
Travaux pratiques EntréesSorties

Dans ce thème, nous verrons comment un programme écrit en C interagit avec son environnement, que ce soit un terminal (la fenêtre dans laquelle il est exécuté) ou des fichiers.

I Exercice : Premier programme

On va travailler sur les types, donc dans le répertoire **src/Entrées-Sorties/**.

(a) Vérifiez que vous êtes bien sur la branche **etudiant**, et si ce n'est pas le cas, allez sur cette branche.

```
git branch
git checkout etudiant
```

(b) Créez le fichier **~/workspace/src/Entrées-Sorties/exercice\arabic{exercicenum}.c** et ouvrez-le.

(c) Tapez le code suivant dans le fichier:

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

```
int
main ( int argc , char * argv [] )
{
    return 0 ;
}
```

(d) Sauvegardez le fichier (avec Ctrl-S).

On va maintenant compiler et exécuter ce programme.

(e) Changer de répertoire pour aller dans **~/workspace/src/Entrées-Sorties**.

```
cd ~/workspace/src/Entrées-Sorties
```

(f) Compiler le programme avec la commande:

```
gcc -o exercice%\arabic{exercicenum}% exercice%\arabic{exercicenum}%.c
```

(g) Exécutez le programme. Que se passe-t'il ?

```
./exercice%\arabic{exercicenum}%
```

Il ne se passe rien.

(h) Ajoutez la ligne suivante juste au dessus de **return 0**:

```
printf ( " Bonjour . " ) ;
```

et compilez et exécutez le programme.

`printf` est une *fonction* qui permet à un programme d'envoyer des données au terminal dans lequel il est exécuté.

(i) Ajoutez `\n` après Bonjour, compilez et exécutez le programme.

`+` est un caractère qui demande au terminal de passer à la ligne suivante.

II Exercice : Affichage d'un entier.

Le 'f' de `printf` signifie *mettre en forme*. Cette fonction est un peu spéciale car contrairement aux fonctions normales en C, elle peut prendre un nombre variable d'arguments, mais il en faut au moins 1.

Le premier argument est une *chaîne de formatage* qui sert à mettre en forme les données qui sont contenues dans les autres arguments, dans l'ordre de lecture. Pour cela, la chaîne de formatage contient des *directives d'affichage*. La première directive sera utilisée pour afficher la première donnée après la chaîne de formatage, la seconde directive sera utilisée pour afficher la seconde donnée après la chaîne de formatage, etc.

Directives d'affichage. Une directive commence toujours par `%`. Il y en a des dizaines, donc on va juste en regarder quelques unes:

- `% d`: permet d'afficher un entier en base dix;
- `% c`: permet d'afficher une lettre (un caractère) à partir d'un entier;
- `% f`: permet d'afficher un nombre à virgule;
- `% s`: permet d'afficher une chaîne de caractères;

Pour l'exercice 2, on commence par copier le contenu du programme `exercice1.c` dans un nouveau fichier