## TP à rendre 4

Vous devez écrire le programme roulette qui va simuler le jeu de hasard du même nom à la différence qu'ici un joueur ne pourra faire des paris que sur un numéro. La syntaxe du programme est la suivante :

```
./roulette <graine> <pari>
```

Les cases de la roulette seront représentées par des processus et la bille par un entier dont la valeur illustre la force avec laquelle la bille est lancée dans la roulette. Une roulette dispose de 37 cases (numérotées de 0 à 36), vous devrez donc créer 37 processus numérotés de 0 à 36. L'ensemble des processus forme un anneau unidirectionnel où chaque processus pourra communiquer avec son voisin (dans le sens horaire) via un tube anonyme comme illustré sur la figure1.

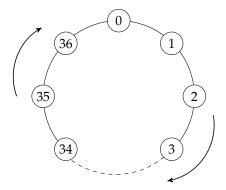


FIGURE 1 – Anneau de processus

Lorsqu'une bille entre dans la roulette, les processus se l'échangent dans le sens horaire via des tubes anonymes. Lorsqu'un processus reçoit une bille, il décrémente sa force et l'envoie au processus suivant. Lorsque la force arrive à 0, la rotation s'arrête et le processus courant doit se terminer avec un code de retour équivalent à son numéro de processus (entre 0 et 36).

Dans votre programme, le paramètre graine permet d'initialiser le générateur aléatoire via la fonction srand afin de pouvoir générer des nombres aléatoires via l'appel à la fonction rand. La force d'une bille devra être tirée aléatoirement dans l'intervalle [0;99]. Le paramètre <pari> correspond à un numéro (compris entre 0 et 36) sur lequel le joueur souhaite parier.

Le processus parent aura la charge de créer tous les processus, de générer la force de la bille et de la lancer dans la roulette via un tube anonyme sur le processus numéroté 0. Ce jet initial ne modifie pas la force de la bille. Ensuite, il doit attendre la terminaison de tous les processus. À la fin, le processus parent devra afficher la force initiale de la bille, le résultat (le numéro de processus sur lequel s'est arrêtée la bille) et la chaine SUCCESS si le résultat correspond au pari initial ou FAILED si le pari est perdu. Les affichages devront respecter le format illustré ci-dessous. En cas de victoire, le programme devra se terminer avec un code de retour nul, ou non nul en cas d'échec.

```
./roulette 145 34
FORCE = 88
RESULT = 14
FAILED
```

Vous écrirez le programme roulette en langage C en n'utilisant que des primitives systèmes; vous pouvez utiliser des fonctions de bibliothèque pour l'affichage et la conversion d'une chaîne de caractères vers un entier.

Vous devrez rendre sur Moodle un unique fichier roulette.c (Moodle sait qui vous êtes, il est inutile d'appeler votre programme Jean-Claude\_Dusse\_L2S4\_Printemps\_2019-2020\_roulette.c, et il est inutile aussi de rendre une archive au format-du-jour – il y aura de sévères pénalités sinon). Votre programme doit

compiler avec cc -Wall -Wextra -Werror -Wvla. Les programmes qui ne compilent pas avec cette commande ne seront pas examinés. Votre programme ne devra pas comporter d'allocation dynamique de mémoire.

Un script de notation est mis à votre disposition sur Moodle. Celui-ci compile votre programme et l'exécute sur des jeux de tests. Il vous affiche la note (sur 4 points) qui servira de base à l'évaluation de votre rendu. N'hésitez pas à lire ces tests et le fichier de log généré : ils peuvent vous aider à mettre votre programme au point.

Ce TP à rendre est individuel. On rappelle que la copie ou le plagiat sont sévèrement sanctionnés.