

TP sur les threads : Jeu du Kems

Sujet du TP

Le but de ce TP est de programmer le jeu du Kems en gérant les joueurs comme des processus indépendants qui partagent la même pioche : le tapis central.

Dans ce TP le jeu de Kems se joue avec un jeu de 32 cartes. Le but du jeu est de réunir 4 cartes de valeur identique (4 rois, 4 neuf, etc...) Avant de commencer à jouer, on forme des paires de joueurs. On distribue 4 cartes à chaque joueur (les 4 cartes d'un joueur constituent le tapis de ce joueur), 4 autres sont étalées sur le tapis central.

Au signal, tout le monde peut échanger les cartes sur le tapis central avec celles de son propre tapis. Si les cartes du tapis central n'intéressent plus personne (plus aucun échange) alors la personne qui s'occupe du tapis central réinsère ces cartes dans le paquet et en remet 4 nouvelles, prises dans le paquet. Dès que quelqu'un a 4 cartes identiques, il doit faire signe à son partenaire, et c'est le partenaire qui doit annoncer : "Kems !" pour signifier que la paire de joueurs a gagné.

Installation

1. Récupérez l'archive `TP_Kems.tar.gz` soit
 - du répertoire `/info/tmp/AnnexesTPM1_177EN004/TP_Kems/`
 - à partir de ma page enseignement
`http://www-lium.univ-lemans.fr/~jacob/enseignement.html#SE`
rubriques Système d'Exploitation / Communication entre processus locaux / Synchronisation / TP
2. Décompressez la et désarchivez la par : `tar xvfz TP_Kems.tar.gz`
3. Compilez en faisant :
 - (a) `cd TP_Kems/`
 - (b) `make -f Makefile all`

Version 1

Dans cette version, chaque joueur joue pour lui-même (pas de partenaire) : quand un joueur a 4 cartes identiques alors il annoncera lui-même "Kems!". Dans cette version, il pourra donc y avoir jusqu'à 7 joueurs.

Le programme `kems.c` dans le répertoire `Etudiants` crée et fait jouer N joueurs séquentiellement (les uns après les autres) sur un tapis central. Les tapis des joueurs et le tapis central sont gérés par un seul processus.

Transformez le programme `kems.c` pour que

- le tapis central
- les tapis des joueurs

soient des processus indépendants qui jouent en même temps (en parallèle). Les joueurs et le tapis central appartenant tous au même utilisateur, utilisez un processus léger pour modéliser chaque tapis. Il faut donc alors

- interdire que deux joueurs retirent la même carte du tapis central en même temps
- pour le tapis central, savoir quand redistribuer les cartes. On peut envisager que si la variable booléenne `echange` reste à `FAUX` un certain temps, alors il faut redistribuer les cartes

Version 2

Réaliser la même chose avec le programme `kems_ncurses.c` qui affiche la partie sur un écran géré par `ncurses`. La difficulté de cette question est de gérer une ressource critique de plus : l'écran.

Version 3

Appliquez la vraie règle du jeu : avant que le jeu ne démarre, organisez des paires de joueurs et quand un joueur a un carré alors il fait signe à son partenaire qui, lui, annoncera "Kems!" pour signifier qu'ils ont gagnés. Dans cette version, il ne pourra donc y avoir que 4 ou 6 joueurs.

Réalisation

Le répertoire `Etudiant` contient un `Makefile` composé des règles pour compiler les programmes exécutables `kems` et `kems_ncurses` vous n'aurez qu'à faire

```
make -f Makefile all
```

pour compiler votre TP.

Rapport

Votre rapport devra contenir

- vos nom et prénom
- les codes sources modifiés de `kems.c` et `kems_ncurses.c`