

IN405 – Système d’exploitation

Projet – LV21

S. Gougeaud

2017/2018

1 Objectif

Le projet consiste en l’implémentation d’une simulation de table de blackjack ([http://fr.wikipedia.org/wiki/Blackjack_\(jeu\)](http://fr.wikipedia.org/wiki/Blackjack_(jeu))). Un ensemble de joueurs possèdent un nombre de jetons défini au début de la simulation, et tentent d’augmenter ce nombre en suivant une stratégie donnée, en battant la banque. Ces paramètres sont renseignés dans un fichier d’entrée. La simulation s’arrête si tous les joueurs ont atteint leur objectif ou sont en banqueroute (zéro jeton) ou si le nombre de mains donné en paramètre d’entrée du programme est atteint.

Chaque projet doit se faire en **binôme**. Le rendu du projet doit se faire avant le **25 Avril 2018 à 23h59** (heure d’été, France métropolitaine) par mail à l’adresse suivante, en fonction de votre chargé de TD :

hugo.bollore@uvsq.fr
mohammed-salah.ibnamar@uvsq.fr
mathieu.tribalat@uvsq.fr

Le mail doit contenir une archive tar.gz composée :

- des sources du projet faites en langage C ;
- des directives de compilation pour le projet (script, Makefile) ;
- d’un rapport de deux/trois pages décrivant brièvement votre implémentation ainsi que les problèmes que vous avez rencontrés.

Chaque binôme sera soumis à une soutenance de 10 à 15 minutes durant laquelle il devra présenter son code, faire une démonstration et répondre aux questions qui lui seront posées.

2 Pré-requis

- Gestion des fichiers.
- Gestion des processus/threads.
- Communication inter/intra processus.
- Gestion des mutex/conditions.

3 Fichier d’entrée du programme

Le programme prend en entrée un fichier constitué de deux parties :

1. l’installation de la table – donne le nombre de joueurs, le nombre de decks de cartes utilisés et le nombre maximum de mains jouées ;

1	#nbJoueurs ; nbDecks ; nbMains
2	3 ; 4 ; 20

2. la caractérisation des joueurs – donne le nombre de jetons de départ, le type de mise par main, la valeur stop et l’objectif à atteindre. Une ligne par joueur.

```

1 #nbJetons ; typeMise ; valStop ; objJetons
2 400;50;16;1000
3 750;10+;18;3000
4 1500;200-;17;3000

```

- Sachant qu’un joueur ne peut pas miser plus que le nombre de jetons qu’il possède, il existe 3 types de mise :
 - (a) mise constante (50) – le joueur misera toujours la même somme (donc 50) ;
 - (b) mise croissante (10+) – le joueur misera le double de la mise précédente s’il perd, et la mise de base s’il gagne (10, 20, 40, 80, 160, etc.) ;
 - (c) mise descendante (200-) – le joueur misera la moitié de la mise précédente s’il perd, et la mise de base s’il gagne (200, 100, 50, 25, 12, etc.).
- La valeur stop est la valeur minimale de la main du joueur pour laquelle il s’arrêtera de piocher.

4 Fichiers de sortie du programme

Le programme créera, pour chaque joueur, un fichier de journalisation qui dira pour chaque main jouée, les cartes qu’il a obtenu, celles de la banque, sa mise et son gain. Chaque ligne du fichier correspond à une main jouée :

```

1 #cartes ; totalJoueur ; banque ; totalBanque ; mise ; gain ; nbJetons
2 J68;24;98;17;50;0;950
3 7X4;21;KQ;20;50;100;1000
4 AK;21;X8;18;50;150;1100

```

- Les cartes sont représentées ainsi : 2, 3, ..., 9, X, J, Q, K, A.
- Le total est la valeur de la main (pour le calcul, cf. section 6).
- Le gain inclut la mise (pour le calcul, cf. section 6).

5 Consignes d’implémentation

Le programme doit être composé de deux groupes d’acteurs :

- la banque – lecture du fichier d’entrée, création des joueurs et exécution de la simulation ;
- les joueurs – communication avec la banque pour connaître les mains, et le gain/perte potentiel.

Le processus principal devra donc créer n joueurs (processus ou threads), puis prendre le rôle de la banque. Lui seul aura accès aux decks de cartes, et communiquera avec l’ensemble des joueurs pour leur indiquer quelles cartes ils ont piochées, connaître les mises effectuées, savoir s’ils veulent piocher d’autres cartes, puis leur indiquer leur gain s’il existe.

La gestion des communications entre les différents acteurs est libre (tubes, mémoire partagée, etc.).

La gestion du deck de cartes est fournie avec le projet, dans l’archive carddeck.tar.gz. Cette archive est composée d’un fichier source et d’un fichier d’en-tête et doit être utilisée dans votre programme. Le fichier d’en-tête contient les informations suffisantes à sa bonne exploitation.

L'exécution du programme doit donc générer un nombre de mains de blackjack défini, pour chaque joueur présent à la table. Les règles à respecter sont indiquées dans la section 6. Les deux dernières règles (doubler et séparer) sont optionnelles et peuvent rapporter des points bonus.

Les consignes suivantes doivent être respectées :

- l'appel au programme doit se faire comme suit :
`./lv21 fichier`
- les appels systèmes doivent être utilisés pour les fonctionnalités suivantes :
 - la gestion des fichiers d'entrée/sortie ;
 - la gestion des processus ;
 - la gestion des tubes.
- les constantes sont les seules variables globales autorisées ;
- chaque allocation doit être libérée avant la fin du programme ;
- chaque création de processus/thread doit être 'attendue'.

Rappel : Pour savoir si une fonction est un appel système ou non, il suffit de regarder le numéro d'indexage de cette dernière dans `man`. 2 signifie que c'est un appel système, 3 une fonction issue d'une bibliothèque.

6 Annexe – rappel rapide des règles du blackjack

1. Déroulement d'un tour de jeu : la banque distribue, carte par carte, deux cartes à chaque joueur et à elle-même. Puis elle demande à chaque joueur s'il souhaite piocher d'autres cartes ou non. Une fois que chaque joueur a terminé son tour, la banque complète sa main en s'arrêtant si elle atteint 17 ou plus. On dit que le joueur perd s'il dépasse 21 (quelque soit le résultat de la banque), fait égalité s'il possède la même valeur que la banque et gagne s'il dépasse la banque ou si la banque dépasse 21.
2. Valeur de la main : chaque carte chiffrée compte pour sa valeur (un 2 vaut deux points, un 3 en vaut trois, etc.), chaque figure (J, Q, K) compte pour dix points et chaque As compte pour onze points si le total de la main ne dépasse pas 21, et pour un point sinon. De plus, une valeur de 21 points obtenus en seulement deux cartes s'appelle un blackjack, et bat une valeur de 21 points obtenus en plus de cartes.
Quelques exemples :
 - 2-9-5 donne 16
 - J-8-8 donne 26
 - 7-2-A donne 20
 - 7-2-A-4 donne 14
 - Q-6-5 donne 21
 - Q-A donne blackjack
3. Gain : si le joueur perd, son gain est nul et il ne récupère donc pas sa mise ; s'il fait égalité, son gain est équivalent à sa mise, il la récupère ; s'il gagne, son gain est équivalent au double de sa mise. Si le joueur gagne en faisant blackjack, soit 21 en deux cartes, alors le gain est équivalent au triple de la mise.
4. Optionnel – doubler la mise : au moment de piocher une carte, un joueur a la possibilité de doubler sa mise. Il pioche alors une dernière carte et peut, s'il gagne face à la banque, doubler son gain initial.
5. Optionnel – séparer la main : si les deux premières cartes du joueur sont identiques (paire), il peut alors les séparer pour former deux mains au lieu d'une et doit mettre la même mise sur la deuxième main. S'il gagne les deux mains, alors il double son gain initial. S'il gagne l'une des deux mains, il récupère sa mise. S'il perd les deux mains, il perd l'ensemble de sa mise.