TP de Système n° 8

Exercice 1 : "`wc`"

Réimplémentez la commande "`wc`", qui compte le nombre de lignes, mots, et caractères présents dans un fichier. Vous réimplémenterez les options "`-c`", "`-l`", "`-w`" et "`-L`".

Exercice 2 : Table de vérité

1. Écrire une fonction `operation(char a, char b, char op)` qui renvoie un booléen donnant le résultat de l'opération binaire « `a op b` », où `op` peut avoir l'une des valeur suivantes : `'+'` (ou), `'.'` (et), ou `'^'` (ou exclusif).
2. Écrire une fonction `parenthese_fermante(char* parenthese_ouvrante)` qui, en fonction d'un pointeur de caractère vers une parenthèse ouvrante, renvoie un pointeur de caractère vers la parenthèse fermante correspondante.
3. Écrire une fonction `assigner(char* chaine1, char *chaine2, char a, char b, char c)` qui copie une chaîne de caractères en remplaçant les caractères `a`, `b` et `c` par leur valeur booléenne donnée en paramètre. Écrire la fonction `main` qui assigne la chaîne donnée en argument pour tous les triplets de valeurs possibles de `a`, `b` et `c`.
4. Écrire une fonction `evaluer(char* debut, char* fin)` qui renvoie un booéen donnant la valeur de la chaîne entre `debut` et `fin` en lançant deux fils pour chaque opérateur rencontré (chaque fils évaluant un côté de l'opérateur). On utilisera les appels système `wait` et `exit` pour remonter la valeur calculée au processus père.
5. Écrire un programme `verite` prenant en paramètre une expression de la forme `((a+b).(c^b)).(c+b)` et donnant la table de vérité. On suppose que l'on a seulement les trois variables binaires `a`, `b` et `c`.