

NACHOS : Entrées/Sorties

---

**DEVOIR 1**

---

Guillaume CHARLET  
Kenji FONTAINE

19 novembre 2017

# Table des matières

<b>1</b>	<b>Description du projet</b>	<b>3</b>
<b>2</b>	<b>Bilan</b>	<b>3</b>
2.1	Partie 1 . . . . .	3
2.2	Partie 2 . . . . .	3
2.3	Partie 3 . . . . .	3
2.4	Partie 4 . . . . .	3
<b>3</b>	<b>Points délicats</b>	<b>4</b>
<b>4</b>	<b>Tests</b>	<b>4</b>
4.1	Partie 1 et 2 . . . . .	4

# 1 Description du projet

Ce travail a été réalisé dans un cadre universitaire, par des étudiants en Master 1 informatique à l'université de Bordeaux.

Un système d'exploitation est un ensemble de programmes permettant de diriger l'utilisation des ressources d'un ordinateur. Il assure la liaison entre l'utilisateur, les applications et le matériel. Nachos est un processus permettant d'émuler un système d'exploitation et du matériel.

L'objectif de ce devoir est de mettre en place sous Nachos un système d'entrées-sorties minimales, permettant d'exécuter de petits programmes. L'ensemble des modifications apportées sont entre balises `#ifdef CHANGED` et `#endif`.

## 2 Bilan

### 2.1 Partie 1

L'objectif de cette première partie est de permettre à l'utilisateur de créer et de manipuler des threads utilisateur en utilisant des appels systèmes qui permettront de manipuler ces threads (créer, lancer, terminer). Le thread ainsi créé partage le même espace mémoire que le thread père (AddrSpace). L'implémentation de ces fonctions est assez primitive et sera améliorée dans la partie 2. Cette première partie s'est relativement bien passée et ne nous a pas posé beaucoup de problème.

### 2.2 Partie 2

Cette partie a pour but d'améliorer les fonctions implémentées dans la partie 1. Nous devons permettre la création de plusieurs threads et faire en sorte qu'ils s'exécutent bien. La gestion de plusieurs threads en utilisant une BitMap nous a posé des difficultés et n'est pas fonctionnelle.

### 2.3 Partie 3

Nous n'avons pas eu le temps de traiter cette partie.

### 2.4 Partie 4

Même chose que pour la partie 3.

### 3 Points délicats

Nos problèmes viennent principalement de la partie 2. La gestion de plusieurs threads en utilisant la BitMap n'est pas fonctionnelle. En lançant la création de plusieurs threads dans le thread principal, nous n'arrivons qu'à lancer le thread principal ainsi que le premier thread de tous les threads que nous voulons créer.

### 4 Tests

Les fichiers de tests se trouvent dans le dossier `/code/test`. Ils sont exécutables en utilisant la commande suivante :

```
$ ~/nachos/code/userprog ./nachos -rs 1234 -x ../test/<fichier_test>
```

#### 4.1 Partie 1 et 2

Ces deux parties étant liées, il n'y a qu'un seul fichier de test. Le fichier est disponible dans le dossier `/code/test`. Il est exécutable en utilisant la commande ci-dessus. Nous observons que le thread principal printant "main" ainsi que seul le premier thread crée dans la boucle for sont lancés.