

TP1 Dylan KAUFFMANN // Manon BRIAND

Exercise 1:

I.

A. 1. touch test1 test2.c testt44.pdf test455 testt41.c test510.c test_all.pdf testmaker

2. mkdir Exercice1 && mv ./test* ./Exercice1

3. a) ls -la *5

b) ls -la testt*

c) ls -la *.pdf

d) ls -la t????t*

B.

1. a) ls -la ./ | grep -v testmaker

b) ls -la ./ | grep -n *455*

c) ls -la ./ | grep ".*.c"

Exercise 2:

II.

AB.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
user console Sep 3 16:35
user ttys001 Sep 16 14:05
user ttys002 Sep 23 12:54
user ttys015 Sep 23 13:47
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
```

C.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
Ven 23 sep 2022 14:02:36 CEST
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
```

D. Le rôle de ">" permet de rediriger la sortie standard d'une commande vers un fichier qui est tout le temps crée.

E.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2
```

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master date > fich
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master
```

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master cat fich
Ven 23 sep 2022 14:02:36 CEST
user console Sep 3 16:35
user ttys001 Sep 16 14:05
user ttys002 Sep 23 12:54
user ttys015 Sep 23 13:47
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master
```

F. Le rôle de ">>" permet de rediriger la sortie standard d'une commande en concaténant sa sortie dans un fichier crée en ouvrant une nouvelle ligne

Exercice 3:

III.

A.

1.

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master cat
```

2.

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master cat
Bonjour
Bonjour
```

3.

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master cat
Bonjour
Bonjour
Pourquoi pas oui bien sur
Pourquoi pas oui bien sur
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master
```

4. La combinaison Ctrl+C envoie un signal pour terminer le processus alors que la commande Ctrl+D donne juste le caractère EOF (End Of File).

5. La commande cat < fich permet d'envoyer le contenu du fichier fich vers l'entrée standard de cat.

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice2 > master cat < fich
Ven 23 sep 2022 14:02:36 CEST
user console Sep 3 16:35
user ttys001 Sep 16 14:05
user ttys002 Sep 23 12:54
user ttys015 Sep 23 13:47
```

B.

1.

```
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2 > master wc fich
5      26      158 fich
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2 > master wc < fich
5      26      158
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice2 > master
```

Exercise 4:

IV.

A.

1. On peut constater que le fichier est vide car le message d'erreur a été envoyé sur la sortie d'erreur.

2.

```
log
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master dater 2> liog
X user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master cat liog
zsh: command not found: dater
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master
```

3.

```
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master ls /usra 2> liog
X user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master cat liog
ls: /usra: No such file or directory
user@AB-MBP-008 > ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice4 > master
```

B.

```
✗ user@AB-MBP-008 ~ ➤ ls -R /Users/ab
Desktop                               Movies
Documents                             Music
Downloads                             Pictures
FortiClientVPNSetup_7.0.2.0069_macosx.dmg Public
Library

/Users/ab/Desktop:
ls: /Users/ab/Desktop: Permission denied

/Users/ab/Documents:
ls: /Users/ab/Documents: Permission denied

/Users/ab/Downloads:
ls: /Users/ab/Downloads: Permission denied

/Users/ab/Library:
ls: /Users/ab/Library: Permission denied

/Users/ab/Movies:
ls: /Users/ab/Movies: Permission denied

/Users/ab/Music:
ls: /Users/ab/Music: Permission denied

/Users/ab/Pictures:
ls: /Users/ab/Pictures: Permission denied

/Users/ab/Public:
Drop Box

/Users/ab/Public/Drop Box:
ls: /Users/ab/Public/Drop Box: Permission denied
✗ user@AB-MBP-008 ~ ➤
```

```
✗ user@AB-MBP-008 ~ ➤ ls -R /Users/ab 2> /dev/null
Desktop                               Movies
Documents                             Music
Downloads                             Pictures
FortiClientVPNSetup_7.0.2.0069_macosx.dmg Public
Library

/Users/ab/Desktop:

/Users/ab/Documents:

/Users/ab/Downloads:

/Users/ab/Library:

/Users/ab/Movies:

/Users/ab/Music:

/Users/ab/Pictures:

/Users/ab/Public:
Drop Box

/Users/ab/Public/Drop Box:
✗ user@AB-MBP-008 ~ ➤ ls -R /Users/ab 2> /dev/null
```

Exercise 5:

```
creation_fichier.c

1  #include <stdio.h>
2  #include <sys/stat.h>
3  #include <sys/types.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7  int main(int argc, char *argv[]) {()
8      /* Chemin vers le nouveau fichier */
9      char* path = argv[1];
10     /* Permissions du nouveau fichier */
11     mode_t mode = S_IRUSR | S_IWUSR | S_IRGRP | S_IWGRP | S_IROTH; /* Crée le fichier */
12     int fd = open (path, O_CREAT|O_APPEND, mode);
13     if (fd == -1){
14         /* Une erreur est survenue, affiche un message et quitte. */
15         perror("open");
16         return 1;
17     }
18     close(fd);
19     return 0;
20 }
```

V.

A. On a choisi les options O_CREAT et O_EXECL car il faut créer le fichier et vu qu'il est existant désormais on bloque la création de celui-ci à nouveau.

B. On peut constater que le fichier est toujours présent et n'a pas été recrée grâce à la balise O_EXECL.

C.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice5 [master] gcc -o creation_fichier.out creation_fichier.c
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice5 [master] ./creation_fichier.out ./oui
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice5 [master]
```

Exercise 6:

VI.

A.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice6 [master] gcc -o read.out read.c
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice6 [master] ./read.out
J'ai lu 20 chars
Voici les chars lu : oh le joli fichiers
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice6 [master]
```

B. Le programme ouvre le fichier test1.txt et lit 20 caractères de celui-ci.

C. Le programme nous renvoie une erreur car il ne trouve pas le fichier et aussi il n'y pas la balise O_CREAT sur la fonction open();

D. Le programme ne les affiche tout simplement pas car le tampon n'est qu'à 20 caractères.

Exercice 7:

```
write.c

1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6
7  const int N=20;
8
9  int main(void){
10     //Try to open the file in read only mode supposing it exists
11     int f2=open("test1.txt", O_WRONLY);
12     if(f2 == -1){
13         printf("open for write : failed\n");
14         exit(1);
15     }
16     else{
17         char* mon_msg="oh le joli fichier";
18         write(f2,mon_msg,strlen(mon_msg));
19     }
20     return 0;
21 }
```

VII.

E.

```
user@AB-MBP-008 ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice7 [master] file test1.txt
test1.txt: ASCII text, with no line terminators
user@AB-MBP-008 ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice7 [master]
```

F.

```
user@AB-MBP-008 ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice7 [master] gcc -o write.out write.c
user@AB-MBP-008 ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice7 [master] ./write.out
user@AB-MBP-008 ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice7 [master]
```

G.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice7$ cat test1.txt
> cat test1.txt
oh le joli fichier%
```

H.

```
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice7$ cat test1.txt
j'aime linuxichier%
user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice7$
```

Exercice 8:

```
read_write.c

1  #include <stdio.h>
2  #include <sys/stat.h>
3  #include <sys/types.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7  int main(int argc, char *argv[]) {
8      const int MAX=1000;
9      int f1, f2, n;
10     char buf[MAX];
11     //vérification du nombre
12     if (argc != 3){
13         fprintf(stderr, "erreur parametres\n"); exit(1); }
14     //ouverture readOnly du premier argument
15     f1=open(argv[1], O_RDONLY);
16     if( f1 == -1){ /* mode lecture */
17         perror("ouverture fichier source impossible"); exit(2);
18     }
19     //ouverture du deuxième argument en écriture et ?
20     f2=open(argv[2], O_WRONLY|O_CREAT|O_APPEND, S_IRWXU);
21     if( f2 == -1 ){//700
22         perror("creation fichier destinataire impossible"); exit(3);
23     }
24     //lecture, ecriture.
25     while ((n = read(f1, buf, MAX)) > 0){
26         printf("Boucle %d\n", n);
27         write(f2, buf, n);
28     }
29     return 0;
30 }
31
```

VIII.

A.

```
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice8 > master > ./read_write.out ./sourcefile.txt ./destfile
Boucle 46
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice8 > master > cat destfile
Je suis un fichier source en screeeddd laaaaa
user@AB-MBP-008 > ~/Documents/Efreil3/unix-os/TP01/RepTrav/Exercice8 > master >
```

B. La copie du fichier est réalisé car il a été par l'appel `open()` et a été rempli par l'appel `write` en lisant ce que contient le fichier `sourcefile.txt`

C. Les options passés à `open` sont une ouverture en écriture seulement, créer le fichier, et ouvrir une nouvelle ligne au cas où il existe déjà.

Exercice 9:

```
readNWriteName.c

1  #include <stdio.h>
2  #include <sys/stat.h>
3  #include <sys/types.h>
4  #include <fcntl.h>
5  #include <unistd.h>
6  #include <stdlib.h>
7  #include <string.h>
8  int main(int argc, char *argv[]) {
9      const int MAX=1000;
10     int f1, f2, n;
11     char buf[MAX];
12     char *srcFile = "name.txt";
13     char *name = "Dylan KAUFFMANN";
14     //vérification du nombre
15     if (argc != 2){
16         fprintf(stderr, "erreur parametres\n"); exit(1);
17     }
18     //ouverture readAndWrite du premier argument
19     f1=open(srcFile, O_CREAT | O_RDWR, S_IRWXU);
20     if( f1 == -1){ /* mode lecture / ecriture */
21         perror("ouverture fichier source impossible"); exit(2);
22     }
23     write(f1, name, strlen(name));
24     //ouverture du deuxième argument en écriture et ?
25     f2=open(argv[1], O_WRONLY|O_CREAT|O_APPEND, S_IRWXU);
26     if( f2 == -1 ){//?
27         perror("creation fichier destinataire impossible"); exit(3);
28     }
29     //lecture, ecriture.
30     printf("Begin while\n");
31     close(f1);
32     f1=open(srcFile, O_CREAT | O_RDWR, S_IRWXU);
33     if( f1 == -1){ /* mode lecture / ecriture */
34         perror("ouverture fichier source impossible"); exit(2);
35     }
36     while ((n = read(f1, buf, MAX)) > 0){
37         write(f2, buf, n);
38         printf("J'ai lu %d char\n", n);
39     }
40     return 0;
41 }
```


IX.

D.

```
● ✖ user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice9 [master] ./readNWriteName.out ./outFile2
Begin while
J'ai lu 15 char
● user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice9 [master] cat name.txt
Dylan KAUFFMANN
● user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice9 [master] cat ./outFile2
Dylan KAUFFMANN
○ user@AB-MBP-008 ~/Documents/EfreiL3/unix-os/TP01/RepTrav/Exercice9 [master]
```