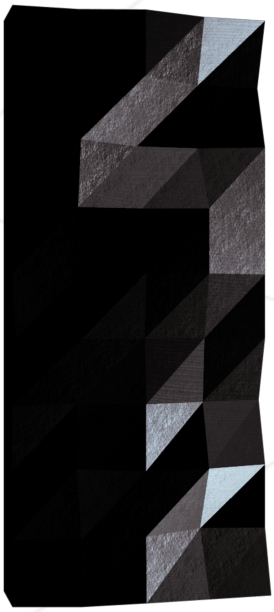

Projet UNIX relativement avancé

Lemipc

42 Staff pedago@staff.42.fr

Résumé: Ce projet ne consiste pas à recoder le lem-in avec des threads pour chaque fourmi.

Table des matières

Chapitre I

Préambule

If I didn't have you
Life would be blue
I'd be Dr. Who without the Tardis
A candle without a wick
A Watson without a Crick
I'd be one of my outfits without a Dick-ie
I'd be cheese without the mac
Jobs without the Wozniak
I'd be solving exponential equations that use bases not
found on your calculator making it much harder to crack
I'd be an atom without a bomb
A dot without the com
And I'd probably still live with my mom

And he'd probably still live with his mom

Ever since I met you
You turned my world around
You supported all my dreams and all my hopes
You're like Uranium 235 and I'm Uranium 238
Almost inseparable isotopes

I couldn't have imagined
How good my life would get
From the moment that I met you, Bernadette

If I didn't have you
Life would be dreary
I'd be string theory without any string
I'd be binary code without a one
A cathode-ray tube without an electron gun
I'd be "Firefly," "Buffy" and "Avengers" without Joss Whedon
I'd speak a lot more Klingon Heghlu'meH QaQ jajvam

And he'd definitely still live with his mom

Ever since I met you
You turned my world around
You're my best friend and my lover
We're like changing electric and magnetic fields
You can't have one without the other

I couldn't have imagined
How good my life would get
From the moment that I met you, Bernadette

Oh, we couldn't have imagined
How good our lives would get
From the moment that we met you, Bernadette

Kate & Riki

Chapitre II

Introduction

Vous avez vu précédemment la communication entre processus sur le réseau, via TCP/IP ; vous allez maintenant manipuler d'autres outils UNIX pour faire communiquer des processus en local. Vous allez découvrir ces outils via la création d'un client de jeu de plateau basique.

Chapitre III

Objectifs

Le projet a pour but de vous faire utiliser les communications inter-processus (IPC), de vous faire comprendre la notion de mémoire partagée, les sémaphores... Tout en créant un jeu !

Chapitre IV

Consignes générales

- Ce projet ne sera corrigé que par des humains. Vous êtes donc libres d'organiser et nommer vos fichiers comme vous le désirez, en respectant néanmoins les contraintes listées ici.
- L'exécutable devra se nommer `lemipc`
- Vous devez rendre un Makefile.
- Si vous êtes malin et que vous utilisez votre biliothèque `libft`, vous devez en copier les sources et le `Makefile` associé dans un dossier nommé `libft` qui devra être à la racine de votre dépôt de rendu. Votre `Makefile` devra compiler la librairie, en appelant son `Makefile`, puis compiler votre projet.
- Votre projet doit être en C et à la Norme. C'est la norminette qui fait foi.
- Vous devez gérer les erreurs de façon raisonnée. En aucun cas votre programme ne doit quitter de façon inattendue (Segmentation fault, etc...).
- Vous devez rendre, à la racine de votre dépôt de rendu, un fichier `auteur` contenant votre login suivi d'un '\n' :

```
$> cat -e auteur
xlogin$
$>
```

- Vous avez le droit d'utiliser toute la libc, exception faite de ce qui possède une correspondance dans votre libft (réputée complète). Vous devez alors utiliser la version de votre lib.
- Vous pouvez poser vos questions sur le forum ...
- Bon courage à tous !

Chapitre V

Partie obligatoire

Vous devez recoder la commande `lemipc` qui n'existe pas.

```
$> man lemipc
No manual entry for lemipc
```

Le but de ce projet est de faire communiquer et interagir des processus entre eux.

V.1 Règles du jeu

Le principe est de faire combattre des joueurs (rassemblés en équipes) sur un plateau en 2 dimensions.

- Une équipe gagne le jeu quand elle est la seule restante sur le plateau.
- Lorsqu'un joueur meurt, il disparaît du plateau.
- Pour tuer un joueur, il faut qu'au moins 2 joueurs d'une même équipe adverse soient à son contact, c'est-à-dire sur une case adjacente à la case où se situe le joueur que l'on veut tuer (y compris en diagonale).
- C'est le joueur qui doit vérifier s'il est entouré avant de bouger. Quand il se rend compte qu'il est entouré d'au moins 2 joueurs d'une même équipe adverse, il doit quitter le plateau et terminer son exécution.
- Une case du plateau ne peut accueillir qu'un seul joueur à la fois.

V.1.1 Déplacement des joueurs

Les joueurs se déplacent d'une case à la fois (une case par lock de la map), dans les directions : haut, bas, droite, gauche (pas de diagonale).
Les déplacements doivent être intelligents. Il faut évidemment poursuivre les joueurs des autres équipes, tout en essayant de se mettre d'accord sur une proie.

V.1.2 Taille de plateau

Le plateau peut faire la taille que vous souhaitez, mais doit pouvoir être changée facilement pour la soutenance. (ex : define)

V.1.3 Nombre de joueurs

Il n'y a pas de limite au niveau du nombre de joueurs, ni du nombre d'équipes.

V.2 Contraintes techniques

- Chaque client est un processus et il ne doit y avoir qu'un seul exécutable, ce qui implique que le premier joueur qui démarre crée les ressources partagées (shm, msgq, sémaphores).

- De la même façon, lorsqu'un joueur quitte le jeu, il doit vérifier s'il est le dernier sur le plateau, car dans ce cas il doit nettoyer tous les IPCs créés par le premier joueur pour éviter qu'ils restent en mémoire (man ipcs(1)).

- Le plateau doit être stocké dans un segment de mémoire partagé (SHM). Chaque joueur peut consulter le contenu du plateau comme il veut, mais pour le modifier il faut respecter les contraintes liées aux ressources partagées et aux accès concurrentiels (sémaphores).

- C'est le joueur qui doit vérifier s'il est entouré avant de bouger. Quand il se rend compte qu'il est entouré d'au moins 2 joueurs d'une même équipe adverse, il doit quitter le plateau et terminer son exécution.

- Un joueur ne peut communiquer avec les autres joueurs que par des MSGQ.

- Sur la map, on peut voir si une case est vide ou si elle contient un joueur, et dans ce cas c'est son numéro d'équipe que l'on voit, et on ne peut pas différencier les joueurs à l'intérieur d'une même équipe.

- Vous devez passer l'équipe en paramètre à l'exécutable.

V.3 Affichage

Vous devez faire un affichage de ce qui se passe sur le plateau :

- soit en mode texte, et dans ce cas :

  - soit seul le premier joueur (celui qui a créé le plateau) affiche le contenu du plateau

    - soit chaque joueur le fait
    - soit un processus spécifique s'en occupe
- soit en mode graphique, et dans ce cas :
    - soit seul le premier joueur (celui qui a créé le plateau) fait l'affichage
    - soit vous pouvez faire un autre executable dédié à l'affichage

Le client graphique ne doit en aucun cas influer sur le jeu. Il se contente d'afficher des informations. Pas de création de la map par le client graphique.

Chapitre VI

Partie bonus

En bonus, vous pouvez faire :

- Une interface graphiquement particulièrement jolie et/ou pratique.
- Des IA très performantes, de différents niveaux. On peut imaginer le passage du niveau de l'IA en paramètre.
- Tout autre bonus que vous jugerez utile et qui le serait aussi aux yeux de vos pairs pendant les soutenances.

Chapitre VII

Rendu et peer-évaluation

Rendez-votre travail sur votre dépot `GiT` comme d'habitude. Seul le travail présent sur votre dépot sera évalué en soutenance.