



جامعة سيدي محمد بن عبد الله
كلية العلوم ظهر المهرارز

جامعة سيدي محمد بن عبد الله
كلية العلوم ظهر المهرارز

Université Sidi Mohamed Ben Abdellah
Faculté des Sciences Dhar Mahraz



Compte Rendu De Travaux Pratique
System d'Exploitation
Travaille Pratique 6 : Les Tubes



SMI – S4

Réaliser par :

Prénom :	CNE :	Nom :
Achraf	1513755449	Tazi
Oussama	1311778906	Nadrani

Exercice1 :

(a)Afficher sur la sortie standard et sur un fichier texte 'resu_cop' le contenu de la commande ps avec les options aux.

Réponse :

```
nadrani@nadrani-pc:~/sys_tp/TpSys3$ ps -aux|tee resu_cop
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.3	4464	3400	?	Ss	04:49	0:07	/sbin/init
root	2	0.0	0.0	0	0	?	S	04:49	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	04:49	0:03	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	04:49	0:00	[kworker/0:0H]
root	7	0.0	0.0	0	0	?	S	04:49	0:11	[rcu_sched]
root	8	0.0	0.0	0	0	?	S	04:49	0:00	[rcu_bh]
root	9	0.0	0.0	0	0	?	S	04:49	0:00	[migration/0]
root	10	0.0	0.0	0	0	?	S	04:49	0:02	[watchdog/0]
root	11	0.0	0.0	0	0	?	S<	04:49	0:00	[khelper]
root	12	0.0	0.0	0	0	?	S	04:49	0:00	[kdevtmpfs]

(b) Concaténer le contenu texte de tous les fichiers du répertoire courant. Dans le résultat de cette première opération, on cherche les lignes qui contiennent la chaine 'linux'. Dans le résultat de cette 2ème opération on élimine les lignes qui contiennent la chaine 'unix'. Comme dernière opération, on compte le nombre de lignes obtenues dans la 3ème opération.

Réponse :

```
nadrani@nadrani-pc:~$ cat * 2>dev|grep "linux"|grep -v "unix"|wc -l
```

0

Exercice2 :

En utilisant les fonctions `sprintf` et `write`, écrire le message "Bonjour" dans un tube, depuis le père

Réponse :

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#define TAILLE_MESSAGE 256
int main(void)
{
    pid_t pid_fils;
    int descripteurTube[2];
    char messageEcrire[TAILLE_MESSAGE];
    pipe(descripteurTube);
    pid_fils = fork();
    if(pid_fils != 0)
    {
        sprintf(messageEcrire, "Bonjour!!");

        write(descripteurTube[1], messageEcrire, TAILLE_MESSAGE);
    }
    return EXIT_SUCCESS;
}
```

```
nadrani@nadrani-pc:~/sys_tp/TpSys6$ gcc -pthread travail1.c -o TP6_1
nadrani@nadrani-pc:~/sys_tp/TpSys6$ ./TP6_1
nadrani@nadrani-pc:~/sys_tp/TpSys6$
```

- *La compilation ne nous donne aucune erreur, cependant, elle n'affiche rien sur la sortie standard, car le père fait seulement l'écriture dans le pipe*

Exercice3 :

En utilisant les fonctions read et printf, écrire le programme qui permet de lire un message envoyé par le père à un fils.

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#define TAILLE_MESSAGE 256
int main(void)
{
    pid_t pid_fils;
    int descripteurTube[2];
    char messageLire[TAILLE_MESSAGE];
    pipe(descripteurTube);
    pid_fils = fork();
    if(pid_fils == 0)
    {
        read(descripteurTube[0], messageLire, TAILLE_MESSAGE);
        printf("Message reçu = \"%s\"", messageLire);
    }
    return 0;
}
```

```
nadrani@nadrani-pc:~/sys_tp/TpSys6$ gcc -pthread travail2.c -o TP6_2
nadrani@nadrani-pc:~/sys_tp/TpSys6$ ./TP6_2
^C
nadrani@nadrani-pc:~/sys_tp/TpSys6$
```

Exercice4 :

Écrivez un programme qui crée un tube et, dont le père crée un fils. Le programme permet la fermeture de l'entrée et de la sortie du tube, (en supposant qu'un message va être envoyé du père vers le fils).

```
exercice4_tp6.c x
#include<stdlib.h>
#include<stdio.h>
#include<unistd.h>
main(){
int t[2];
pid_t P;
if(pipe(t) !=0){
printf("erreur de création d'un tube");
exit(-1);}
P=fork();
if(P==0)
close(t[1]);
if(P !=0)
close(t[0]);
return 0;
}
```

Exercice5 :

En faisant attention à la fermeture de l'entrée et de la sortie du tube, aussi en se servant des fonctions : read, printf, sprint et write, sans oublier le test des différents cas d'erreurs, écrivez un programme qui crée deux processus : le père écrit le message « Bonjour, fils. Je suis ton père ! ». Le fils le récupère, puis l'affiche.

```

#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>
#include <sys/wait.h>
#define TAILLE_MESSAGE 256
int main(void)
{
    pid_t pid_fils;
    int descripteurTube[2];
    char messageLire[TAILLE_MESSAGE], messageEcrire[TAILLE_MESSAGE];
    pipe(descripteurTube);
    pid_fils=fork();
    if(pid_fils==-1)
    {
        fprintf(stderr, "ERRUR de creation du processus.\n");
        return 1;
    }

    if(pid_fils)
    {
        close(descripteurTube[1]);
        read(descripteurTube[0], messageLire, TAILLE_MESSAGE);
        printf("je suis le fils %d.\n j'ai reçu le message suivant du pere =\n%s\n", getpid(), messageLire);
    }
    else
    {
        close(descripteurTube[0]);
        sprintf(messageEcrire, "Bonjour fils je suis ton pere");
        printf("\nje suis le pere =%d j'envoie le message suivant\n%s\n" a mon fils\n", getpid(), messageEcrire);
        write(descripteurTube[1], messageEcrire, TAILLE_MESSAGE);
    }
    wait(NULL);
    return 0;
}

```

```

nadrani@nadrani-pc:~/sys_tp/TpSys6$ gcc -pthread travail4.c -o TP6_4
nadrani@nadrani-pc:~/sys_tp/TpSys6$ ./TP6_4

je suis le pere =5132 j'envoie le message suivant"Bonjour fils je suis ton pere"
a mon fils
je suis le fils 5131.
j'ai reçu le message suivant du pere ="Bonjour fils je suis ton pere"

```