

# INF203 : Compte-rendu TP7

## Arguments de la ligne de commande et C

### Entrées-sorties fichiers - Fichiers standard

---

Alexandre Dupré, Maxime Jaunatre, Clément Raspail | INF - 3  
[Mail](#) | 23 mars 2021

---

## Syntaxe

Pour ce compte rendu la syntaxe des commandes sera la suivante :

```
[~chemin]: commande  
retour de la commande
```

Exemple :

```
[~/INF203]: ls  
sauve_TP1 TP1 TP2
```

Si la commande est interactive et demande d'appuyer sur entrée, un caractère '`->`' est indiqué. Les fichiers sont en *italique* et les commandes (ou détails de retour de commande) en **gras**. Un script sera donc en gras quand il sera appelé comme une commande.

Les fichiers sources en C sont compilés avec clang, et un nom est donné avec l'option **clang -o**. Cela implique que les programmes seront appelés par un autre nom que **a.out**.

## Pour s'échauffer : quelques gammes avec des chaînes de caractères

[a] **argc** représente le \$# de shell + 1 soit le nombre d'argument donné en paramètre, plus le nom du fichier.

```
0 [~/INF203/TP7]: cat somme.c  
1 #include<stdio.h>  
2  
3 // print arguments with + between and = at the end of the printf  
4 int main (int argc, char *argv[]) {  
5     int i ;  
6     for (i=1 ; i<argc ; i++){  
7         printf("%s ", argv[i]) ;  
8         if(i != argc-1)  
9             printf("+ ");  
10    }  
11    printf("=");  
12    printf("\n");  
13    return 0;  
14 }
```

[b] Quand on ne met pas d'argument la commande renvoie un egale car on l'affiche a la fin de la boucle donc tout le temps.

[c]

```
0 [~/INF203/TP7]: cat somme.c  
1 #include<stdio.h>
```

```

2
3 int char2int(char c[]){
4     int val = 0;
5     val = val + c[0] - '0';
6     for(int i = 1 ; c[i] != '\0' ; i++){
7         val = val * 10 + (c[i] - '0');
8     }
9     return val;
10 }
11
12 int main (int argc, char *argv[]) {
13     int i ;
14     int s = 0 ;
15     for (i=1 ; i<argc ; i++){
16         s = s + char2int(argv[i]);
17         printf("%s ", argv[i]) ;
18         if(i != argc-1)
19             printf("+ ");
20     }
21     printf("= %d", s);
22     printf("\n");
23     return 0;
24 }

```

[~/INF203/TP7]: clang -o somme somme.c ; ./somme 29 10  
29 + 10 = 39

[~/INF203/TP7]: clang -o somme somme.c ; ./somme  
= 0

[d] [e]

[~/INF203/TP7]: clang -o mes\_entrees\_sorties mes\_entrees\_sorties.c  
[~/INF203/TP7]: ./mes\_entrees\_sorties Candidé\_chapitre1.txt truc.txt  
Rentre un nombre correct d'arguments  
[~/INF203/TP7]: ./mes\_entrees\_sorties truc.txt  
truc.txt n'a pas pu être ouvert en lecture  
[~/INF203/TP7]: ls -l Candidé\_chapitre1.txt  
-rw-r--r-- 1 maxime maxime 4992 mars 23 09:01 Candidé\_chapitre1.txt  
[~/INF203/TP7]: chmod u-r Candidé\_chapitre1.txt  
[~/INF203/TP7]: ls -l Candidé\_chapitre1.txt  
--w-r--r-- 1 maxime maxime 4992 mars 23 09:01 Candidé\_chapitre1.txt  
[~/INF203/TP7]: clang -o mes\_entrees\_sorties mes\_entrees\_sorties.c ; ./mes\_entrees\_sorties Candidé\_chapitre1.  
txt  
Candidé\_chapitre1.txt n'a pas pu être ouvert en lecture

[f]

```

0 [~/INF203/TP7]: cat mes_entrees_sorties.c
1 #include <stdio.h>
2
3 int main (int argc, char *argv[]) {
4     FILE *f ;
5     FILE *g ;
6     char c ;
7     if(argc == 2){
8         f = fopen(argv[1], "r") ;
9         if (f == NULL) {

```

```

10         printf("%s n'a pas pu Ãatre ouvert en lecture\n", argv[1]) ;
11         return 2 ;
12     }
13
14     fscanf(f, "%c", &c) ;
15     while (!feof(f)) {
16         printf("%c", c) ;
17         fscanf(f, "%c", &c) ;
18     }
19     printf("\n");
20 } else if (argc == 3) {
21     f = fopen(argv[1], "r") ;
22     g = fopen(argv[2], "a") ;
23     if (f == NULL) {
24         printf("%s n'a pas pu Ãatre ouvert en lecture\n", argv[1]) ;
25         return 2 ;
26     }
27     if (g == NULL) {
28         printf("%s n'a pas pu Ãatre ouvert en lecture\n", argv[2]) ;
29         return 2 ;
30     }
31     fprintf(g, "%c", '\n') ;
32     fscanf(f, "%c", &c) ;
33     while (!feof(f)) {
34         fprintf(g, "%c", c) ;
35         fscanf(f, "%c", &c) ;
36     }
37
38 } else {
39     printf("Rentrez un nombre correct d'arguments (1) \n");
40     return 1 ;
41 }
42 fclose(f) ;
43 return 0 ;
44 }

```

[g] [h] **cat \*.c** affiche dans le terminal le contenu de tout les fichiers dont l'extention est **.c**. **cat** sans arguments entre dans une boucle while attendant une entrée standard pour l'afficher dans le terminal après. Il faut entrer la combinaison de touches *ctrl-d* pour sortir de cette boucle.

[i]

```

0 [~/INF203/TP7]: ./mon_cat mon_cat.c
1 #include <stdio.h>
2
3 int main (int argc, char *argv[]) {
4     FILE *f ;
5     char c ;
6     int i ;
7     if (argc == 1){
8         char c;
9         fscanf(stdin, "%s", &c);
10        while( !feof(stdin)){
11            printf("%s\n", &c);
12            fscanf(stdin, "%s", &c);
13        }
14        return 0;

```

```

15     }
16
17     for(i = 1; i < argc ; i++ ){
18         //printf("l'argument numero %d est %s\n",i, argv[i]) ;
19         f = fopen(argv[i], "r") ;
20         if (f == NULL) {
21             printf("%s n'a pas pu Ãatre ouvert en lecture\n", argv[i]) ;
22             return 2 ;
23         }
24
25         fscanf(f, "%c", &c) ;
26         while (!feof(f)) {
27             printf("%c", c) ;
28             fscanf(f, "%c", &c) ;
29         }
30         printf("\n");
31         fclose(f) ;
32     }
33     return 0 ;
34 }

```

[j]

```

0 [~/INF203/TP7]: cat occurrences.c
1 #include <stdio.h>
2
3 int main(int argc, char *argv[]){
4     FILE* f;
5     int list[128];
6     char c;
7     // init list to 0
8     for (int i = 0; i < 128; i++){
9         list[i] = 0;
10    }
11    // open file
12    f = fopen( argv[1], "r");
13    if ( f == NULL){
14        printf("Rentrez un nom de fichier");
15        return 1;
16    }
17    // scan file
18    fscanf(f, "%c", &c);
19    list[c]++;
20    while(!feof(f)){
21        fscanf(f, "%c", &c);
22        list[c]++;
23    }
24    // print value
25    for (int i = 32; i < 127; i++){
26        // on évite les caractères spéciaux (saut de ligne) et DEL
27        if(list[i] > 0){
28            printf("Il y a %d occurrences du caractere '%c'\n", list[i], i);
29        }
30    }
31
32    return 0;

```

```
[~/INF203/TP7]: ./mon_cat test.txt
Never gonna give you up
Never let you down
[~/INF203/TP7]: ./occurences test.txt
Il y a 7 occurences du caractÃ¨re ' '
Il y a 2 occurences du caractÃ¨re 'N'
Il y a 1 occurences du caractÃ¨re 'a'
Il y a 1 occurences du caractÃ¨re 'd'
Il y a 6 occurences du caractÃ¨re 'e'
Il y a 2 occurences du caractÃ¨re 'g'
Il y a 1 occurences du caractÃ¨re 'i'
Il y a 1 occurences du caractÃ¨re 'l'
Il y a 4 occurences du caractÃ¨re 'n'
Il y a 4 occurences du caractÃ¨re 'o'
Il y a 1 occurences du caractÃ¨re 'p'
Il y a 2 occurences du caractÃ¨re 'r'
Il y a 1 occurences du caractÃ¨re 't'
Il y a 3 occurences du caractÃ¨re 'u'
Il y a 3 occurences du caractÃ¨re 'v'
Il y a 1 occurences du caractÃ¨re 'w'
Il y a 2 occurences du caractÃ¨re 'y'
```