

# INF203 : Compte-rendu TP8

## Unités de compilation en C

---

Alexandre Dupré, Maxime Jaunatre, Clément Raspail | INF - 3  
[Mail](#) | 1<sup>er</sup> avril 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**.

### Gestion de l'ensemble des joueurs

[a] Le nom du type représentant un ensemble de joueurs est **joueurs**. Les joueurs sont représentés sous forme d'un tableau qui regroupe tout les joueurs avec leurs nombres de billes. Le cardinal maximal du nombre de joueurs est 100 et la taille maximum du nom d'un joueur est de 32.

[b]

```
[~/INF203/TP8]: clang billes.c joueurs.c generer_entier.c
```

```
[~/INF203/TP8]: ./a.out essai bob
```

Copie successive des arguments dans l'**ensemble** :

```
essai — bob —
```

Ensemble de joueurs dans lequel la recherche est faite :

```
{ }
```

```
Qui voulez-vous ? essai
```

```
— Joueur absent de l'ensemble
```

```
Qui voulez-vous ? bob
```

```
— Joueur absent de l'ensemble
```

```
Qui voulez-vous ? q
```

[c]

```
0 [~/INF203/TP8]: cat joueurs.c
1 #include <string.h>
2 #include <stdio.h>
3 #include "joueurs.h"
4
5 void init_joueurs(joueurs *ens){
6     ens->nb = 0;
7 }
8
9 int ajouter_joueur(joueurs *ens, char *nom, int billes) {
10     int nb = ens->nb;
11     int place = trouver_joueur(ens, nom);
12
13     if (place == -1) {
14         strcpy(ens->T[nb].pseudo, nom);
15         ens->T[nb].nb_billes = billes;
16         ens->nb++;
17     } else {
18         nb = place;
19     }
20     return nb;
21 }
22
23 int nombre_joueurs(joueurs *ens) {
24     return ens->nb;
25 }
26
27 char *nom_joueur(joueurs *ens, int i) {
28     if (i < ens->nb){
29         return ens->T[i].pseudo;
30     }
31     else
32         return NULL;
33 }
34
35 int billes_joueur(joueurs *ens, int i) {
36     if (i < ens->nb){
37         return ens->T[i].nb_billes;
38     }
39     else
40         return 0;
41 }
42
43 int trouver_joueur(joueurs *ens, char *nom) {
44     int test = -1;
45     for (int i = 0; i < ens->nb && test == -1; i++){
46         if (!strcmp(nom, ens->T[i].pseudo)){
47             test = i;
48         }
49     }
50     return test;
51 }
```

[~/INF203/TP8]: clang -obille billes.c joueurs.c generer\_entier.c

```

[~/INF203/TP8]: ./bille gizmo stripe mugger flasher bogart stripe
Copie successive des arguments dans l'ensemble :
  gizmo – stripe – mugger – flasher – bogart – stripe –
Ensemble de joueurs dans lequel la recherche est faite :
{ gizmo 402 stripe 223 mugger 350 flasher 90 bogart 67 }
Qui voulez-vous ? gizmo
402
Qui voulez-vous ? stripe
223
Qui voulez-vous ? mogway
-- Joueur absent de l'ensemble
Qui voulez-vous ? q

```

## Lecture et écriture d'un ensemble dans un fichier

[d]

```

0 [~/INF203/TP8]: cat joueurs_out.c
1 #include <stdio.h>
2 #include <string.h>
3 #include "joueurs.h"
4
5 void ecrire_les_joueurs(joueurs *ens, char *nom_fich){
6     FILE *f;
7     f = fopen(nom_fich, "w");
8
9     //write number
10    fprintf(f, "%d\n", ens->nb);
11    //write players
12    for (int i = 0; i < ens->nb; i++){
13        for(int ii = 0; ens->T[i].pseudo[ii] != '\0'; ii++){
14            fprintf(f, "%c", ens->T[i].pseudo[ii]);
15        }
16        fprintf(f, " %d\n", ens->T[i].nb_billes);
17    }
18    fclose(f);
19 }

```

[e]

```

[~/INF203/TP8]: tail -n 3 billes.c
    ecrire_les_joueurs(&ens_joueurs, "gremlins.txt");
    return 0;
}
[~/INF203/TP8]: clang -obille billes.c joueurs.c generer_entier.c joueurs_out.c
[~/INF203/TP8]: ./bille gizmo stripe
Copie successive des arguments dans l'ensemble :
  gizmo – stripe –
Ensemble de joueurs dans lequel la recherche est faite :
{ gizmo 479 stripe 204 }
Qui voulez-vous ? bob
-- Joueur absent de l'ensemble
Qui voulez-vous ? gizmo
479
Qui voulez-vous ? q
[~/INF203/TP8]: cat gremlins.txt

```

Il serait possible de prendre le dernier argument de la commande **billes** pour donner le nom de fichier.

[f] *Etant donné que l'on a enregistré les joueurs dans gremlins.txt, on charge ce fichier.*

```

0  [~/INF203/TP8]: cat joueurs_in.c
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include "joueurs.h"
5  #include "joueurs_in.h"
6
7  joueurs lire_les_joueurs(char *nom_fich){
8      FILE *f;
9      f = fopen(nom_fich, "r");
10     joueurs ens;
11     char nb_string[3] = "";
12     char c;
13     char nb_j[32] = "";
14     char nb_bille[4] = "";
15     int nb;
16
17     //init
18     init_joueurs(&ens);
19
20     // lecture—écriture des joueurs
21     fscanf(f, "%c", &c);
22     while (c!='\n') {
23         strncat(nb_string, &c,1);
24         fscanf(f, "%c", &c);
25     }
26     nb = atoi(nb_string);
27     printf("string : %d\n", nb);
28
29     for(int i = 0; i < nb ; i++){
30         // noms des joueurs
31         strcpy(nb_j, "");
32         fscanf(f, "%c", &c);
33         while (c!=' ') {
34             strncat(nb_j, &c,1);
35             fscanf(f, "%c", &c);
36         }
37         // nb billes joueurs
38         strcpy(nb_bille, "");
39         fscanf(f, "%c", &c);
40         while (c!='\n' && !feof(f)) {
41             strncat(nb_bille, &c,1);
42             fscanf(f, "%c", &c);
43         }
44         ajouter_joueur(&ens, nb_j, atoi(nb_bille));
45     }
46
47     fclose(f);

```

```

48     return ens;
49 }

```

Le fichier de sortie n'est pas modifié, du moins il est remplacé par exactement les mêmes valeurs, aucune modifications ne sont faites. Pour le vérifier, il suffit d'enregistrer les joueurs dans un nouveau fichier et d'utiliser la commande suivante : **diff ancien\_joueur.txt nouveau\_joueur.txt**.

### 3 Gestion d'un ensemble

[g]

```

[~/INF203/TP8]: tail -n3 joueurs.c
void modifier_billes(joueurs *ens, int i, int cpt) {
    ens->T[i].nb_billes = cpt;
}
[~/INF203/TP8]: clang -ogestion joueurs*.c gestion.c operations.c
[~/INF203/TP8]: ./gestion gremlins.txt mogway.txt
gizmo 113
stripe 157

```

```

Requête (G,P,T,Q) : T
Nom du joueur qui gagne ? gizmo
Nom du joueur qui perd ? stripe
Montant ? 40

```

```

Requête (G,P,T,Q) : Q
gizmo 153
stripe 117

```

```

40 [~/INF203/TP8]: tail -n23 operations.c
41 int transfert(joueurs *e) {
42     /***** A COMPLETER *****/
43     int i, j, montant, soldeg, soldep ;
44     char nomp[TAILLE_MAX_NOM], nomg[TAILLE_MAX_NOM] ;
45
46     printf("Nom du joueur qui gagne ? ") ;
47     scanf("%s", nomg) ;
48     printf("Nom du joueur qui perd ? ") ;
49     scanf("%s", nomp) ;
50     i = trouver_joueur(e, nomp);
51     j = trouver_joueur(e, nomg);
52     if (i == -1 || j == -1) {
53         printf("Inconnu\n") ; return 1 ;
54     }
55     printf("Montant ? ") ;
56     scanf("%d", &montant) ;
57
58     soldeg = billes_joueur(e, j) ;
59     soldep = billes_joueur(e, i) ;
60     modifier_billes(e, j, soldeg+montant) ;
61     modifier_billes(e, i, soldep-montant) ;
62     return 0 ;
63 }

```