INF 111, L1, 2004/2005

Feuille 11

Novembre 2004

Redirections Unix et chaînes de caractères

Laure Gonnord
`http://www-verimag.imag.fr/~gonnord/`

1 Quelques expérimentations

On fournit dans le fichier `copy.c` une procédure et un programme principal :

```
#include <stdio.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 (`gcc -Wall -o copy copy.c`) et de l'exécuter :

- Que fait ce programme ?
- Écrire un fichier de nom `toto` (sans extension, ce n'est pas un programme), qui contient :
  `truc`
  `bidule`
  `plouf`
  (pas de saut de ligne après `plouf`).
- Dans le terminal, faire `copy <toto`. Que ce passe-t-il ?
- Que va-t-il se passer si on fait `copy <toto >titi` ? Vérifier. Comparer avec la commande `cp` d'Unix.

2 À faire

▶ Pour les exos ci-dessous, on écrira les programmes sur le modèle précédent. Les programmes ne sont pas par ordre croissant de difficulté.

- Écrire un programme qui compte le nombre de caractères (y compris les espaces, les sauts de lignes, les tabulations) d'un fichier.
- Écrire un programme qui compte le nombre de lignes d'un fichier.
  Attention ! le programme nedit modifie un petit peu les fichiers que vous enregistrez. Si vous ne mettez pas de saut de ligne à la fin de votre fichier, il en crée un. Tenez-en compte lors du test de vos programmes.
- Écrire un programme qui compte le nombre de mots d'un fichier. Comparer avec la commande `wc` d'Unix.
  Indication : On se posera la question "qu'est-ce qu'un mot ?" et on pourra écrire deux fonctions auxiliaires `manger_mot` *et* `manger_blancs`.
- Écrire un programme qui lit une phrase terminée par un point et qui remplit un tableau avec cette phrase (blancs compris).
- 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
- Écrire un programme qui écrit le fichier dans le sens inverse des mots. Sur le fichier `toto`, le programme donnera :
  `plouf`
  `bidule`
  `truc`
  On pourra se contenter de ne mettre qu'un seul espace entre chaque mot :
  `ploup bidule truc`
  Indication : Comme dans l'exercice précédent, on va stocker les mots. De quel type de tableau a-t-on besoin ?
- Écrire un programme qui liste des 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 ?*
- Écrire un programme qui dit si le fichier est un palindrome. Sur le fichier `toto` il répondra `FAUX` (0) et sur le fichier contenant `tu l as trop ecrase cesar ce port salut` il répondra `VRAI` (1). On ignore les espaces.
- Écrire une FONCTION qui prend en argument deux chaînes de caractères (sous forme de tableaux de caractères de taille `MAX` dans lesquels les fins de chaînes sont repérées par `'\0'`) et qui dit si ces deux chaînes sont anagrammes l'unes de l'autre. Par exemple, `marine` et `irmane` sont anagrammes.
- Écrire un programme qui dit si un certain mot (un "motif") est présent dans un fichier.
- Écrire un programme qui donne le codage de César du fichier. Le codage de César est obtenu en remplaçant les caractères par le 13ième caractère suivant : le caractère `'a'` est remplacé par le caractère `n` (et vice-versa). Ainsi le mot `marine` est remplacé par `znevar`. On fera attention à ne pas modifier les blancs du fichier.
- Écrire un programme qui dit si les parenthèses d'un fichier sont bien imbriquées. On ignore les autres caractères. Dans `(())()` les parenthèses sont bien imbriquées, mais pas dans `((((()(`.
- On suppose que l'on dispose d'un certain nombre de MOTS "interdits" (stockés comment ?). Écrire un programme qui réécrit un fichier en un fichier dans lequel les mots interdits ont été remplacés par autant d'étoiles que leur taille. Par exemple, si `ploup` est interdit, le texte `toto ploup trucploup` est remplacé par `toto ***** trucploup`.