

# Rapport technique faroShell

## Résumé du projet :

Nous sommes 5 étudiants en INFO3 : Arthur Millet / Bastien Jean / Guillaume Potier / Sébastien Berlioux / Simon Bessenay.

Ce projet a pour but de créer un mini shell Linux en langage C. Pour cela on va coder les principales commandes du Shell et utiliser un interpréteur de commande de type bash.

Le readme.md est disponible sur Git pour plus de détails.

## Le MakeFile :

Explication de comment on l'a fait

## Mini-cours :

- Appels systèmes
- Structures utilisés

## Détails sur le Bash :

blabla

## Comment le faire marcher :

blabla

## Détails sur chaque commande usuelles :

Il faut pour chaque commande, exemple: ca sert à faire ça..... Et pour ca on a utilisé ça.....  
En expliquant les fonction principal et nos choix.

1. **mkdir** (option) nom du répertoire
2. **cd** chemin
3. **pwd**
4. **ls** (option) fichier
5. **cat** (option) fichier
6. **mv** (option) fichier destination
7. **cp** (option) fichier destination
8. **rm** (option) fichier
9. **du** (option) fichier

10. **chmod** (option) mode fichier
11. **chown** (option) propriétaire fichier
12. **chgrp** (option) groupe fichier
13. **echo** (option) message
14. **su**

### Les 3 formes :

1. Exécutables indépendants
2. Intégrés à l'exécutable interpréteur de commande
3. Sous forme de librairie dynamique à chargement explicite (L'interpréteur de commande charge l'ensemble des commandes (librairies) depuis un répertoire donné)

### Interpréteur de commande de type bash : [EXPLICATION]

Propose un prompt à l'utilisateur grâce à ...

A) Analyse la ligne de commande pour exécuter des fonctions

- Commandes intégrées (voir binôme #2)
- Programmes utilisateurs

B) Prise en compte des opérateurs de redirection :

- Redirection de flux : | ; < ; > ; << ; >>
- Opérateurs logiques : || ; &&
- Détachement du terminal : &

### Interface choisi :

Récrire en mieux (beaucoup mieux) ce qu'on avait déjà fait

### Bilan final (tout en restant dans la technique) :

Notre bash il est cool, on a réussi à faire ça....., pas réussi à faire ça ..... On apprend grâce au projet que ça.....

## Proposition:

1 page	MakeFile
1 page	Appels Systemes
1 page	Structures utilisés
3 pages	Les commandes soit 4-5 commande par pages (+les 3 formes)
1 page	Interpréteur de commande
1 page	Interface
1 Page	Bilan final

Total = 9 pages

## Reste à faire:

1. Finir toutes les fonctions (à optimiser avec des free / commentaires / indentations)
2. Résoudre le problème : **Erreur de segmentation (core dumped)**
3. **Faire valider / analyser le code (via cppchecker, frama-c, valgrind,...)**
4. Ajouter des options en plus
5. Mettre un message de sortie quand on quitte le bash
6. Finir le rapport technique (10 pages max)
7. Faire un Readme digne de ce nom (avec commande pour lancer)
8. Tout mettre en ligne sur Git avant vendredi 27/04 à 23h59 (pull du prof)
9. Réseau ?