

Marc Feeley

---

Le TP1 a pour but de vous introduire au système d’exploitation minimal que vous devrez étendre de diverses façons dans les travaux pratiques. Nous utiliserons le nom “MINOS1” pour ce SE dans le cadre du TP1. Il faut écrire un “boot loader” ainsi qu’un pilote (“device driver”) pour le clavier et la souris. Une partie importante du TP1 est de lire les sources de MINOS1 pour comprendre son fonctionnement et le lien entre les différentes composantes.

Ce travail vous demandera de rechercher certaines informations par vous-même dans des livres (tel “The Indispensable PC Hardware Book”) et des documents sur le web. Citez correctement dans le code toutes vos sources d’information pour éviter de plagier. Les connaissances acquises dans ce travail seront testées à l’examen intra.

---

## 1 Survol de MINOS1

Les sources incomplets de MINOS1 vous sont fournis et il faut les modifier pour implanter les fonctionnalités précisées dans ce document. Pour obtenir une copie des sources que vous pouvez modifier, entrez la commande:

```
% tar xzf ~dift2245/tp1/minos1.tar.gz
```

Cela va créer le sous-répertoire `minos1`. Vous pouvez dès maintenant compiler MINOS1 en entrant la commande:

```
% make
```

dans le sous-répertoire `minos1`. Cela crée le fichier “`floppy`” qui est une image de floppy 1.44 MB qui contient le SE. Après toute modification des sources, assurez-vous de faire “`make`” pour créer le fichier “`floppy`” à nouveau.

Pour exécuter le système d’exploitation dans “`floppy`” en utilisant le simulateur d’IBM-PC “`bochs`”, il suffit d’entrer:

```
% ~dift2245/bochs/bochs -q
```

et puis taper `<return>` et entrer `c` (pour “continue”) dans `bochs`. Comme vous pouvez le voir le chargement de MINOS1 échoue, et l’une de vos tâches est de compléter le “boot loader”.

Pour exécuter le système d’exploitation sur un véritable IBM-PC, il faut transférer l’image “`floppy`” sur disque floppy, et l’insérer dans le lecteur de floppy au démarrage du PC. Sous Linux, vous pouvez transférer l’image “`floppy`” sur un disque floppy avec la commande:

```
% dd if=floppy of=/dev/floppy
```

Chaque endroit qui doit être modifié dans les sources est indiqué par un commentaire contenant quatre étoiles. Pour les retrouver facilement, entrez la commande:

```
% fgrep "****" *
```

## 2 Tâches à accomplir

### 2.1 “Boot loader”

Modifiez le fichier “bootsect.s” pour que le chargement en mémoire du kernel de MINOS1 se fasse correctement. Le kernel est stocké sur le floppy à partir du secteur “KERNEL\_SECTOR” (une constante symbolique). L’adresse mémoire où le kernel doit être chargé est donnée par la constante symbolique “KERNEL\_START” et la longueur du kernel en octets est donnée par la constante symbolique “KERNEL\_SIZE”.

Après avoir chargé le kernel, il faut éteindre le moteur du lecteur de floppy (cela peut se faire en envoyant 0 au port d’E/S 0x3f2), et puis brancher à l’adresse de chargement du kernel (utilisez, l’instruction “`ljmp $(KERNEL_START>>4), $0`”).

Après chaque secteur lu, affichez un “.” à l’écran pour indiquer que le chargement progresse. S’il y a une erreur de lecture branchez à l’étiquette “cannot\_load” pour signaler le problème.

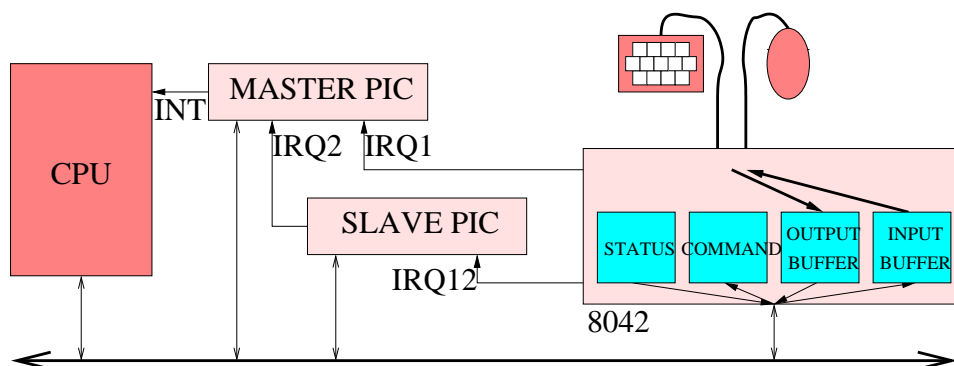
Pour faire la lecture d’un secteur, utilisez l’interruption logicielle 0x13 fonction 0x02 du BIOS. Vous pouvez supposer les paramètres de disque floppy 1.44 MB, c’est-à-dire: 512 octets par secteur, 18 secteurs par piste (“track”), et 2 têtes de lecture.

Consultez les liens sur la page web qui je l’espère seront une bonne source d’inspiration. Ne copiez pas le code des autres (que ce soit des sites web ou d’autres étudiants) car ce sera considéré comme un plagiat. Utilisez les liens sur la page web comme documentation pour vous guider vers votre propre solution.

\*\*\* LE PLAGIAT SERA SÉVÈREMENT PUNI. \*\*\*

### 2.2 Contrôle du clavier et de la souris par interruptions

Vous trouverez une description détaillée du fonctionnement du clavier et de la souris PS/2 dans le livre “The Indispensable PC Hardware Book” et sur le site web mentionné sur la page web du cours. J’en donne ici une description sommaire. Le diagramme suivant montre comment ces composantes sont connectées.



Le clavier et la souris PS/2 sont branchés au PC au moyen d’un lien sériel bidirectionnel (c’est-à-dire que ces périphériques peuvent transmettre des informations au PC et également recevoir des informations de configuration du PC). Le PC contient un micro-contrôleur, le 8042, qui s’occupe de l’échange d’information entre le CPU et ces deux périphériques. Ainsi, pour envoyer une commande à la souris, il faut envoyer au 8042 une “commande d’écriture de la souris” (en utilisant le registre de commande et le “input buffer”) et le 8042 va relayer le contenu du “input buffer” à la souris. C’est le même principe pour le clavier, mais avec une commande différente. Lorsqu’une information est transmise du clavier (ou souris) vers le PC, le 8042 tamponne l’information dans un registre interne (le “output buffer”), et informe le CPU (soit avec une interruption (IRQ1 pour le clavier, IRQ12 pour la souris) ou un bit dans le registre de statut) que cette

information est disponible pour être lue. Le CPU doit prendre garde de ne pas surcharger le 8042 qui est relativement lent. Avant d'envoyer une deuxième commande au 8042 il est bon de faire une attente d'une fraction de seconde.

Voici les ports d'E/S à utiliser pour communiquer avec le 8042 (notez que certains ports donnent accès à des registres différents en lecture et écriture et que ces registres contiennent chacun un octet):

- Registre "status" = port 0x64 en lecture seulement (le bit 0 indique si le "output buffer" contient une donnée (1=oui, 0=non), le bit 1 indique si le "input buffer" contient une donnée (1=oui, 0=non))
- Registre "command" = port 0x64 en écriture seulement
- Registre "output buffer" = port 0x60 en lecture seulement
- Registre "input buffer" = port 0x60 en écriture seulement

Certaines commandes prennent un octet en paramètre. Dans ce cas, après avoir envoyé une commande au port 0x64, il faut envoyer le paramètre au port 0x60. Les commandes du 8042 qui sont pertinentes pour ce travail sont:

- 0x60: commande pour configurer le 8042; cette commande utilise un paramètre pour indiquer le mode (les bits doivent être à zéro sauf le bit 0, qui doit être à 1 pour activer IRQ1 à la réception d'un "keycode" du clavier, et le bit 1, qui doit être à 1 pour activer IRQ12 à la réception d'un octet de la souris, et le bit 6, qui doit être à 1 pour que les "keycodes" soient convertis correctement)
- 0xa7: commande pour désactiver la souris; aucun paramètre
- 0xa8: commande pour activer la souris; aucun paramètre
- 0xad: commande pour désactiver le clavier; aucun paramètre
- 0xae: commande pour activer le clavier; aucun paramètre
- 0xd4: commande pour configurer la souris; le paramètre indique l'opération demandée à la souris:
  - 0xff: provoque un "reset" de la souris; après le reset la souris renvoie les codes suivants pour indiquer que l'opération s'est exécutée sans problèmes: 0xfa, 0xaa, et 0; la souris est ensuite placée dans un mode d'attente
  - 0xf4: sort la souris de son mode d'attente (à partir de ce moment elle se mettra à envoyer des paquets de 3 octets pour signaler les changements de position de la souris et la pression des boutons); la souris renvoie le code 0xfa pour indiquer que l'opération s'est exécutée sans problème

Dans le fichier "ps2.cpp" vous devez compléter l'implantation des fonctions `setup_ps2` (qui initialise le 8042, le clavier, et la souris), `process_keyboard_data` (qui est appelée par le traiteur d'interruption lorsqu'un octet est reçu du clavier) et `process_mouse_data` (qui est appelée par le traiteur d'interruption lorsqu'un octet est reçu de la souris).

Vous devez décoder les "keycodes" pour émuler un clavier AT standard et afficher les caractères obtenus sur la console (par exemple, si on presse les touches "shift" et "J" on obtient un "J" majuscule, et si on presse les touches "ctrl" et "J" on obtient une fin de ligne). Vous trouverez dans le fichier "ps2.cpp" une table qui facilite la transformation d'un "keycode" en code ASCII.

Pour la souris, il faut faire appel à la fonction `video::screen.move_mouse`, lorsque la souris est déplacée. Les déplacements du pointeur à l'écran doivent être cohérents avec les déplacements de la

souris. Je vous signale que dans bochs il faut cliquer le bouton de droite en même temps que la touche CTRL pour que bochs prenne le contrôle de la souris. De plus l'émulation de la souris dans bochs n'est pas très bonne (parfois la souris saute abruptement). Faites un test sur un vrai PC pour vous assurer que cela marche correctement.

Pour un bonus de 5%, implantez un mouvement de souris accéléré lorsque la souris est bougée rapidement. Pour de petits déplacements en X et Y (dx et dy) la fonction `video::screen.move_mouse` est appelée avec dx et dy inchangés. Pour des déplacements un peu plus grands on envoie  $2*dx$  et  $2*dy$  à la fonction `video::screen.move_mouse`. Et ainsi de suite pour des plus gros déplacements. Il ne faut pas utiliser un multiplicateur trop grand car cela risque de nuire au contrôle de la souris par l'utilisateur. Le comportement de la souris doit être le même quel que soit la vitesse du CPU.

### 3 Évaluation

Il n'y a pas de rapport à rédiger pour le TP1. Il suffit de faire la remise des fichiers à modifier ainsi que le fichier "floppy". Il faut aussi faire un test en présence du démonstrateur sur un vrai PC, pour montrer que votre système fonctionne correctement. Le démonstrateur vous expliquera la procédure à la période de démonstration.

- Pondération:
  - 40%: Implantation du "boot loader"
  - 30%: Implantation du clavier
  - 30%: Implantation de la souris
  - bonus 5%: Implantation du mouvement de souris accéléré
- L'élégance et la lisibilité du code, l'exactitude et la performance sont des critères d'évaluation.
- **Vous devez faire le travail par groupes de 2 personnes. Vous devez confirmer la composition de votre équipe (noms des coéquipiers) au démonstrateur au plus tard à la démonstration du 5 octobre. Si vous ne trouvez pas de partenaire d'ici quelques jours, venez me voir.** Indiquez vos noms dans le fichier "makefile" tel que demandé.
- Vous avez jusqu'à 23h59 le 25 octobre pour remettre vos fichiers. Vous serez pénalisés de 30% par jour de retard.
- Vous devez remettre vos programmes par remise électronique grâce à la commande suivante:

```
remise ift2245 tp1 floppy makefile bootsect.s ps2.cpp
```

Le contenu du fichier `floppy` que vous remettez doit correspondre aux fichiers que vous avez remis, et le contenu du disque que vous utilisez pour faire le test en présence du démonstrateur.