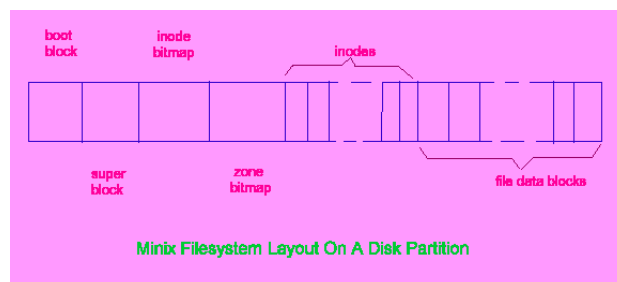


Rapport

Minix File system

ITI 2^{ème} soir
2014 / 2015



Sebastien Chassot, Andre-Luc Rrobyr
Juin 2015

Table des matières

1	Introduction	3
2	Le serveur de blocs	3

1 Introduction

2 Le serveur de blocs

la problématique de la déconnexion du client

Détecter la déconnexion d'un client n'est pas trivial. En effet, lors d'un *shutdown()*,

Solutions envisagées

Une des solutions les plus simple serait de traiter un échange à la fois ; le client se connecte, envoie une requête, reçoit la réponse et se déconnecte. Le server fait la même chose ; attend sur *accept()* qu'un client se connecte, attend une requête y répond et se déconnecte.

Cette solution est simple, permet de mieux maîtriser l'état du client et du server mais est couteuse en connexions.

Une autre solution est d'accepter un client et rentrer dans une boucle, un *fork()* ou un thread et traiter les requêtes du client tant qu'il ne se déconnecte pas. Cette solution est plus élégante mais elle nécessite de détecter la déconnexion du client.

Une alternative possible est d'ajouter au protocole une requête (un message) de déconnexion qui synchronise le client et le server.