

INSA Lyon
4ème année
Spécialité Informatique

**Gestion des ressources physiques des systèmes
informatiques**

Projet

Kevin Marquet

Novembre 2013

1 Description du projet

L’objectif général de ce projet est d’illustrer les notions vues en cours associées à l’ordonnancement et aux systèmes temps-réel. Vous illustrerez ces notions à la fois au-dessus de Linux (la distribution *Raspbian* est présente sur les cartes SD initialement : pensez à sauvegarder le fichier `kernel.img` si vous l’écrasez) et dans votre mini-OS et dans les deux cas, on s’intéresse à l’exécution *sur Raspberry Pi*.

Sur chacun des systèmes, vous devrez tester différentes politiques d’ordonnancement permettant de mettre en évidence :

- le besoin de synchronisation entre processus ;
- le besoin d’ordonnancement à priorités ;
- l’impact de la politique d’ordonnancement sur les performances ;
- les différents critères de choix des algorithmes d’ordonnancement ;
- le fait que les processus sont bornés en temps ou en I/O.

Concrètement, vous devrez :

- Terminer votre mini-OS (aller jusqu’au bout du sujet précédent) ;
- Implémenter, dans votre mini-OS, différentes politiques d’ordonnancement différentes de round-robin, dont au moins une à priorités ;
- Implémenter un ensemble de processus fonctionnant au-dessus de votre mini-OS, permettant d’illustrer les points ci-dessus.
- Implémenter un ensemble de processus fonctionnant au-dessus de Linux, permettant d’illustrer les différences entre l’ordonnanceur `SCHED_OTHER` et les ordonnanceurs temps-réel `SCHED_RR` et `SCHED_FIFO`.
- Préparer exposé et démonstrations (voir plus bas).

2 Suggestions d’applications

Voici quelques exemples d’applications que vous pouvez utiliser ou implémenter pour illustrer les concepts vus en cours. Gardez à l’esprit que vous cherchez à illustrer ces concepts, et pas “juste” à programmer un jeu vidéo ou à jouer de la musique.

2.1 Les philosophes

Le programme donné dans le sujet d’implémentation du mini-OS est un bon exemple, bien que ce soit un exemple jouet, permettant de tester vos primitives de synchronisation. Si vous avez d’autres idées, on est preneur, c’est un peu répétitif sinon lors des évaluations ;)

2.2 Clignotage de la LED

Les fonctions `led_on()` et `led_off()` permettent d’allumer et éteindre la LED. Assurez-vous que votre mini-OS fonctionne avec deux processus, l’un éteignant la LED régulièrement, l’autre l’allumant. Sous Linux, vous pouvez utiliser <http://wiringpi.com/>.

2.3 Lecteur MIDI

On vous fournit en ligne une archive comprenant un lecteur midi fonctionnel pour votre mini-OS. Attention, ce code est pas (encore) tout à fait safe et certains hexanômes ont passé du

temps de debug là-dessus, parfois sans succès (sorry...); pour d'autres ça a marché facilement donc faites juste attention à pas y passer trop de temps.

2.4 Installer des logiciels sous Linux

Vous êtes libres d'installer les logiciels que vous voulez sous Linux (lecteur vidéo etc.). Également, on vous fournit une archive `pmidi.tgz` comprenant les sources d'un lecteur midi fonctionnant au dessus de Linux. Pour compiler, `make`. Pour exécuter, tapez `play_midi_file <filename>`. Attention, pour cela vous aurez besoin d'installer le paquet `timidity++`.

Pour installer des paquet, vous aurez besoin de connecter le Raspberry Pi à internet. Pour cela, servez-vous d'un portable comme passerelle. Ci-dessous, les commandes à taper pour une passerelle sous Linux (ce n'est pas garanti de marcher à tous les coups, utilisez vos cours de réseau) :

— Sur le Raspberry Pi :

```
sudo ifconfig eth0 192.168.0.2 netmask 255.255.255.0
sudo route add default gw 192.168.0.1
```

— Sur le PC :

```
sudo ifconfig eth0 192.168.0.1 netmask 255.255.255.0
echo 1 > /proc/sys/net/ipv4/ip_forward /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
/sbin/iptables -A FORWARD -i eth0 -o eth1 -m state --state RELATED,ESTABLISHED -j ACCEPT
/sbin/iptables -A FORWARD -i eth1 -o eth0 -j ACCEPT
```

Attention : pour jouer du son sous Linux et l'entendre, il vous faut indiquer à la couche ALSA que vous voulez que le son sorte par la sortie casque et pas HDMI :

```
sudo amixer cset numid=3 1.
```

2.5 Lecteur WAV

On vous fournit en ligne un lecteur de `.wav` (archive `wav_player.tgz`), fonctionnant uniquement au-dessus de Linux (désolé, pas eu le temps de coder le lecteur `.wav` pour le mini-OS, même si vous pouvez reprendre les couches basses du lecteur de fichiers MIDI). Attention, un décodeur lecteur wav, ça ne demande pas beaucoup de ressources, donc pas forcément utile pour vous.

2.6 Un clavier pour votre mini-OS

Le tutoriel <http://www.cl.cam.ac.uk/projects/raspberrypi/tutorials/os/input01.html> expose comment récupérer les appuis de touches par *polling*. Vous pourriez donc avoir un processus dont c'est le rôle.

2.7 Synthétiseur de son (ou autre action suite à l'appui sur une touche)

Dans votre mini-OS, vous pouvez jouer un son différent selon la touche du clavier (ou afficher/éteindre la LED). Sous Linux, c'est aussi possible : il existe plein de logiciel libre disponible.

2.8 Jeux : casse-briques, t  tris etc.

C'est pas compliqu  , si vous utilisez les bonnes librairies (en tous cas au-dessus de Linux...). Et parfait pour illustrer probl  mes de latence, de synchronisation (surtout si vous jouez de la musique en m  me temps), de performances... mais   a demande peut-  tre un peu de boulot. En m  me temps, vous   tes 6 :)

2.9 Lecture vid  o

Au-dessus de Linux, vous devez pouvoir vous en sortir. Au-dessus de votre mini-OS, c'est d  conseill   : on n'a pas pris le temps d'  crire ou r  cup  rer un driver vid  o (mais n'h  sitez pas :))

3 D  roulement et rendu

Le projet dure 5 s  ances. Une bonne partie de la derni  re s  ance sera consacr  e    l'  valuation. Cette   valuation prendra la forme d'un expos   oral pendant lequel vous pr  senteriez votre projet    l'aide de quelques slides et de d  monstrateurs. Bien que ces crit  res soient susceptibles d'  voluer, cet oral sera not   de la mani  re suivante :

- Mise en   vidence de la bonne synchronisation de processus + d  mo d'un probl  me r  solu : 3 points ;
- Mise en   vidence des probl  matiques d'ordonnancement (exemple : le lecteur de musique rame) dans votre mini-OS : 3 points ;
- Justification du choix de l'utilisation des politiques d'ordonnancement choisies : 2 points ;
- Analyse de performances, exp  riences et analyse : 2 points.
- Analyse th  orique de vos algorithmes : *waiting time*, *response time* etc. : 2 points
- Analyse exp  rimentale de vos algorithmes : latence observ  e etc. : 2 pts ;
- Mise en   vidence de probl  matiques d'ordonnancement sous Linux : 3 pts ;
- Mise en   vidence du compromis CPU / requ  tes I/O de vos applications : 2 points ;
- Clart   de la pr  sentation : 1 pt.