

Rapport du Travail Pratique 1

(TP1)

Par

Charles Langlois et François Poitras

Rapport présenté à

M. Stefan Monnier

Dans le cadre du cours de

Systèmes d'exploitation (IFT2245)

26 janvier 2016

Université de Montréal

Table des matières

1	Fonctionnement du shell	3
1.1	Traitement de l'entrée	3
1.1.1	Commandes externes	3
1.1.2	Commandes internes	3
1.2	Gestion des erreurs	3
2	Problèmes rencontrés	3

1 Fonctionnement du shell

Le programme démarre en affichant le *path* vers le répertoire de travail courant, qui correspond au répertoire où le programme *ch* est situé. À partir de ce point, toutes les commandes disponibles qui sont définies dans la variable d'environnement *PATH* peuvent être exécutées, avec leurs arguments respectifs. Il est aussi possible de rediriger la sortie ou l'entrée d'une commande en utilisant le symbole '>' ou '<', comme dans la plupart des terminaux sous Linux. De la même façon, l'utilisateur peut utiliser des *pipes* avec le symbole '|'. Il est possible d'utiliser une astérisque pour faire référence à tous les fichiers normaux — c'est-à-dire tous les fichiers sauf '.' et '..' — dans le répertoire courant. Par exemple, la commande *echo ** affichera le nom de tous les fichiers du répertoire courant. Entre deux entrées, l'utilisateur peut quitter le programme en utilisant la commande *quit* ou encore en utilisant CTRL-D.

1.1 Traitement de l'entrée

Notre shell commence par éliminer tous les espaces au début de la commande, s'il y en a. Ensuite, on lit caractère par caractère jusqu'à ce qu'un caractère spécial soit atteint. Ces caractères sont le retour à la ligne, les opérateurs de redirection, de pipe, un espace ou la fin du fichier (*EOF*).

1.1.1 Commandes externes

aaaaaaaaaa

1.1.2 Commandes internes

aaaaaaaaaa

1.2 Gestion des erreurs

Les messages d'erreurs sont des citations inspirées de Margaret Thatcher. Même si cela peut paraître ridicule, les messages d'erreurs sont quand même descriptifs du problème qui s'est produit. On arrive sans problème à comprendre ce qui s'est passé. Par exemple, si on entre une commande qui n'existe pas, le shell nous répondra «There's no such thing as <command>. Only families and individuals.» où <command> sera le nom de la commande entrée.

2 Problèmes rencontrés

Puisque nous parlons de programmation en C, l'un des principaux problèmes est bien évidemment la gestion de la mémoire. Les *segmentation fault* furent nombreux au cours du développement et parfois subtils.