INF203 : Compte-rendu TP8
Unités de compilation en C

Alexandre Dupré, Maxime Jaunatre, Clément Raspail | INF - 3
Mail | 30 mars 2021

Syntaxe

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

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

Example :

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

Si la commande est interactive et demande d'appuyer sur entrée, une 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 et **joueurs**. Les joueurs sont representés sous forme d'un tableau qui regroupe tout les joueurs avec leurs nombres de billes. Le cardinal maximal du nombre de joueur 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
```

On note qu'on ne peut pas compiler plusieurs fichier et spécifier le nom de sortie avec l'option **-o**.

[c]

```
[~/INF203/TP8]: cat joueurs.c
#include <string.h>
#include <stdio.h>
#include "joueurs.h"

void init_joueurs(joueurs *ens){
  ens->nb = 0;
}

int ajouter_joueur(joueurs *ens, char *nom, int billes) {
  int nb = ens->nb;
  int place = trouver_joueur(ens, nom);

  if (place == -1) {
    strcpy(ens->T[nb].pseudo, nom);
    ens->T[nb].nb_billes = billes;
    ens->nb++;
  } else {
    nb = place;
  }
  return nb;
}

int nombre_joueurs(joueurs *ens) {
  return ens->nb;
}

char *nom_joueur(joueurs *ens, int i) {
  if (i < ens->nb){
    return ens->T[i].pseudo;
  }
  else
    return NULL;
}

int billes_joueur(joueurs *ens, int i) {
  if (i < ens->nb){
    return ens->T[i].nb_billes;
  }
  else
    return 0;
}

int trouver_joueur(joueurs *ens, char *nom) {
  int test = -1;
  for (int i = 0; i < ens->nb && test == -1; i++){
    if (!strcmp(nom, ens->T[i].pseudo)){
      test = i;
    }
  }
  return test;
}
```

```
[~/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
```

[d]

```
[~/INF203/TP8]: cat joueurs_out.c
#include <stdio.h>
#include <string.h>
#include "joueurs.h"

void ecrire_les_joueurs(joueurs *ens, char *nom_fich){
    FILE *f;
    f = fopen(nom_fich, "w");

    //write number
    fprintf(f, "%d\n", ens->nb);
    //write players
    for (int i = 0; i < ens->nb; i++){
        for(int ii = 0; ens->T[i].pseudo[ii] != '\0'; ii++){
            fprintf(f, "%c", ens->T[i].pseudo[ii]);
        }
        fprintf(f, " %d\n", ens->T[i].nb_billes);
    }
    fclose(f);
}
```

[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
2
gizmo 479
stripe 204
```

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.*

```
[~/INF203/TP8]: cat joueurs_in.c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "joueurs.h"
#include "joueurs_in.h"

joueurs lire_les_joueurs(char *nom_fich){
    FILE *f;
    f = fopen(nom_fich, "r");
    joueurs ens;
    char nb_string[3] = "";
    char c;
    char nb_j[32] = "";
    char nb_bille[4] = "";
    int nb;

    //init
    init_joueurs(&ens);

    // lecture-ecriture des joueurs
    fscanf(f, "%c", &c);
    while (c!='\n') {
        strncat(nb_string, &c,1);
        fscanf(f, "%c", &c) ;
    }
    nb = atoi(nb_string);
    printf("string : %d\n", nb);

    for(int i = 0; i < nb ; i++){
        // noms des joueurs
        strcpy(nb_j, "");
        fscanf(f, "%c", &c);
        while (c!=' ') {
            strncat(nb_j, &c,1);
            fscanf(f, "%c", &c) ;
        }
        // nb billes joueurs
        strcpy(nb_bille, "");
        fscanf(f, "%c", &c);
        while (c!='\n' && !feof(f)) {
            strncat(nb_bille, &c,1);
            fscanf(f, "%c", &c) ;
        }
        ajouter_joueur(&ens, nb_j, atoi(nb_bille));
    }

    fclose(f);
    return ens;
}
```