

# RAPPORT TP1

Abdoulaye DIALLO, Theophyl Hoguet

## I . Bilan

Nous avons implémenté toutes les questions sauf la Partie VIII, normalement tout fonctionne, nous avons testé toutes les fonctions après les avoir implémenter. La partie VIII n'a pas été traitée par manque de temps. Mis à part la limitation indiquée dans la partie III, nous n'avons plus de bugs. Cependant il nous reste une fuite mémoire, on ne sait pas si c'est normal. Nous avons vérifié que chaque new avait bien un delete et que chaque malloc avait bien un free. De plus nous avons normalement bien mis les `#ifdef CHANGED` et `#endif`. Nous avons aussi un formateur automatique qui a changé beaucoup d'indentations et de mise en forme. Nous avons choisis de mettre le "copyStringFromMachine" et le "copyStringToMachine" dans le fichier consoleDriver.cc car il y avait un acces facile à l'objet "machine" et il est pour le moment seulement appelé dans les appels systèmes de consoleDriver.

## II . Points délicats

Au cours de ce projet, plusieurs points délicats ont été abordés. La transition du module principal (main) aux cas de test (test cases) a posé des défis. Les appels de fonctions pour les tests de méthodes ont parfois créé de la confusion. Pour résoudre ces problèmes, nous avons effectué plusieurs tests, modifié les entrées et analysé en profondeur le fichier makefile. Nous avons réussi à établir un lien entre le fichier start.S et les tests, permettant ainsi les appels système remontant jusqu'au noyau.

Un point particulièrement complexe était de comprendre comment un entier (int) pouvait représenter une adresse dans copyStringFromMachine. Nous avons réalisé que l'entier from représentait en réalité l'adresse dans la mémoire virtuelle, clarifiant ainsi ce concept.

Dans la partie V : "Le choix du fichier dans lequel mettre cette fonction est de votre ressort, il y a plusieurs endroits qui sont appropriés, à vous de motiver votre choix dans le rapport ! "

Nous avons decide de mettre copyStringFromMachine dans le fichier consoledriver.cc car

## III . Limitations

Notre implémentation présente une limitation majeure en ce qui concerne la gestion des chaînes de caractères longues. Nous avons défini une limite `MAX_STRING_SIZE` à 2048 caractères pour GetString. Ainsi, la fonction ne lit pas au-delà de cette limite. Une solution possible serait d'implémenter une boucle pour traiter tous les caractères au-delà de cette limite, bien que cela puisse poser des problèmes de gestion mémoire. Hormis ce problème de taille du buffer, notre implémentation de GetString fonctionne correctement.

## IV . Tests

Pour tester le bon fonctionnement de nos méthodes nous avons décidé d'écrire une fonction main comme dans le test putchar donne. Pour chacun de nos tests on appelle la méthode qu'on désire tester, puis on modifie les paramètres et la limite du buffer afin de détecter tout problème. Nos tests nous ont permis de modifier notre implémentation et de mieux comprendre le fonctionnement des codes donnés. Le seul grand problème que nous avons rencontré est détaillé dans la partie "Limitations" de ce rapport.

- Putchar: on test toutes les lettres de l'alphabet
- Getchar: testé dans ConsoleDriverTest, on attend que getChar obtienne un char et on l'affiche avec putChar entre '<' '>'
- PutString : rempli un tableau de caractère qu'on affiche
- GetString : rempli un tableau avec GetString qu'on obtient du terminal qu'on affiche avec PutString
- GetInt : on remplit un int et après on l'affiche avec PutInt
- PutInt : on affiche 42