

M2

Objectif (x86-64/Sparc)[Dimitri]

L'objectif de ce stage est double.

L'étudiant commencera par le portage de l'exo-noyau Hexo sur les processeurs x86 64 bits. Hexo et MutekH s'exécute déjà nativement sur l'architecture intel x86 32 bits classique des PCs.

Ce stage implique la prise en main du code du projet MutekH et plus particulièrement de l'hexo noyau Hexo et son portage x86. Le stagiaire devra développer les routines en assembleurs et en C des différents services: démarrage du système, handler d'interruption, changement de contexte, mémoire virtuelle, ... Ce code pourra s'inspirer du portage 32 bits existant. L'exécution du système sur une machine muni d'un processeur 64 bits permettra de valider cette partie du stage.

Une fois le portage sur processeurs x86-64 réalisé, l'étudiant aura assez de connaissance sur le fonctionnement de l'exo-noyau Hexo. Il pourra alors réaliser le portage sur processeur SPARC V8. Pour cette partie, la validation sera réalisée sur la plate forme de simulation SoCLib.

Objectif (NFS)[Nicolas et Joel]

L'objectif de ce stage est le développement d'un driver de système de fichier Network File System (NFS) pour la nouvelle couche de système de fichier virtuel de MutekH. Ce driver permettra à MutekH d'accéder aux fichiers au travers du réseau et viendra compléter la collection de systèmes de fichiers déjà supportés: Vfat, Iso9660 et RamFS.

Le stagiaire devra prendre en main le projet MutekH et particulièrement la pile réseau (libnetwork) et la gestion du système de fichiers (libvfs) pour développer le driver. Un serveur de fichiers NFS sera mis en place, l'accès aux fichiers depuis une application tournant sur MutekH permettra de valider le stage.

Objectif (Composant réseau SoCLib et le driver MutekH associé)?

L'objectif de ce stage est double. Tout d'abord il s'agit de développer un modèle de périphérique réseau pour la plate forme de prototypage virtuelle SoCLib. Ceci permettra à une simulation tournant sous SocLib? d'accéder au réseau de la machine hôte. Le modèle de simulation utilisera les mécanisme de DMA classiques qui sont déjà employé par d'autres composants. Il exploitera également la technologie TUN/TAP qui permet de créer facilement une interface réseau virtuelle sur la machine hôte afin d'immerger la simulation dans un vrai réseau.

Le second objectif est le développement du driver associé pour piloter l'interface réseau depuis MutekH qui contient déjà la pile réseau nécessaire et des pilotes pour des périphériques réseaux existants.

Le stage sera validé en exécutant une application sous MutekH capable d'échanger des paquets avec le réseau de la machine hôte.

Objectif (Libe2fs)[Nicolas et Joel]

Objectif (Coprocesseur Crypto pour SoCLib et le driver MutekH associé)[Alexandre et Geoffrey]

Objectif (Driver de MMU pour ARM9 et boot sur GP32)[Dimitri]