

INF203 - Travaux pratiques, séance 5

Arguments de la ligne de commande et C Entrées-sorties fichiers - Fichiers standard

Arguments de la ligne de commande en C

[TP5] Lisez, compilez et exécutez *aff_arg.c*.

[a] Que représente `argc`? ■

Dupliquez le programme *aff_arg.c* sous le nom *somme.c*. Modifiez *somme.c* pour qu'il affiche uniquement chacun de ses arguments : à l'exécution, le programme *somme* aura le comportement suivant :

```
./somme abc 678 groumph  
abc 678 groumph
```

Modifiez *somme.c* afin que le programme *somme* affiche des signes '+' et '=' entre les arguments, de la manière suivante :

```
./somme abc 678 groumph  
abc + 678 + groumph =  
./somme ploum  
ploum =
```

[b] Qu'affiche votre programme lorsqu'aucun argument ne lui est fourni? ■

Dans la suite, on supposera que les arguments donnés au programme *somme* sont des chaînes de caractères représentant des entiers positifs, c'est-à-dire constituées des caractères {'0','1','2',...,'9'}. Complétez et modifiez *somme.c* pour que le programme *somme* calcule et affiche la somme des entiers représentés par ses arguments :

```
./somme 19 5 18  
19 + 5 + 18 = 42
```

[c] Vérifiez que tout se passe bien lorsqu'aucun argument n'est fourni, au besoin corrigez, et joignez le texte de votre programme final à votre compte-rendu. ■

Lecture depuis un fichier, écriture dans un fichier

Lisez, compilez et exécutez *lecture_fich.c*. Dupliquez le sous le nom *mes_entrees_sorties.c*. Modifiez ce nouveau programme pour que le nom du fichier lu soit donné en argument de la ligne de commande. Lorsque le nombre d'arguments est incorrect, votre programme affichera un message d'erreur approprié et terminera son exécution. Compilez et testez : avec un fichier existant, avec un fichier qui n'existe pas, avec trop ou trop peu d'arguments ...

[d] Quel message d'erreur affichez vous si le nombre d'arguments est incorrect? ■

Enlevez-vous les droits en lecture sur *Candide_chapitre1.txt* : que va-t-il se passer si vous donnez ce nom de fichier en argument à votre programme? Vérifiez ...

Dupliquez *mes_entrees_sorties.c* sous le nom *mon_cat.c*. Ce programme servira un peu plus loin. Modifiez *mes_entrees_sorties.c* pour qu'il prenne deux arguments qui sont deux noms de fichiers, et copie le contenu du premier fichier dans le second. Votre programme affichera des messages d'erreur appropriés si nécessaire.

Exercice complémentaire :

Modifiez *mes_entrees_sorties.c* pour qu'il ait le comportement suivant :

- avec 2 arguments, il se comporte comme précédemment
- avec 1 nom de fichier en argument, il affiche le contenu de ce fichier à l'écran
- il affiche un message d'erreur dans les autres cas.

Indication : l'écran est associé de façon standard à une variable de type `FILE*` appelé `stdout`. Utilisez cette information pour apporter le minimum de modifications à votre programme.

[e] Joignez le texte de *mes_entrees_sorties.c* à votre compte-rendu (avec ou sans exercice complémentaire) ■

Réviser le comportement de la commande *cat*.

[f] Quel est le résultat de l'exécution de la commande `cat *.c` ? ■

Modifiez *mon_cat.c* pour que *mon_cat* se comporte de la même façon que *cat* lorsqu'il a au moins 1 argument.

Exercice complémentaire :

[g] Quel est le comportement de la commande *cat* sans argument ? ■

Modifiez votre programme *mon_cat.c* pour reproduire ce comportement.

Indications :

1. le clavier est associé à la variable `stdin` de type `FILE*`.
2. pour simuler la fin de fichier sur `stdin`, il faut entrer la combinaison de touches <CTRL-D> en début de ligne.

[h] Joignez le texte de *mon_cat.c* à votre compte-rendu (avec ou sans exercice complémentaire) ■

Caractères imprimables

Écrivez un programme *occurences.c* qui prend en argument un nom de fichier, et affiche à l'écran le nombre d'occurences (si celui-ci est non nul) de chaque caractère imprimable de ce fichier. Exemple :

```
cat essai
ABvvvAA
```

```
vwx
wwwwwwAwwwAB
AAA
```

```
./occurences essai
Il y a 8 occurences du caractère A
Il y a 2 occurences du caractère B
Il y a 4 occurences du caractère v
Il y a 10 occurences du caractère w
Il y a 1 occurences du caractère x
```

[i] Comment faites-vous pour retrouver l'ensemble des caractères imprimables ? Joignez le texte de *occurences.c* à votre compte-rendu. ■

La commande de la semaine : find

La commande *find* permet de (re-)trouver des fichiers ou répertoires dans l'arborescence. Elle prend en argument un répertoire <R> et une expression <E> qui décrit un critère de recherche, et affiche à l'écran les fichiers ou sous-répertoires de <R> qui satisfont le critère décrit par <E>. Exécutez les commandes suivantes, observez et comprenez ce que vous obtenez :

- `find ~ -name "*.c"` : tous les fichiers qui se terminent par `.c` dans l'ensemble de vos répertoires.
- `find ~/INF203 -type d` : tous les sous-répertoires de votre répertoire *INF203*.
- `find ~/INF203 -executable` : les fichiers et répertoires pour lesquels vous avez le droit `x`.
- `find $HOME -type f -executable` : tous vos fichiers exécutables.
- `find . -mtime 0` : tous les fichiers modifiés depuis moins de 24 heures.

[j] Combien y-a-t-il de fichiers appelés *gcc* dans le répertoire */opt* ? ■