

# Manipulation d'un Mini Système de Gestion de Fichiers

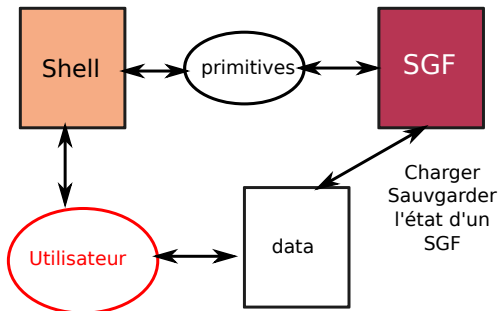
Projet système

IATIC3 ISTY

21 Avril 2015

# Émulateur de gestionnaire de fichiers avec un interpréteur bash

Créer un interpréteur de commandes shell  
Les commandes shell utilisent des primitives  
Les primitives interagissent avec le SGF



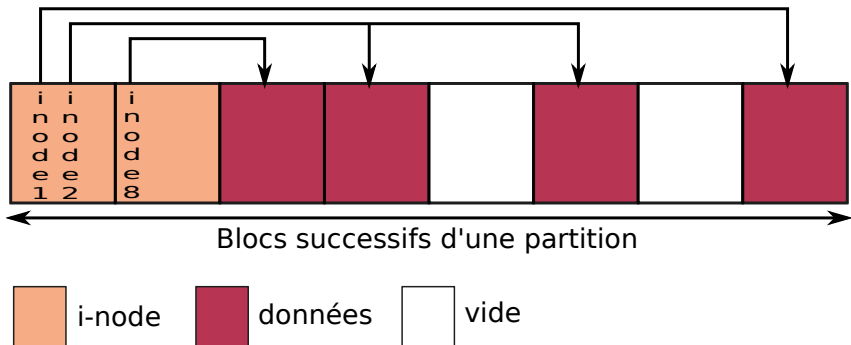
# Objectifs

- Reproduire la structure interne des inodes/blocs et l'arborescence des fichiers (faire le SGF)
- Définir un ensemble de primitives qui permettent d'interagir avec le SGF
- Définir un ensemble de commandes shell qui utilisent ces primitives
- Faire un interpréteur basique de ce mini bash
- Pouvoir sauvegarder et recharger l'état d'un SGF

# Plan

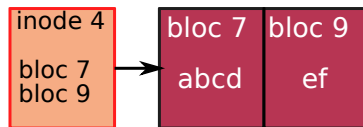
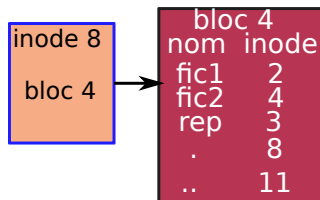
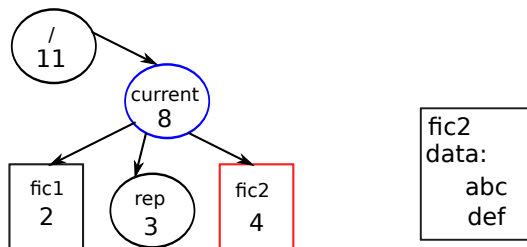
- 1 Système de fichiers
- 2 Primitives
- 3 Interpréteur shell
- 4 Réalisation du projet

# Système de fichiers

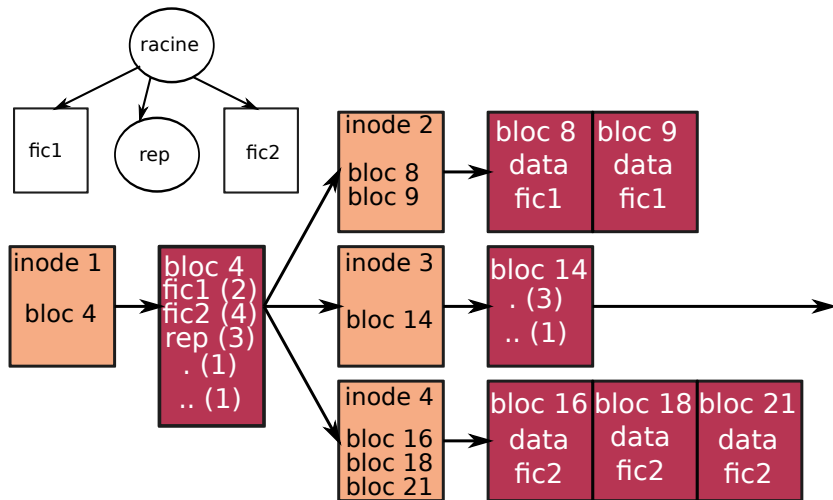


## Système de fichiers

# Fichiers et répertoires



# Structure arborescente



# Primitives des fichiers à définir

- `creat(nom,mode)`: crée un fichier et retourne son inode
- `open(nom,mode)`: ouvre/crée un fichier et retourne son inode
- `close(inode)`: ferme un fichier
- `read(inode,buffer,nombre)`: lit nombre octets mis dans buffer
- `write(inode,buffer,nombre)`: écrit nombre octets pris dans buffer
  - Gérer l'éventuelle allocation de nouveaux blocs pour les données



# Primitives des répertoires/fichiers à définir

- `mkdir(nom)`: créer un répertoire
- `rmdir(nom)`: supprime un répertoire vide
- `link(nom1,nom2)`: fait le lien entre une nouvelle entrée du répertoire et un fichier déjà existant
- `unlink(nom)`: efface une entrée du répertoire
  - Si aucune entrée sur un fichier, supprimer son inode et ses blocs de données

# Surcouche shell

- Commandes sur des fichiers : `cp rm mv cat ln 'echo "texte" > file'`
- Commandes sur des répertoires: `ls mkdir rmdir`
- Implémenter `cd` dans le shell
- `df`: informations sur le superbloc
  - Nombre de blocs disponibles
  - Nombre d'inodes disponibles
  - Taille en octets de l'espace libre sur le disque

# Exemples

- Exécuter dans le shell dans le répertoire R: `mv A B`
  - ① Appeler la primitive `link(A,B)`
    - ① Si B correspond à un fichier dans R, il faut le retirer de la table de R
    - ② Obtenir le numéro d'inode de A à partir de la table de R
    - ③ Créer une nouvelle entrée B/inode dans le répertoire R
    - ④ Supprimer l'entrée A dans la table de R
- Exécuter dans le shell `'echo "texte" > file'`
  - ① Appeler la primitive `open(file)`
    - ① Créer le fichier et on récupère l'inode
  - ② Appeler la primitive `write(inode,buff,nombre)`
    - ① Ajouter de nouveaux blocs avec le texte du fichier.
  - ③ Appeler la primitive `close(inode)`
    - ① Fermer le fichier

# Le projet

- Groupe de 4
- Coder le projet en C
- Implémenter le SGF, les primitives, le shell et la sauvegarde/recharge du SGF.
- Prévoir des tests de validation en utilisant la sauvegarde/recharge du SGF

# Gestion du code source

- Structurer le code
  - Séparer les primitives, les commandes shell, la sauvegarde/recharge et la structure du SGF dans des fichiers différents
  - Commenter les fonctions du projet
- Compiler le projet avec un Makefile
- Utiliser git pour gérer les différentes versions (Github)

# Le rapport

- Structure
  - Introduire le contexte
  - Pour chaque objectif
    - Expliquer les algorithmes utilisés
    - Expliquer les choix d'implémentation dans le projet
    - Montrez des tests, exemples de votre programme
  - Conclure
- Conseils
  - Utiliser des schémas
  - Passer un vérificateur d'orthographe
  - Faire des citations si besoin

# Modalités de rendu final

- Rendre une archive à l'adresse [mihail.popov@prism.uvsq.fr](mailto:mihail.popov@prism.uvsq.fr)
  - Readme.txt à la racine qui explique ce qui a été fait
  - Dossier projet avec les sources, les tests automatisés et un Makefile qui compile le projet
  - Rapport à la racine
- Pas de plagiat
- Pénalités pour non respect des consignes

# Déroulement du projet

- Rapport de mi-projet pour le lundi 18 mai 23h59
- Archive à rendre pour la semaine du 8 juin
- Soutenance avec slides pour la semaine du 8 juin



# Bonus

- Ajouter des méta données sur les inodes
  - Droits d'accès (gérer les fichiers en lecture/écriture seule)
  - Date de modification
- Gérer plusieurs utilisateurs dans le shell
- Ajouter une version réduite de find et grep dans le shell
- Étendre le mini shell

# Bibliographie

- Programmation linux en pratique chapitre 5 de A. Robbins
- Architecture de l'ordinateur de A. Tanenbaum chapitre 6.4.3.a
- Article Système de fichiers de Wikipedia