

Devoir final INF3173

Jean Privat

du 16 au 17 aout 2021

- Chaque devoir est **individuel**.
- Le sujet est en Markdown, répondez directement dans le fichier. Codage UTF-8, fins de lignes Unix.
- Répondez sur les lignes avec des chevrons (>), répondez après les chevrons ; vous pouvez en modifier le nombre, mais laissez les chevrons en place pour que je voie vos réponses.
- **Ne touchez pas** au reste du sujet. Un **diff** entre votre copie et le sujet ne **doit** faire apparaitre que vos réponses.
- Les réponses devront être aussi **détaillées** et **argumentées** que possible (en respectant la taille de la réponse attendue).
- Vous avez jusqu'à 23h55 du dernier jour 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.
- La qualité du français ainsi que la rigueur, la précision et la justesse dans les explications et dans le vocabulaire du cours utilisé seront prises en compte.

Identification des étudiants

- Nom:
- Prénom:
- Code permanent:

Note: Vérifiez que votre code permanent est correct.

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=2407308>

Programmation

Le méchant professeur donne un exercice aux étudiants:

Écrivez un programme C qui affiche chaque ligne de l'entrée standard à l'envers (sans toucher à l'ordre des lignes) et affiche un '*' tous les 10 octets affichés.

Ainsi, pour le fichier `exemple.txt` suivant

```
Hello, World!  
Bonjour  
le  
monde!
```

on veut obtenir la sortie suivante.

```
!dlroW ,ol*leH  
ruojno*B  
el  
!edno*m
```

Un étudiant distrait propose le programme `p.c`:

```
#include <stdlib.h>  
#include <stdio.h>  
#include <unistd.h>  
#include <sys/types.h>  
  
int main() {  
    int p[2];  
    pipe(p);  
  
    pid_t f = fork();  
    if (f == 0) {  
        dup2(p[1], 1);  
        system("rev");  
    }  
  
    close(p[1]);  
    dup2(p[0], 0);  
    char buf[10] = "";
```

```
        while(read(0, buf, 10) > 0) {
            write(1, buf, 10);
            write(1, "*", 1);
        }
    }
```

Première expérience

L'étudiant teste sur sa machine, la sortie est correcte jusqu'à un certain point puis le programme bloque et l'étudiant doit faire Ctrl-C.

```
$ ./p < exemple.txt
!dlroW ,ol*leH
ruojno*B
e1
!edno*m
e1
!edno*
```

Note: il est possible que sur votre machine vous n'observiez pas ce comportement, cf. seconde expérience.

Q1

Quels sont les processus créés ou manipulés par ce programme? Détaillez et justifiez votre réponse en indiquant, entre autres, pour chaque processus où dans le programme celui-ci est créé et quel est son processus parent.

Q2

Dans l'expérience, expliquez pourquoi le programme affiche deux fois « e1 !edno* » ? Justifiez votre réponse.

Q3

Dans l'expérience, indiquez pour chacun des descripteurs de fichiers de chacun des processus: à quoi correspond-il (quels fichiers ou tubes ou autre), quand est-il créé (ou rendu disponible au processus) et quand est-il fermé. Précisez également où ces événements ont lieu dans le programme.

Q4

Dans l'expérience, indiquez dans quel état (actif, prêt, bloqué, zombi, etc.) est chacun des processus. Détaillez et justifiez votre réponse en indiquant, entre

autres, ce que chaque processus est en train de faire (ou d'attendre) et à quel endroit du programme il est.

Q5

Peut-on caractériser (ou pas) la situation comme un interblocage, un « live lock », une famine, etc. ? Justifiez votre réponse.

Seconde expérience

L'étudiant, peu satisfait du comportement de son programme attribue le problème à sa machine, son compilateur ainsi qu'à la `libc`, et décide de compiler et exécuter son programme sur une machine non boguée.

Il se connecte sur le serveur de son université, recompile son programme et retente l'expérience, mais cette fois c'est pire: le programme affiche plein d'étoiles avec quelques caractères de temps en temps (ce qui remplit son écran) et bloque à nouveau.

```
$ ./p < exemple.txt
```

```
****l*****!*****W
*****o*****H*
*****e***** , *****
*****o*****o*****
****!*****r*****B*****d*****o*****
```

Q6

Expliquez ce qui s'est passé dans cette expérience. Indiquez, entre autres, pourquoi il y a plein d'étoiles, pourquoi toujours un blocage (est-ce le même blocage que dans l'expérience 1?).

Q7

Quelle est la cause selon vous de la différence de comportement entre les deux machines? Que pensez-vous de l'attitude de l'étudiant. Justifiez votre réponse.

Correction

Q8

En vous adressant à l'étudiant distrait, indiquez et expliquez les nombreux problèmes dans son code ainsi que ce qui aurait dû être fait.

Q9

En appliquant vos indications de la question précédente, proposez une version corrigée et fonctionnelle du programme `p.c` (contrainte: vous devez garder l'utilisation de tubes et l'appel à la commande `rev`).

TODO

Extra

Q10

Quelle est votre capsule préférée du cours INF3173. Expliquez pourquoi.