

# Projet Nachos

## M1 Informatique/MOSIG

Amine Aït-Mouloud  
Sébastien Avril  
Jean-Yves Bottraud  
El Hadji Malick Diagne

Janvier 2015

# Plan

Introduction

Implémentation

- Console

- Multithreading

- Gestion de la mémoire

- Système de fichiers

- Réseau

Gestion du projet

Conclusion

# Principales fonctionnalités

- ▶ Gestion d'entrées/sorties
- ▶ Multithreading
- ▶ Mémoire virtuelle et multi-processus
- ▶ Système de fichiers
- ▶ Transmission de données et de fichiers sur le réseau

# Plan

Introduction

Implémentation

- Console

- Multithreading

- Gestion de la mémoire

- Système de fichiers

- Réseau

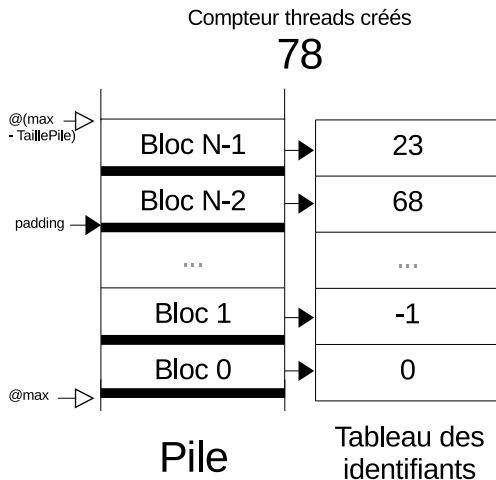
Gestion du projet

Conclusion

## Console: Lecture/Écriture

- ▶ Pas de choix d'implémentation particulier.
- ▶ Écriture et lecture bloqués par deux sémaphores différents.
- ▶ Problème de deux consoles s'exécutant en même temps.

# Multithreading: Identifiant et emplacement en pile



- Identifiant unique dans le processus.

## Multithreading: Attente

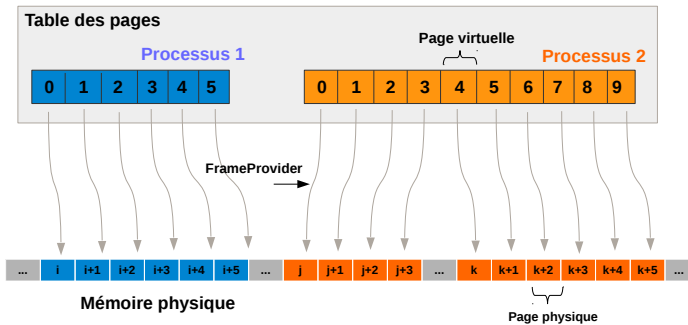
- ▶ Liste de threads en attente et qui ils attendent.
- ▶ Thread qui attend mis en pause et relancé à la fin du thread qu'il attend.
- ▶ Attendre si thread en cours d'exécution, retour direct sinon.
- ▶ Utilisation d'un compteur de thread créés et du tableau des identifiants pour connaître l'état d'exécution d'un thread.

# Gestion de la mémoire: Multi-processus

- ▶ Création d'un processus
  - ▶ StartUserProcess
- ▶ Arrêt automatique
  - ▶ Attente des threads appartenant à ce processus
  - ▶ Vérification du nombre de processus
- ▶ Relation entre processus
  - ▶ Compteur
  - ▶ Evolution possible



# Gestion de la mémoire: Mapping VPN-PPN



- ▶ Partage de la mémoire physique entre différents processus.
- ▶ Table de pages: mapping entre pages virtuelles du processus et pages physiques.
- ▶ Utilisation d'un FrameProvider pour obtenir les pages libres.

# Gestion de la mémoire: What else?

- ▶ Shell
  - ▶ Attente active
- ▶ Les problèmes
  - ▶ L'ordonnancement
  - ▶ bzero
- ▶ Les tests
  - ▶ Création de plusieurs processus simple
  - ▶ Création de plusieurs processus compliqué
  - ▶ Création d'un processus avec un executable erroné
  - ▶ Vérification des arrêts automatiques

## Système de fichiers: Mise en place

- ▶ 8 fichiers/dossiers maximum par dossier (+ "." et "..").
- ▶ Mise en place d'une hiérarchie de répertoire.
- ▶ Prise en compte du pathname.
- ▶ Dossier `"/System"` contient tous les exécutables (à l'image de `/usr/bin` pour Linux).

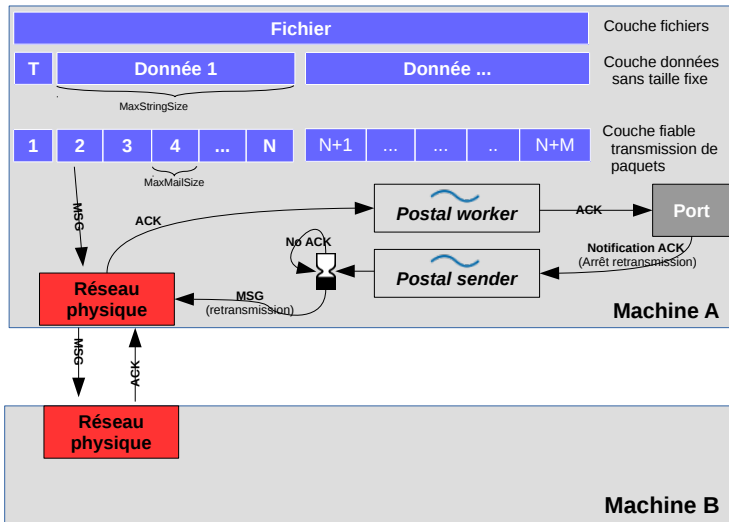
## Système de fichiers: Tests

- ▶ Appel système pour créer, supprimer et se déplacer entre les répertoires.
- ▶ Lancer des exécutables dans le système de Nachos.

# Système de fichiers: Problèmes et extensions

- ▶ Problème de création de processus.
- ▶ Table de fichiers ouverts.

# Réseau: Protocole utilisé



T=Taille du fichier ; 1...N...N+M = identifiant du paquet

## Réseau: API fournie

- ▶ Interface simple d'envoi et réception de fichiers et de données.
- ▶ L'utilisateur n'a pas à configurer des connexions entre deux machines.
- ▶ Envois synchrones avec attente d'acquittement pour chaque message.
- ▶ Retransmission périodique du message s'il n'est pas acquitté.

# Plan

Introduction

Implémentation

- Console

- Multithreading

- Gestion de la mémoire

- Système de fichiers

- Réseau

Gestion du projet

Conclusion



# Gestion du projet

- ▶ Utilisation de Git.
- ▶ Parallélisation des tâches.
- ▶ Travail par équipes de deux.
- ▶ Documentation tout au long du projet.

# Plan

Introduction

Implémentation

- Console

- Multithreading

- Gestion de la mémoire

- Système de fichiers

- Réseau

Gestion du projet

Conclusion

# Conclusion

## ► Commentaires :

- Nous a aidé à comprendre les fonctionnalités bas niveau du système.
- Nous avons mieux maîtrisé les partie 3 et 4.
- Manque de documentation sur l'architecture globale et les liens entre les fichiers.
- Projet intéressant et agréable.

# Conclusion

Fin.

Here's a potatoe

