



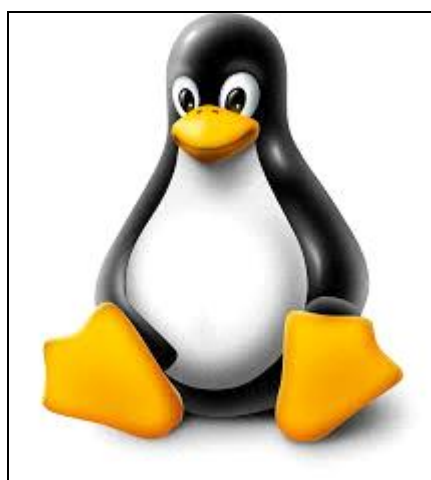
جامعة سيدي محمد بن عبد الله
كلية العلوم ظهر المهرارز

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 4 : Les Fichiers



SMI – S4

Réaliser par :

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

Travail 1 : *Après exécution du programme ci-dessous que remarquez-vous lorsque après la demande de l'entrée standard vous faites :*

<enter>

a<enter>

a<backspace>b<enter>

abcd<enter>

```
#include <stdio.h>
int main (void)
{
    int x = fgetc(stdin);
    printf ("x = %d ('%c')\n", x, x);
    return 0;
}
```

Résultat :

```
achrafkey@ubuntu:~/TpSys4$ gcc travaille1.c -o ex1
achrafkey@ubuntu:~/TpSys4$ ./ex1
a
x = 97 ('a')
achrafkey@ubuntu:~/TpSys4$ ./ex1
a b
x = 97 ('a')
achrafkey@ubuntu:~/TpSys4$ ./ex1
abcd
x = 97 ('a')
achrafkey@ubuntu:~/TpSys4$
```

Observation :

<enter>

Lorsqu'on clique sur la touche « entrer », on remarque que le programme affiche le code ASCII de \n qui égale à 10, et entre les parenthèses le \n prend sa valeur et fait un retour à la ligne.

a<enter>

La même chose se déroule pour l'affichage de caractère saisi et son code ASCII.

a<backspace>b<enter>

Ici le caractère extrait est a car « fgetc » lit le premier caractère saisi.

abcd<enter>

Lorsqu'on saisit une chaîne de caractère, le programme prend seulement le premier et l'affiche sur la sortie standard.

Travail 2 : *En utilisant la fonction `getc()` ou `getchar()`, écrire un programme C qui permet de lire un caractère à partir de l'entrée standard et de l'afficher sur la sortie standard.*

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>

int main(void){
    printf("saisir un caractere ");
    char x = fgetc(stdin);
    printf("x = %c\n", x);
    return 0;
}
```

```
achrafkey@ubuntu:~/TpSys3$ gcc travail2.c
achrafkey@ubuntu:~/TpSys3$ ./a.out
total 28
-rwxrwxr-x 1 achrafkey achrafkey 7345 Apr  3 07:21 a.out
-rwxrwxr-x 1 achrafkey achrafkey 7333 Apr  3 06:51 ex1
-rw-rw-r-- 1 achrafkey achrafkey  261 Apr  3 06:54 travail1.c
-rw-rw-r-- 1 achrafkey achrafkey  261 Apr  3 06:54 travail1.c~
-rw-rw-r-- 1 achrafkey achrafkey  146 Apr  3 07:06 travail2.c
-rw-rw-r-- 1 achrafkey achrafkey    0 Apr  3 07:06 travail2.c~
achrafkey@ubuntu:~/TpSys3$
```

Travail 4 : *Ecrire un programme C correspondant à la commande Shell :*
Cat fichier1>>fichier2.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main(int argc, char *argv[]){
    FILE *f1,*f2;
    char c;
    if(argc != 3){
        printf("nombre de paramètre incorrect \n");
        exit(1);
    }
}
```

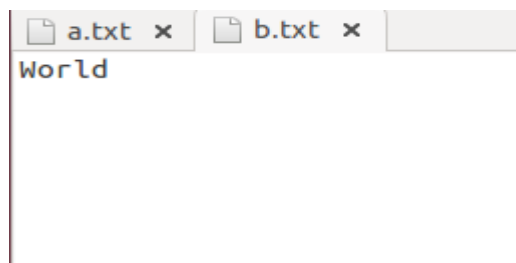
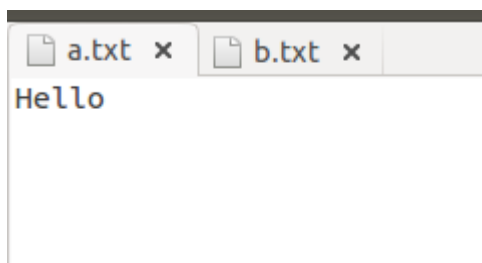
```

    }
    if((f1=fopen(argv[1],"r"))==NULL){
        printf("Ouverture impossible de %s\n", argv[1]);
        exit(2);
    }
    if((f2=fopen(argv[2],"a"))==NULL){
        printf("Ouverture impossible de %s\n", argv[2]);
        exit(3);
    }
    while(c=fgetc(f1) != EOF){
        putc(c,f2);
    }
    fclose(f1);
    fclose(f2);
    return 0;
}

```

Résultat :

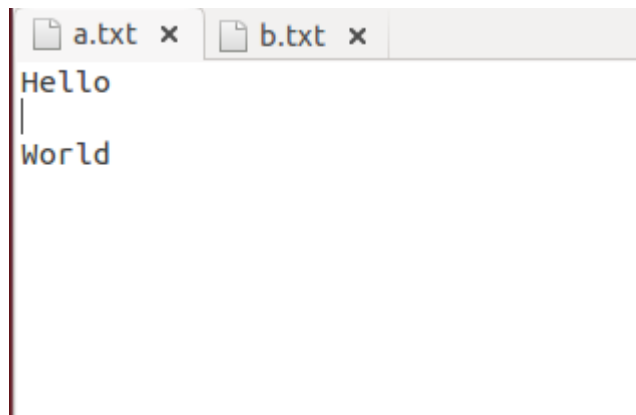
Avant l'appel de la fonction :



L'exécution :

```
achrafkey@ubuntu:~/TpSys4$ gcc travaille4.c -o ex1
achrafkey@ubuntu:~/TpSys4$ ./ex1 a.txt b.txt
achrafkey@ubuntu:~/TpSys4$
```

Après l'appel à la fonction



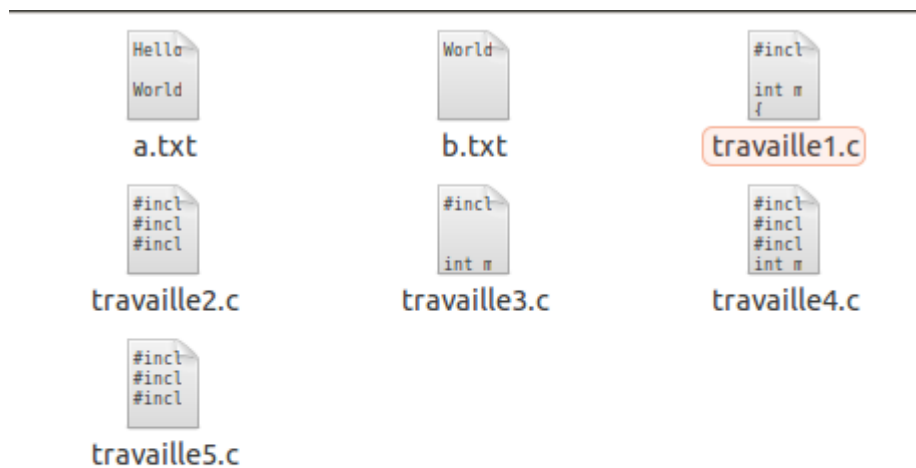
```
a.txt x b.txt x
Hello
World
```

Travail 5 : *Ecrire un programme appelé **ECRI_TEXT** qui permet de placer du texte dans le fichier **test.txt** en utilisant les fonctions **fprintf()** et **fputs()**. Vérifier que l'ouverture de **test.txt** n'a pas connue d'erreur.*

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main(int argc, char *argv[]){
    FILE *f1;
    char c;
    if((f1=fopen("test.txt","w"))==NULL){
```

```
printf("Ouverture impossible de fichier test");  
exit(1);  
}  
fputs("This is a test", f1);  
fclose(f1);  
return 0;  
}
```

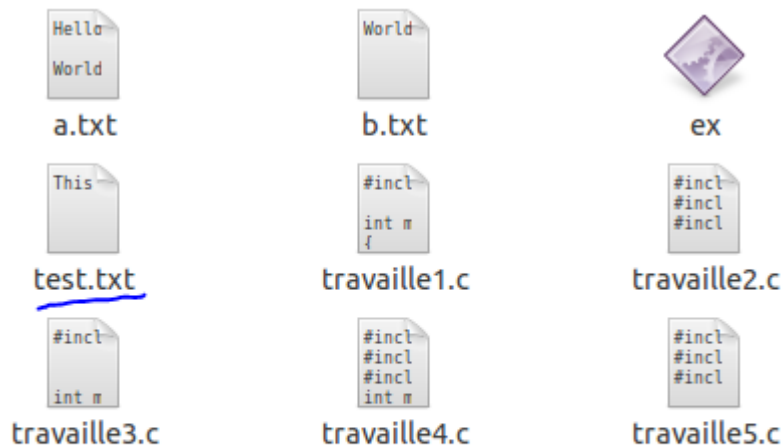
Avant l'exécution :



L'exécution:

```
chrafkey@ubuntu:~/TpSys4$ gcc travaille5.c -o ex  
chrafkey@ubuntu:~/TpSys4$ ./ex  
chrafkey@ubuntu:~/TpSys4$
```

Après l'exécution:



Travail 6 : (Suite de l'exercice 5). Affichez sur la sortie standard le contenu du fichier test.txt en utilisant les fonctions fscanf() et fgets(). Vérifier que l'ouverture de test.txt n'a pas connue d'erreur.

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h>
int main(){
    FILE *f1;
```



```
char str[60];  
f1 = fopen("test.txt", "r");  
if(f1 == NULL){  
    printf("Erreur d'ouvrir le fichier");  
    exit(1);  
}  
  
if(fgets(str, 60, f1)!=NULL){  
    puts(str);  
}  
fclose(f1);  
return 0;  
}
```

```
achrafkey@ubuntu:~/TpSys4$ gcc travaille6.c -o ex  
achrafkey@ubuntu:~/TpSys4$ ./ex  
This is a Test  
  
achrafkey@ubuntu:~/TpSys4$
```