 NOM : Prénom :

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. Vous pourrez donner les pid que vous souhaitez, il faut juste qu'ils soient cohérents vis-à-vis de l'exécution du programme.

```
int main()
{
	int i, retourExit, retourFork=1, pid;

	for(i=0; i < 4; i++)
	{
		if(retourFork>0)
		{
			printf("Mon PID est %d et i=%d\n",getpid(),i);
			retourFork=fork();
		}
	}
	if( (pid=wait(&retourExit))>0 ) printf("Code de retour du fils %d:
%d\n",pid,WEXITSTATUS(retourExit));
	exit(i);
}//fin du main
```

Exercice 2 :

Nous considérons le code suivant :

```
int fd ;
Char buffer[9] ;
strcpy(buffer, « tototiti ») ;
strcpy(buffer,« tata ») ;
```

On suppose que fichier1.txt existe et contient « 123456789101112 ». Quel sera le contenu du fichier après les ouvertures/écritures suivantes :

- if((fd=open(« fichier1.txt », O_WRONLY | O_TRUNC))>0) write(fd,buffer,strlen(buffer)) ;
- if((fd=open(« fichier1.txt », O_WRONLY | O_APPEND))>0) write(fd,buffer,strlen(buffer)) ;
- if((fd=open(« fichier1.txt », O_WRONLY))>0) write(fd,buffer,strlen(buffer)) ;
- if((fd=open(« fichier1.txt », O_WRONLY))>0) write(fd,buffer,sizeof(buffer)) ;

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 ;
    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”);
    dup(1);
    dup2(1,fd1);
}
```

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

 NOM : Prénom :

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

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.

Question de cours :

Pourquoi certaines commandes du shell sont internes ?

Détaillez le fonctionnement de l'appel système execvp ?