

TACOS

Julien ALAIMO, Hugo FEYDEL, Olivier HUREAU

2019-2020

1 Fonctionnalités intéressante/importante

une (courte) partie présentant rapidement les fonctionnalités intéressante/importante de votre noyau (ce qui vous démarque de vos concurrents, ce qu'on peut faire avec votre logiciel, ...)

Un de nos points majeurs de développement a été la robustesse du noyau. En effet, nous avons essayé au maximum de prévoir quels seraient les erreurs utilisateurs possible afin de pouvoir relever des exceptions ou continuer le programme sans problème tout en vérifiant certaines mesures de sécurité (pas d'overflow)

2 Spécifications

une partie "spécifications" listant ce qui est disponible pour les programmes utilisateurs. Il faut mettre ici le genre d'information que vous trouvez dans les pages man. On doit donc trouver tous les appels systèmes implémentés avec leur prototype, la description des arguments, la description du fonctionnement (fonctionnalités utilisateurs, pas implémentation) de l'appel système, de la valeur de retour éventuelle, la signalisation des erreurs, ... Si vous avez également une bibliothèque utilisateur, vous devez décrire ses fonctions de la même manière que les appels systèmes.

2.1 Entrées/Sorties

- **void PutChar(char c);** Ecris le caractère "char c" sur la sortie standard
- **void PutString(char * string);** Ecris le chaîne de caractère "char * string" sur la sortie standard. La chaîne de caractère doit finir par '\0'. La taille maximal de la chaîne de caractère est de **JULIEN MODIFIE ICI STP**. L'appel système **PutString** est moins coûteux que plusieurs appels système **PutChar**

- **int GetChar();** Retourne la valeur ascii d'un caractère rentré dans l'entrée standard. La fonction attend qu'un caractère soit disponible. Attention il peut y avoir blocage.
- **void GetString(char * string, int taille);** Ecris la chaîne de caractère rentrée dans l'entrée standard à l'adresse de la chaîne de caractère en paramètre **char * string**. La taille de cette chaîne de caractère sera inférieure ou égale au paramètre **int taille**. **JULIEN MODIFIE ICI, c'est peut être faux.**
- **void GetInt(int * n);** Ecris l'entier rentrée dans l'entrée standard à l'adresse pointée par le paramètre **n**. L'entier peut être positif comme négatif. La valeur absolue de l'entier ne doit pas être supérieure à

$$abs(10^{10} - 1)$$

(valeur absolue). Sinon une erreur est levée.

- **void PutInt(int n);** Ecris l'entier en paramètre **int n** sur la sortie standard. L'entier peut être positif comme négatif. La valeur absolue de l'entier ne doit pas être supérieure à

$$10^{10} - 1$$

(valeur absolue). Sinon une erreur est levée.

3 Tests Utilisateurs

une partie "tests utilisateurs" décrivant les programmes de test que vous avez réalisés, ce qu'ils montrent, ...

4 Implémentation

une partie "implémentation" qui explique les points importants de votre implémentation. C'est donc la seule partie qui parle du détail du code que vous avez écrit. Expliquez vos choix d'implémentation.

5 Scolaire

une partie plus "scolaire" où vous décrivez l'organisation de votre travail (planning, ...), commentaires constructifs sur le déroulement du projet, ...