Fiche n° 1 de TP

Découverte de l'environnement de travail et premiers programmes

Objectifs : utiliser le terminal pour réaliser les manipulations de base sur le système de fichiers, écrire un programme et le compiler dans l'environnement de travail.

Prérequis : commandes shell de gestion des fichiers, connaitre la structure minimale d'un programme C.

Les précisions qui figurent ici sont valables pour la matière « bases de la programmation impérative » (BPI pour faire court), enseignée au premier semestre, mais également pour la matière « algorithmique 1 » du second semestre. Les salles de TP sont celles du rez-de-chaussée du bâtiment U2 du site du Madrillet.

Rendu

À la fin de chaque séance de TP, vous devez IMPÉRATIVEMENT déposer vos fichiers sources sur la plateforme UniversiTice, dans la section dédiée au TP (à l'endroit même où vous avez récupéré la fiche de TP). Le dépôt de vos fichiers devra se faire *via* une archive au format zip contenant un répertoire nommé TP*n* (pour le rendu du TP numéro *n*) dans lequel se situeront vos fichiers sources.

Visibilité

Pour accéder à vos programmes en dehors des salles de TP, vous pouvez utiliser l'espace de stockage qui vous est réservé sur l'ENT. Il s'agit de la rubrique **Stockage/Espace de stockage** du menu de l'ENT.

Travailler sous Linux

Les séances de TP se dérouleront sous Linux.

Exercice 1

1) Ouvrez un terminal. Quel est le répertoire courant ? Quels sont les fichiers et les répertoires qu'il contient ?

2) Placez-vous dans le répertoire `ENT-`*login* où *login* désigne votre identifiant de connexion.

3) Créez un répertoire `BPI` dans `ENT-`*login* puis un répertoire `TP1` dans `BPI`. Vous devrez par la suite créer en début de chaque TP un répertoire adéquat dans `BPI`.

4) Téléchargez l'archive `TP1_src.zip` dans le répertoire `BPI` et décompressez la à l'aide de la commande unzip.

5) Copiez le fichier `TP1_src/bonjour.c` dans le répertoire `TP1`.

6) Affichez sur la console le contenu du fichier `bonjour.c`.

7) Compilez le fichier `bonjour.c` (la commande ci-dessous doit être écrite sur une seule ligne) :

```
gcc -c -std=c2x -Wall -Wconversion -Werror -Wextra -Wfatal-errors -Wpedantic
    -Wwrite-strings -O2 bonjour.c
```

puis réalisez l'édition des liens :

```
gcc bonjour.o
```

8) Quel est le nom de l'exécutable créé ? Renommez le en `bonjour`. Comment peut-on spécifier le nom de l'exécutable à créer lors de la compilation ?

9) Quels sont les droits associés au fichier exécutable ?

10) Exécutez le programme `bonjour`. Que se passe-t-il ?

11) Supprimez le fichier `bonjour.o` créé lors de la compilation.

12) Supprimez le répertoire `TP1_src` créé lors de la décompression de l'archive `TP1_src.zip`.

Travailler avec l'EDI

Réglage de l'EDI Geany

Geany est l'environnement de développement intégré (EDI) que vous utiliserez pour la programmation en C. Il permet tout à la fois d'éditer des programmes, de les compiler et de les exécuter.

Pour une utilisation en conformité avec les normes actuelles, il est nécessaire de réaliser quelques réglages.

Lancez Geany en cliquant sur l'icône **Geany** présente sur le bureau.

1) Réglages de l'éditeur. Dans la fenêtre **Éditer** > **Préférences** :

- **a)** *suppression du bip.* Sous l'onglet **Général** > **Divers**, décochez la case **Émettre un bip…** ;
- **b)** *paramétrage de l'indentation.* Sous l'onglet **Éditeur** > **Indentation**, fixez la valeur du paramètre **Largeur** à 2 et celle du paramètre **Type** en cochant le bouton **Espaces** ;
- **c)** *guides d'indentation.* Sous l'onglet **Éditeur** > **Affichage**, cochez la case **Afficher les guides d'indentation** ;
- **d)** *guide de limite droite.* Sous l'onglet **Éditeur** > **Affichage**, fixez la valeur du paramètre **Colonne** du marqueur des longues lignes à 80.

Enregistrez ces modifications en cliquant sur le bouton **Valider**.

2) Réglages du compilateur. Il s'agit ici d'ajouter de nouvelles options aux commandes de compilation et de construction fournies par défaut par Geany.

Cliquez sur **Document** > **Définir le type de fichier** > **Langages de Programmation** > **Fichier source C**. Dans la fenêtre **Construire** > **Définir les commandes de construction**, fixez les valeurs du champ **Commande** des paramètres :

a) **Compiler** à :

```
gcc -c -std=c2x -Wall -Wconversion -Werror -Wextra
    -Wfatal-errors -Wpedantic -Wwrite-strings -O2 "%f"
```

b) **Construire** à :

```
gcc -std=c2x -Wall -Wconversion -Werror -Wextra
    -Wfatal-errors -Wpedantic -Wwrite-strings -O2 -o "%e" "%f"
```

Cliquez sur le bouton **Valider**.

Nouveau fichier

Afin de simplifier la création d'un nouveau fichier `C`, nous allons créer un modèle de fichier. Voici comment procéder :

- Ouvrez une fenêtre de gestion de fichier.
- Télecharger et copier le fichier `main.c` dans le dossier `~/Modèles`.
- Ouvrez un terminal dans le dossier et créez un lien vers ce fichier dans le répertoire : `~/.config/geany/templates/files` :

```
ln main.c ~/.config/geany/templates/files/main.c
```

Depuis la fenêtre de gestion de fichier :

- Allez dans le répertoire de travail du TP1 puis ,avec le bouton droit, ouvrez le menu **Créer Document** > **main.c**.
- Renommez le fichier `main.c` en `premier.c`.
- Cliquez avec le bouton droit sur le fichier et choisisser **Propriétés…** dans le menu.
- Modifier le champ **Ouvrir avec** en choisissant le logiciel Geany.

Maintenant, si vous double-cliquez sur `premier.c`, le fichier s'ouvre dans l'EDI Geany.

Pour créer un nouveau fichier C sous l'EDI Geany :

- Vous pouvez cliquer sur l'icône **Nouveau**. Renommez immédiatement après ce fichier et sauvegardez-le : dans la fenêtre **Fichier** > **Enregistrer sous**, sélectionnez le dossier dans lequel il doit être rangé (si nécessaire, créez ce dossier à l'aide du bouton **Créer un dossier**), choisissez le nom, nécessairement suivi de l'extension « `.c` », puis cliquez sur le bouton **Enregistrer**.

– Vous pouvez choisir **Nouveau (selon modèle)** dans le menu **Fichier** et choisir **main.c**. Le fichier créé sera automatiquement au bon format. Il ne restera plus qu'à le sauvegarder à l'endroit désiré en prenant bien soin de garder l'extension « `.c` ». Par exemple, vous pouvez créer le fichier « `second.c` » dans le répertoire du TP1.

Exécution

Les programmes valides peuvent être lancés directement sous l'EDI Geany en appuyant simplement sur le bouton **Exécuter**.

Il sera parfois nécessaire d'exécuter les programmes en ligne de commande. Vous pourrez alors utiliser le terminal ou plus simplement cliquer sur l'icône de l'invite de commande dans l'EDI.

Exercice 2

Ouvrez à l'aide dans l'EDI Geany le fichier `bonjour.c`. Construisez puis exécutez le programme.

Exercice 3

1) Tapez le programme suivant dans le fichier `aff-val.c` :

aff-val.c

```c
/**
 * Équipe pédagogique BPI
 * Exemple de manip I/O.
 */

/* Appel des bibliothèques */
#include <stdio.h>
#include <stdlib.h>
#include <assert.h>

/* Déclarations des fonctions */

/* Fonction principale */

int main(void) {
  printf("Entrez une valeur entiere :\n");

  int k;
  assert(scanf("%d", &k) == 1);

  printf("La valeur entree est : %d\n", k);

  return EXIT_SUCCESS;
}

/* Définitions des fonctions */
```

aff-val.c

Testez-le.

2) Remplacez la ligne

```c
  assert(scanf("%d", &k) == 1);
```

par

```c
  scanf("%d", &k);
```

Que remarquez vous ?
Réécrivez la ligne initiale.

3) Modifiez-le afin qu'il affiche la somme de deux valeurs entières lues sur l'entrée.

Exercice 4

Inspirez-vous de l'exercice précédent pour produire, dans le fichier `calc.c`, un programme qui, pour deux valeurs entières lues sur l'entrée, affiche leur somme, leur différence, leur produit, leur quotient et leur moyenne.

Exercice 5

Tapez le programme suivant :

syntax-error.c

```
/**
 * Équipe pédagogique BPI
 * À corriger!
 */

/* Appel des bibliothèques */
#include <stdio.h>
#include <stdlib.h>

/* Déclarations des fonctions */

/* Fonction principale
int main
entier p, q; réel   x,y;

  scanf("%d %f %d", p,q,x)
            y=p * x
              /q
              print('%f', &y);
  return SUCCESS_STORY;
}
```

syntax-error.c

Modifiez ce programme de manière à ce qu'il compile sans erreur ni avertissement.