Lille iut

Contrôle TP

Programmation Système
DUT Informatique
Le 17 mai 2017
2h

Consignes

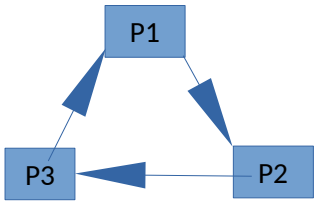
- Vous devez **impérativement** utiliser un Makefile pour la compilation et utiliser les options de compilation **-W -Wall -Werror**
- Lisez **attentivement** la totalité de l'énoncé avant de foncer tête baissée ;
- Une fois le TP terminé, vous devez rendre impérativement vos fichiers (votre Makefile et votre .c) selon le mode opératoire retenu par l'enseignant qui vous surveille ;
- Vos NOM, PRÉNOM et NOM DE MACHINE doivent figurer en commentaire sur la première ligne de chacun de vos fichiers (le caractère '*#*' permet de commencer un commentaire dans un Makefile) ;
- Vous penserez à tester les codes de retour des appels système ! 6.Commentez au maximum votre code.

Travail demandé

Le but du programme que vous allez écrire est de simuler le jeu de la patate chaude. Chaque joueur sera représenté par un processus et la patate par un entier. Le lancé de la patate d'un joueur à un autre sera représenté par l'envoi de l'entier représentant la patate dans un tube reliant un joueur à un autre. A chaque envoi, cet entier est décrémenté, s'il tombe à 0, la patate explose et le joueur a perdu, les autres gagnent.

Pour simplifier les choses, nous allons limiter le jeu à **trois** joueurs et chaque joueur lancera toujours la patate au même voisin.

Voici un schéma représentant la modélisation du jeu par des processus et des tubes. Les carrés représentent les processus (et donc les joueurs) et les flèches représentent les tubes.



Il y aura donc trois tubes. Un tube reliera le joueur 1 au joueur 2, un autre tube reliera le joueur 2 au joueur 3 et un dernier tube terminera la boucle en reliant le joueur 3 au joueur 1.

Exercice 1 : Initialisation de la patate

Afin de commencer le jeu, il nous faut une valeur aléatoire pour la patate. Plutôt que d'utiliser la fonction rand(), nous allons nous servir d'un fichier particulier du système, le fichier /dev/urandom.

Le fichier /dev/urandom est un fichier qui contient un nombre infini de valeurs aléatoires. Pour obtenir un nombre aléatoire, il suffit d'ouvrir ce fichier, de lire une valeur et de le refermer ensuite.

Écrire une fonction

```
int obtenir_valeur_aleatoire(int  borne_superieure);
```

qui lit un entier dans le fichier /dev/urandom et se sert de cette valeur pour retourner un entier entre 1 et borne_superieure.

Attention, l'entier que vous allez lire dans le fichier peut prendre une valeur négative !

Exercice 2 : Algorithme d'un joueur

L'algorithme de jeu pour un joueur est très simple :

1.Lire la valeur de la patate depuis une entrée ;

2.Décrémenter cette valeur ;

3.Si la valeur de la patate **est nulle**, quitter en affichant un message ;

4.Écrire la valeur de la patate dans une sortie ;

5.Recommencer en 1 tant qu'il y a des valeurs à lire depuis l'entrée.

Q 1. Écrire une fonction :

```
void lancer_patate(int out, int valeur);
```

qui effectue le lancement de la patate en écrivant valeur dans le descripteur out et qui affiche un message annonçant le lancement.

Q 2. Écrire une fonction :

```
int recevoir_patate(int in);
```

qui attends l'arrivée d'une patate en lisant un entier depuis le descripteur in et retourne sa valeur après avoir affiché un message. Si la fin de fichier est atteinte sur le descripteur in la fonction doit retourner -1.

Q 3. Écrire une fonction :

```
void demarrer_recepteur_patate(int in, int out);
```

qui implémente l'algorithme précédent à l'aide des fonctions lancer_patate et recevoir_patate. Si, après avoir décrémenté la valeur de la patate, cette valeur est 0, le programme doit quitter à l'aide de la fonction exit.

Exercice 3 : Programme principal

Vous allez maintenant écrire le main de votre programme qui lance le jeu. L'algorithme du main est le suivant :

- Obtenir une valeur de patate initiale de façon aléatoire ;
- Créer trois tubes ;
- Créer trois processus fils qui devront chacun appeller la fonction demarrer_recepteur_patate avec les bons paramètres (les bonnes entrées et sorties de tube) de façon à respecter le schéma précédent.
- Lancer la patate à un des processus fils (en appellant la fonction lancer_patate avec l'entrée d'un des tubes en paramètre)
- Attendre la fin des trois processus.

Voici un exemple d'affichage pour le programme que vous devez écrire :

```
selosse@acajou01:~$ ./patate
processus de pid 23678: Je lance la patate 4
processus de pid 23679: J'ai reçu la patate 4
processus de pid 23679: la patate a comme valeur 3, ca brule ! j'envoie au suivant
processus de pid 23679: Je lance la patate  3
processus de pid 23680: J'ai reçu la patate 3
processus de pid 23680: la patate a comme valeur 2, ca brule ! j'envoie au suivant
processus de pid 23680: Je lance la patate  2
processus de pid 23681: J'ai reçu la patate 2
processus de pid 23681: la patate a comme valeur 1, ca brule ! j'envoie au suivant
processus de pid 23681: Je lance la patate  1
processus de pid 23679: J'ai reçu la patate 1
processus de pid 23679: Explosion de patate, j'ai perdu
processus de pid 23680: Même pas mal, c'est gagné !
processus de pid 23681: Même pas mal, c'est gagné !
selosse@acajou01:~$
```

Quelques conseils

- Pensez à fermer les descripteurs des tubes que vous n'utilisez pas !
- Cet exercice peut être réalisé sans aucune allocation mémoire.
- Monsieur man est votre ami, ne l'oubliez pas.

Ne sortez pas de la salle avant de vous être assuré auprès de l'enseignant que le TP a été correctement rendu

---