

# Devoir final INF3173

Jean Privat

du 15 décembre 2020 au 21 décembre 2020

- Chaque devoir est **individuel**.
- Le sujet est en Markdown, répondez directement dans le fichier. Codage UTF-8, fins de lignes Unix (`\n` seul).
- Répondez sur les lignes avec des chevrons (`>`), répondez après les chevrons et laissez les chevrons en place (pour que je vois vos réponses).
- Les limites maximales en nombre de mots des réponses seront interprétés avec intelligence.
- Ne touchez pas au reste du sujet. Un **diff** entre votre copie et le sujet ne **doit** faire apparaître que vos réponses.
- Les réponses devront être aussi **détaillées** et **argumentées** que possible (en respectant la taille de la réponse attendue).
- Vous avez jusqu'au lundi 21 décembre 2020 à 23h55 pour remettre votre fichier Markdown via Moodle. Vous pouvez faire autant de remises que vous le souhaitez, seule la dernière sera considérée.
- Merci de **ne pas discuter** du contenu du sujet sur Mattermost (ou via tout autre médium).
- Si vous avez des questions, vous pouvez toujours tenter de me les poser en message privé sur Mattermost, mais je me réserve le droit de **ne pas répondre** (ou de répondre après la date de remise).
- Tout non-respect du format ou des consignes sera pénalisé.

## Identification des étudiants

- Nom:
- Prénom:
- Code permanent:

## Quiz

La première partie du devoir est sous forme d'un quiz indépendant. Quand le quiz est commencé il doit être terminé en moins de 30 minutes.

- <https://www.moodle2.uqam.ca/coursv3/mod/quiz/view.php?id=20892>

04

## pipee.c

Un utilisateur débutant développe un programme `pipee.c` qui permet d'agrémenter l'exécution de commandes en les connectant automatiquement (via des tubes) au programme `lolcat` qui colore ainsi la sortie des commandes. Le nom de l'outil est sans doute un hommage à l'appel système `pipe` et à <http://caca.zoy.org/wiki/libcaca>.

```
#include <stdlib.h>
#include <unistd.h>
int main(int argc, char **argv) {
    int p[2];
    pipe(p);
    dup2(p[1], 1);
    system(argv[1]);
    dup2(2, 1);
    dup2(p[0], 0);
    system("lolcat");
    return EXIT_SUCCESS;
}
```

Malheureusement, le programme ne fonctionne pas très bien.

### Q1 ls

Si on exécute la commande « `pipee ls -l` » cela affiche bien la liste colorée des fichiers mais l'option `-l` semble n'avoir pas été prise en compte (la liste des fichiers est simple au lieu d'être détaillée) mais surtout le programme bloque après l'affichage (mais on peut forcer la terminaison avec Ctrl-C).

Sans juger de la qualité du programme ni indiquer ce qui aurait dû être programmé, expliquez et détaillez le comportement de l'exécution de cette commande, en particulier en expliquant pourquoi `-l` n'est pas pris en compte, pourquoi le programme bloque, et ce qui s'est passé quand on a fait Ctrl-C.

150 mots maximum.

### Q2 words

Si on exécute la commande « `pipee tac < /usr/share/dict/words` », le programme semble immédiatement se bloquer sans rien afficher. On fait un Ctrl-C et le programme débloque et commence à afficher joyeusement le contenu du fichier `words` en partant de la dernière ligne mais bloque à nouveau en plein milieu du fichier. Un second Ctrl-C termine alors la commande.

Même question mais en expliquant en particulier quelle est la cause de chacun des blocages, ce qui se passe à chacun des Ctrl-C, et pourquoi le fichier `words` n'est affiché qu'en partie.

150 mots maximum.

### Q3 Interblocage

Dans chacun des 3 cas de blocages précédemment identifiés, indiquez s'il s'agit d'un interblocage. Si oui, identifiez quels sont les éléments (processus/ressources/événements) qui forment l'attente circulaire. Si non, identifiez ce qui débloquera éventuellement la commande.

150 mots maximum.

### Q4 Critique

Jugez de la qualité du programme et, « en vous adressant » à ce programmeur débutant, expliquez les (nombreux) problèmes de son programme. Commencez chaque nouveau problème sur une nouvelle ligne.

150 mots maximum.

### Q5 Solution

Proposez une version simple, bien faite et fonctionnelle de `pipee.c` qui utilise correctement `pipe` et `dup2`. Note: vous n'êtes pas obligé d'utiliser la fonction `system` par contre.

```
// TODO
```