



# Lem-IPC

English version coming soon!

*Summary: Ce projet ne consiste pas à recoder le lemin avec des thread pour chaque fournis.*

# Contents

<b>I</b>	<b>Préambule</b>	<b>2</b>
<b>II</b>	<b>Sujet</b>	<b>4</b>
<b>III</b>	<b>Infos pratiques</b>	<b>6</b>

# Chapter 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

# Chapter II

## Sujet

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.

La FAQ contenue dans le fichier `faq.txt` fait partie intégrante du sujet.

Règles du jeu:

Le principe est de faire combattre des joueurs (rassemblés en équipes) sur un plateau en 2 dimensions. Pour qu'une équipe gagne le jeu, il faudra qu'elle soit 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 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). Il faut également que l'équipe de ces 2 joueurs (ou plus) soit différente de celle du joueur à tuer. Lorsqu'un joueur 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.

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`, `semaphores`). 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 (`semaphores`).

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 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 crée 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 fait l'affichage, soit vous pouvez faire un autre exécutable dédié à l'affichage.

# Chapter III

## Infos pratiques

- 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 bibliothèque `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 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 !