POLYTECH LILLE

Université de Lille

Département IMA / 3A (S5)

TP 6 : Redirections, algorithmes simples de chaînes

Objectifs

Manipuler les entrées/sorties C (via les redirections) pour la deuxième fois. Implémenter en C quelques algorithmes de chaînes de caractères.

Avant de commencer, créer un répertoire pour le TP6.

1 Expérimentations avec les redirections

On fournit (à l'URL http ://www.cristal.univ-lille.fr/~grisoni/IMA/) dans le fichier `tpchaines.c` une procédure et un programme principal :

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

void mange_et_reecrit(void)
{
  char c;
  c = getchar();
  while ( c != EOF )
    {
      putchar(c);
      c = getchar();
    }
}

int main(void)
{
  mange_et_reecrit();
  return 0;
}
```

On demande de compiler le programme et de l'exécuter :

1. Que fait ce programme ?
2. Écrire un fichier de nom `toto.txt` qui contient :

   ```
   truc
   bidule
   plouf
   ```

   (pas de saut de ligne après `plouf`).

3. Dans le terminal, faire `./nombinaire <toto.tx`. Que ce passe-t-il ?

4. Que va-t-il se passer si on fait `./nombinaire <toto.txt >titi.txt` ? Vérifier. Comparer avec la commande `cp` d'Unix.

Remarque : on pourrait utiliser scanf au lieu de getchar.

2 Algorithmes de chaînes sur les fichiers

Pour les exercices suivants, on écrira les programmes sur le modèle précédent (**un .c par question**).

1. Écrire un programme qui compte le nombre de caractères (y compris les espaces, les sauts de lignes, les tabulations) de l'entrée standard. Utiliser les redirections pour compter le nombre de charactères du fichier `toto`.
2. Écrire un programme qui imprime les lettres qui ne sont pas apparues dans le fichier (on suppose que les lettres sont uniquement des minuscules).
   Indication : On pourra utiliser un tableau de 26 *cases que l'on mettra à jour à chaque fois que l'on rencontrera une lettre. Comment trouver l'indice de la case à mettre à jour ?*
3. Écrire un programme qui écrit le fichier à l'envers. Sur le fichier `toto`, le programme donnera :

   ```
   fuolp
   eludib
   curt
   ```

   Indication : On se demandera comment mémoriser les caractères du fichier
4. Écrire un programme qui dit si un certain mot (un "motif") est présent dans le fichier.
5. Ecrire un programme qui détermine le mot le plus long d'un fichier.