pipe(tab)<0) perror("Erreur pipe");
    dup2(tab[1],fd2);
    close(0);
    dup(tab[0]);
    dup2(fd1, tab[1]);
}
```

Complétez le tableau ci-dessous de manière à indiquer vers quel fichier pointe chaque case du tableau après la dernière instruction (dup2(..)).

Pointeur[0]
Pointeur[1]
Pointeur[2]
Pointeur[3]
Pointeur[4]
Pointeur[5]
Pointeur[6]

Rappel du man de dup et dup2 :**Nom**

dup, dup2 - Dupliquer un descripteur de fichier.

Synopsis

```
#include <unistd.h>
int dup(int oldfd);int dup2(int oldfd, int newfd);
```

Description

dup et **dup2** créent une copie du descripteur de fichier *oldfd*.

Après un appel réussi à **dup** ou **dup2**, l'ancien et le nouveau descripteurs peuvent être utilisés de manière interchangeable. Ils partagent les verrous, les pointeurs de position et les drapeaux. Par exemple si le pointeur de position est modifié en utilisant **lseek** sur l'un des descripteurs, la position est également changée pour l'autre. Les deux descripteurs ne partagent toutefois pas le drapeau Close-on-exec.

dup utilise le plus petit numéro inutilisé pour le nouveau descripteur.

dup2 transforme *newfd* en une copie de *oldfd*, fermant auparavant *newfd* si besoin est.

Valeur Renvoyée

dup et **dup2** renvoient le nouveau descripteur, ou -1 s'ils échouent, auquel cas *errno* contient le code d'erreur.

Exercice 4 :

7 erreurs se sont introduites dans le code ci-dessous. Trouvez les et commentez.

```
int main()
{
    int fd1 ;
    char* buffer, string[100];
    string="Une chaine de caractere\n";
    if(fd1=open("fichier1.txt",O_WRONLY | O_CREAT)<0) {
        perror("Erreur open");
        exit(1);
    }
    if(read(1,&buffer,sizeof(buffer))!=NULL)
        if(write(fd1,buffer,sizeof(buffer))==NULL) perror("Erreur write");

    if(write(2,string,sizeof(string))<0) perror("Erreur write");
    close(fd1);
    return(0);
}
```

Question de cours :

A quoi sert l'appel système wait() ? Quels sont ces champs ? Donnez des exemples d'utilisation.

Pourquoi un exit(-1) est-il incorrect ?

Quel est l'effet du flag SA RESTART dans sigaction ?