



POLYTECH' GRENOBLE

RICM 4ÈME ANNÉE

NachOS

Etape 3: Multithreading

Étudiants:
Elizabeth PAZ
Salem HARRACHE

Professeur:
Vania MARANGOZOVA

février 2012

1 Mise en place des threads *utilisateurs*

Avant de nous intéresser au Multithreading, nous allons tous d'abord nous intéresser au fonctionnement des threads sous NachOS.

1.1 Fonctionnement des threads sous NachOS

Un thread NachOS comporte une structure de données (ContextBlock) contenant : *status* : *JUST_CREATED*, *RUNNING*, *READY*
L'emplacement(bas)delapilestackTop : *SommetdelapilemachineState* : *Lecontenudel'ensemblederegistresuserRegisters* :
Lecontenudechaqueregistre(Threaduser)space : *L'espace'd'adressage(Threaduser)*

Un thread est tout d'abord créé avec la méthode *Thread()*. Elle initialise la pile et affecte *JUST_CREATED* à l'attribut *status*. C'est à dire que le thread est prêt à être exécuté.
[...je comprends pas trop ici]

La fonction *saveUserState* : Sert à sauvegarder le contexte d'exécution du programme lors d'une commutation. Les informations telles que l'état des registres ou encore le pointeur de pile sont sauvegardées dans le contextBlock. Ce contexte sera restauré lorsque le Thread est élu avec la méthode *restoreUserState*.

1.2 Description de l'installation d'un programme utilisateur dans l'environnement NachOS

Le lancement pas à pas permet de voir ce qui se passe lorsqu'on exécute un programme dans l'environnement Nachos.

Initializing address space, num pages 12, size 1536 Initializing code segment, at 0x0, size 384 Initializing data segment, at 0x180, size 32 Initializing stack register to 1520 Starting thread "main" at time 10

On peut voir dans le constructeur *AddrSpace*

2 Plusieurs threads par processus

3 Terminaison automatique