TP de Système n° 10

Exercice 1 :

Écrire un programme qui évalue une expression en notation polonaise passée en ligne de commande. On suppose que les opérandes sont des entiers et que les opérations de base donnent des résultats entiers. On utilisera les notations '+', '-', 'x' (et non '*') et '/' pour les opérateurs.

Une expression en notation polonaise se présente récursivement de la façon suivante : 'opérateur expression1 expression2' et signifie 'expression1 opérateur expression2'. Vous trouverez à l'adresse suivante `http://www.liafa.jussieu.fr/~amicheli/Ens/Systeme/` le fichier `parseur.c` qui contient la fonction `parseur` prenant en argument un tableau de chaînes de caractères, représentant une expression en notation polonaise, et renvoyant l'indice du début du tableau correspondant à l'expression expression2.

Le programme devra récursivement faire évaluer les expressions expression1 et expression2 par deux processus fils.

1. Pour communiquer avec les fils, utiliser des fichiers temporaires dans lesquels le processus père lira les résultats intermédiaires.
2. Que se passe-t-il si l'on omet l'appel à `wait()` avant de lire dans le fichier ?
3. Remplacer les fichiers temporaires par des tubes.

```
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <fcntl.h>

void erreur(const char *mess) {
  perror(mess); exit(2);
}
void usage(const char *s) {
  fprintf(stderr, "Usage : %s reference\n", s); exit(1);
}
int main(int argc, char *argv[]) {
  int d1, d2, d3;
  char buf[3];

  if (argc != 2) usage(argv[0]);
  if ((d1 = open(argv[1], O_RDONLY)) == -1) erreur("open 1");
  if ((d2 = open(argv[1], O_WRONLY)) == -1) erreur("open 2");
  if ((d3 = open(argv[1], O_RDWR)) == -1) erreur("open 3");
  buf[read(d1, buf, 2)] = '\0';
  printf("lu sur d1 : %s\n", buf);
  write(d2,"abc",3);
  buf[read(d3, buf, 2)] = '\0';
  printf("lu sur d3 : %s\n", buf);
  write(d3,"xxxx",4);

  if ((d1 = dup(d3)) == -1) erreur ("dup 1");
  if ((d2 = dup(d3)) == -1) erreur ("dup 2");
  buf[read(d1, buf, 2)] = '\0';
  printf("lu sur d1 : %s\n", buf);
  write(d2,"abc",3);
  buf[read(d3, buf, 2)] = '\0';
  printf("lu sur d3 : %s\n", buf);
  write(d3,"xxxx",4);

  exit(EXIT_SUCCESS);
}
```