

# Travail pratique #1

IFT-2245

January 16, 2018

Dû le 10 Février à 23h59

## 1 Survol

Ce TP vise à vous familiariser avec la programmation système dans un système d'exploitation de style POSIX. Les étapes de ce travail sont les suivantes:

1. Parfaire sa connaissance de C et POSIX.
2. Lire et comprendre cette donnée. Cela prendra probablement une partie importante du temps total.
3. Lire, trouver, et comprendre les parties importantes du code fourni.
4. Compléter le code fourni.
5. Écrire un rapport. Il doit décrire **votre** expérience pendant les points précédents: problèmes rencontrés, surprises, choix que vous avez dû faire, options que vous avez sciemment rejetées, etc... Le rapport ne doit pas excéder 5 pages.

Ce travail est à faire en groupes de 2 étudiants. Le rapport, au format  $\LaTeX$  exclusivement et le code sont à remettre par remise électronique avant la date indiquée. Chaque jour de retard est -25%. Indiquez clairement votre nom au début de chaque fichier.

Si un étudiant préfère travailler seul, libre à lui, mais l'évaluation de son travail n'en tiendra pas compte. Si un étudiant ne trouve pas de partenaire, il doit me contacter au plus vite. Des groupes de 3 ou plus sont **exclus**.

## 2 CH: un shell

Vous allez devoir implanter une ligne de commande, similaire à `/bin/sh`, qui sait:

1. **Le Shell** - être capable d'exécuter des commandes  
\$ ./ch  
votre shell> echo bonjour  
Output: bonjour  
Autre commandes que nous allons tester: `cat`, `ls`, `man`, `tail`, ...
2. **Les Variables** - votre shell doit être capable de faire:  
votre shell> TEST=bonjour  
votre shell> echo \$TEST  
Output: bonjour
3. **Une boucle “For”** - votre shell doit être capable de faire:  
votre shell> for i in 1 2 3 ; do echo bonjour \$i ; echo allo \$1 ; done  
Output:  
bonjour 1  
allo 1  
bonjour 2  
allo 2  
bonjour 3  
allo 3
4. **Comprendre l'utilité du status de terminaison d'un processus avec `&&` et `||`**:  
votre shell> cat nofile && echo Le fichier existe.  
Output:  
cat: nofile: No such file or directory  
votre shell> cat nofile || echo Le fichier n'existe pas  
Output:  
cat: nofile: No such file or directory Le fichier n'existe pas

Le programme doit être exécutable sur `arcade.iro`. Cela ne vous empêche pas bien sûr de le développer sur un système différent, e.g. sous Windows avec Cygwin, mais assurez-vous que le résultat fonctionne *aussi* sur `arcade.iro`.

## 3 Cadeaux

Vous recevez en cadeau de bienvenue les fichiers `Makefile`, `rapport.tex`, et `ch.c` qui contiennent le squelette (vide) du code et du rapport que vous devez rendre.

### 3.1 Remise

Pour la remise, vous devez remettre deux fichiers (`ch.c` et `rapport.pdf`) par la page StudiUM du cours. Assurez-vous que tout fonctionne correctement sur `arcade.iro`.

## 4 Détails

- La note sera divisée comme suit: 20% pour tache 1 et 2, 40% pour tache 3, et 20% pour le rapport.
- Tout usage de matériel (code ou texte) emprunté à quelqu'un d'autre (trouvé sur le web, ...) doit être dûment mentionné, sans quoi cela sera considéré comme du plagiat.
- Le code ne doit en aucun cas dépasser 80 colonnes. indications supplémentaires.
- La note sera basée d'une part sur des tests automatiques, d'autre part sur la lecture du code, ainsi que sur le rapport. Le critère le plus important, et que votre code doit se comporter de manière correcte. Ensuite, vient la qualité du code: plus c'est simple, mieux c'est. S'il y a beaucoup de commentaires, c'est généralement un symptôme que le code n'est pas clair; mais bien sûr, sans commentaires le code (même simple) est souvent incompréhensible. L'efficacité de votre code est sans importance, sauf s'il utilise un algorithme vraiment particulièrement ridiculement inefficace.