

PROJET UE INF4097 - Systèmes d'exploitation

Implémentation d'extensions du noyau xv6-riscv

Nom : SAKTA NZIA Pierrick Miguel

Matricule: 22Y1042

Introduction

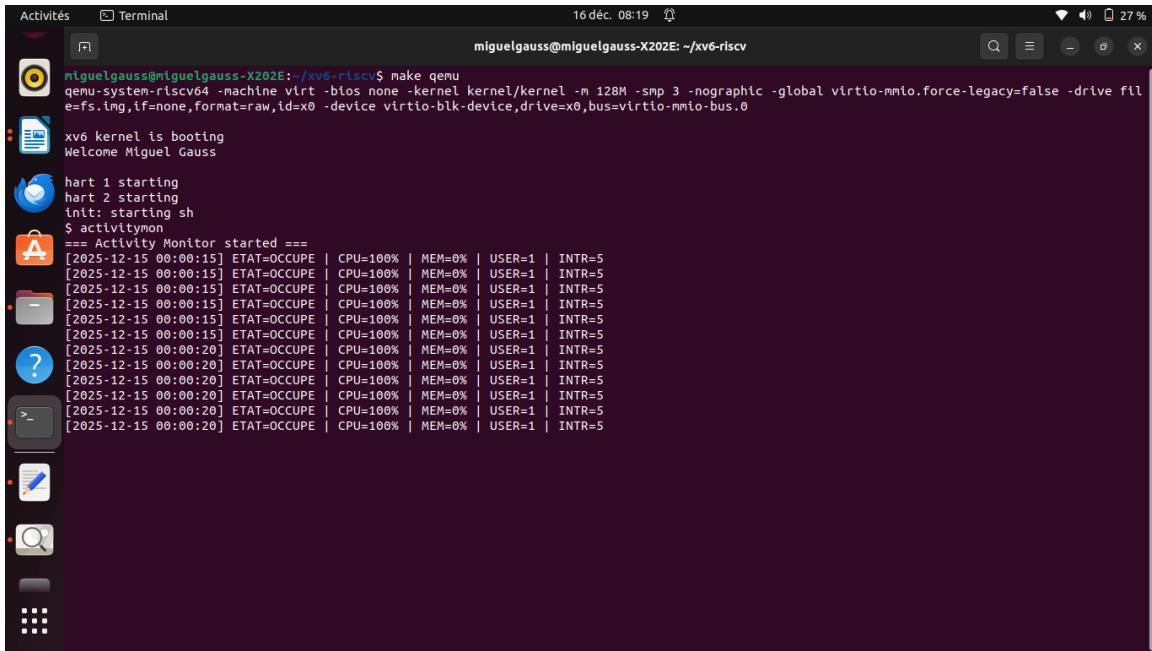
Ce projet a pour objectif de renforcer la compréhension des mécanismes internes d'un système d'exploitation à travers l'étude et la modification du noyau xv6-riscv. Les travaux réalisés portent sur la création d'un nouvel appel système, la mise en œuvre d'un démon de surveillance, la modification de l'ordonnanceur du système et l'implémentation d'une stratégie avancée de gestion de la mémoire.

Question 2 : Création d'un appel système et démon de surveillance

Cette question consistait à implémenter un nouvel appel système permettant de récupérer des informations sur l'activité du système. L'appel système **getactivity** a été ajouté au noyau xv6-riscv afin de fournir des informations telles que l'état du processeur, l'utilisation CPU et mémoire.

L'appel système a été implémenté dans le fichier **kernel/sysproc.c**, déclaré dans **kernel/syscall.h** et référencé dans **kernel/syscall.c**. Un stub assembleur a été ajouté dans **user/usys.S** afin de rendre l'appel accessible depuis l'espace utilisateur.

Un programme utilisateur nommé **activitymon** a ensuite été développé. Ce programme fonctionne comme un démon et interroge le noyau toutes les 5 secondes afin d'afficher périodiquement l'état du système.



```
miguelgauss@miguelgauss-X202E:~/xv6-riscv$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
: xv6 kernel is booting
Welcome Miguel Gauss
hart 1 starting
hart 2 starting
init: starting sh
$ activitymon
Activity Monitor started ===
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:15] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:20] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:20] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:20] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
[2025-12-15 00:00:20] ETAT=OCCUPE | CPU=100% | MEM=0% | USER=1 | INTR=5
```

Capture d'écran : Compilation et exécution de activitymon

Difficultés rencontrées (Question 2)

Plusieurs difficultés ont été rencontrées lors de l'implémentation de l'appel système. Tout d'abord, une erreur est survenue lors de l'utilisation de la fonction **argaddr()**, celle-ci retournant **void** dans **xv6-riscv**. Cette erreur a nécessité une adaptation du code. De plus, l'absence de la fonction **sleep** dans l'espace utilisateur a provoqué une erreur de linkage (*undefined reference to sleep*), corrigée par l'ajout du stub correspondant dans **usys.S**.

Ces difficultés ont permis de mieux comprendre la chaîne complète de fonctionnement d'un appel système dans **xv6**.

Question 3 : Modification de l'ordonnanceur

Dans cette question, il était demandé de modifier la politique d'ordonnancement du noyau. Un ordonnanceur expérimental de **type Lottery Scheduler** a été implémenté. Chaque processus se voit attribuer un nombre de tickets représentant sa probabilité d'obtenir le processeur.

À chaque décision d'ordonnancement, le scheduler calcule la somme des tickets des processus prêts, effectue un tirage aléatoire pondéré et sélectionne le processus gagnant. Cette approche permet d'introduire une notion de priorité tout en conservant une certaine équité.

```
miguelgauss@miguelgauss-X202E: ~/xv6-riscv
miguelgauss@miguelgauss-X202E:~/xv6-riscv$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
qemu-system-riscv64: -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0: Failed to get "write" lock
Is another process using the image [fs.img]?
make: *** [Makefile:179 : qemu] Erreur 1
miguelgauss@miguelgauss-X202E:~/xv6-riscv$ make qemu
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

xv6 kernel is booting
Welcome Miguel Gauss

hart 2 starting
hart 1 starting
init: starting sh

$ cpuburn 10
```

Capture d'écran : Test du scheduler avec processus CPU-bound

Difficultés rencontrées (Question 3)

Lors de la modification de l'ordonnanceur, une erreur critique de type kernel panic : **panic: release** est apparue au démarrage du système. Cette erreur était due à une mauvaise gestion des verrous (**spinlocks**) dans le scheduler. Le problème a été corrigé en respectant strictement la discipline d'acquisition et de libération des verrous autour de la fonction **swtch()**, conformément au modèle original de xv6-riscv.

```
miguelgauss@miguelgauss-X202E: ~/xv6-riscv
miguelgauss@miguelgauss-X202E:~/xv6-riscv$ make qemu
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_hello user/hello.o user/vlib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_hello > user/hello.asm
riscv64-unknown-elf-objdump -t user/_Hello | sed '1,/SYMBOL TABLE/d; s/.* //; /^$d' > user/hello.sym
riscv64-unknown-elf-gcc -Wall -Werror -Wno-unknown-attributes -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=medany -ffreestanding -fno-common -nostdlib -fno-builtins-trnrcp -fno-builtins-strlen -fno-builtins-memset -fno-builtins-memmove -fno-builtins-memcmp -fno-builtins-log -fno-builtins-bzero -fno-builtins-strchr -fno-builtins-exit -fno-builtins-malloc -fno-builtins-putc -fno-builtins-free -fno-builtins-memcpy -Wno-main -fno-builtins-printf -fno-builtins-fprintf -fno-builtins-vprintf -I. -fno-stack-protector -fno-pie -no-pie -c -o user/activitymon.o user/activitymon.c
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_activitymon user/activitymon.o user/vlib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_activitymon > user/activitymon.asm
riscv64-unknown-elf-objdump -t user/_activitymon | sed '1,/SYMBOL TABLE/d; s/.* //; /^$d' > user/activitymon.sym
riscv64-unknown-elf-gcc -Wall -Werror -Wno-unknown-attributes -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mmodel=medany -ffreestanding -fno-common -nostdlib -fno-builtins-trnrcp -fno-builtins-strlen -fno-builtins-memset -fno-builtins-memmove -fno-builtins-memcmp -fno-builtins-log -fno-builtins-bzero -fno-builtins-strchr -fno-builtins-exit -fno-builtins-malloc -fno-builtins-putc -fno-builtins-free -fno-builtins-memcpy -Wno-main -fno-builtins-printf -fno-builtins-fprintf -fno-builtins-vprintf -I. -fno-stack-protector -fno-pie -no-pie -c -o user/cpuburn.o user/cpuburn.c
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_cpuburn user/cpuburn.o user/vlib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_cpuburn > user/cpuburn.asm
riscv64-unknown-elf-objdump -t user/_cpuburn | sed '1,/SYMBOL TABLE/d; s/.* //; /^$d' > user/cpuburn.sym
mkfs/mkfs fs.img README user/.cat user/.echo user/.forktest user/.grep user/.lnt user/.kill user/.ls user/.mkdir user/.rm user/.sh user/.stressfs user/.usertests user/.grind user/.wc user/.zombie user/.logstress user/.orphan user/.hello user/.activitymon user/.cpuburn
nmeta 47 (boot, super, log blocks 31, inode blocks 13, bitmap blocks 1) blocks 1953 total 2000
balloc: first 1016 blocks have been allocated
balloc: write bitmap block at sector 46
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0

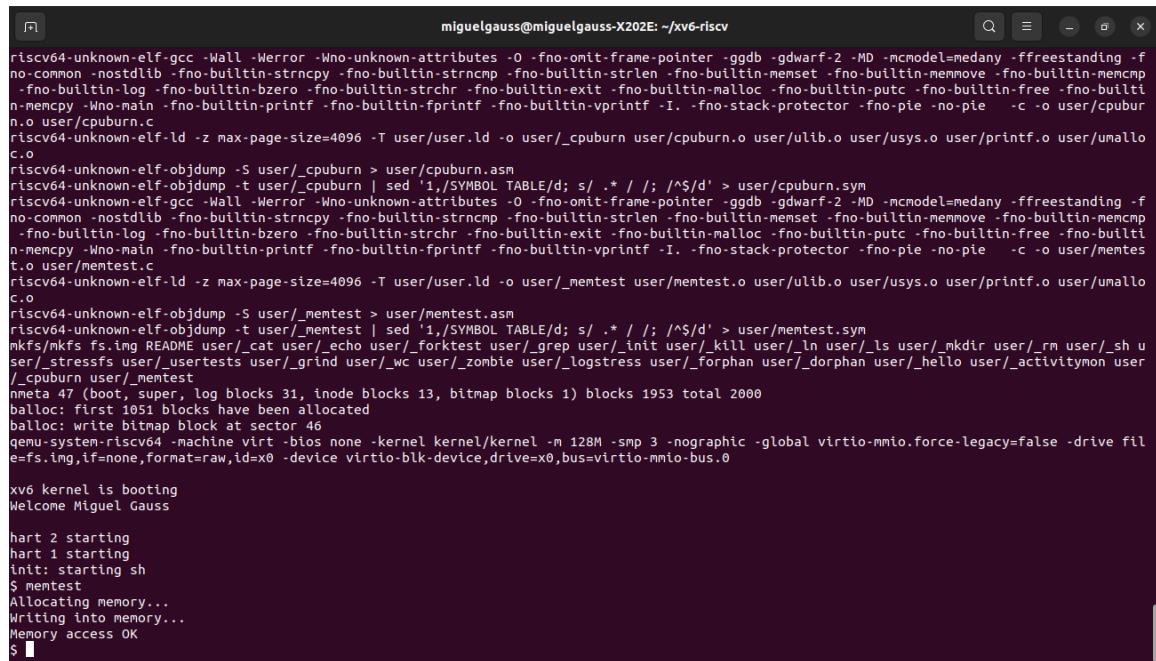
xv6 kernel is booting
Welcome Miguel Gauss

hart 2 starting
hart 1 starting
panic: release
```

Question 4 : Gestion mémoire avancée

Cette question portait sur l'implémentation d'une stratégie de gestion mémoire avancée. La politique de **Lazy Allocation** a été choisie afin de retarder l'allocation des pages physiques jusqu'au premier accès effectif.

La fonction **sbrk()** a été modifiée pour n'augmenter que la taille virtuelle du processus. Lorsqu'un accès à une page non mappée se produit, un défaut de page est généré. Ce défaut est intercepté dans la fonction **usertrap()**, qui alloue dynamiquement une page physique et la mappe dans l'espace d'adressage du processus.



```
miguelgauss@miguelgauss-X202E: ~/xv6-riscv
riscv64-unknown-elf-gcc -Wall -Werror -Wno-unknown-attributes -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmmodel=medany -ffreestanding -fno-common -nostdlib -fno-builtin-strncpy -fno-builtin-strncmp -fno-builtin-strlen -fno-builtin-memset -fno-builtin-memmove -fno-builtin-memcmp -fno-builtin-log -fno-builtin-bzero -fno-builtin-strchr -fno-builtin-exit -fno-builtin-malloc -fno-builtin-putc -fno-builtin-free -fno-builtin-nMemcpy -Wno-main -fno-builtin-printf -fno-builtin-fprintf -fno-builtin-vprintf -I. -fno-stack-protector -fno-pie -no-pie -c -o user/cpuburn.o
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_cpuburn user/cpuburn.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_cpuburn > user/cpuburn.asm
riscv64-unknown-elf-objdump -t user/_cpuburn | sed '1,/SYMBOL TABLE/d; s/.* / /; /^\$/d' > user/cpuburn.sym
riscv64-unknown-elf-gcc -Wall -Werror -Wno-unknown-attributes -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmmodel=medany -ffreestanding -fno-common -nostdlib -fno-builtin-strncpy -fno-builtin-strncmp -fno-builtin-strlen -fno-builtin-memset -fno-builtin-memmove -fno-builtin-memcmp -fno-builtin-log -fno-builtin-bzero -fno-builtin-strchr -fno-builtin-exit -fno-builtin-malloc -fno-builtin-putc -fno-builtin-free -fno-builtin-nMemcpy -Wno-main -fno-builtin-printf -fno-builtin-fprintf -fno-builtin-vprintf -I. -fno-stack-protector -fno-pie -no-pie -c -o user/memtest.o
riscv64-unknown-elf-ld -z max-page-size=4096 -T user/user.ld -o user/_memtest user/memtest.o user/ulib.o user/usys.o user/printf.o user/umalloc.o
riscv64-unknown-elf-objdump -S user/_memtest > user/memtest.asm
riscv64-unknown-elf-objdump -t user/_memtest | sed '1,/SYMBOL TABLE/d; s/.* / /; /^\$/d' > user/memtest.sym
mkfs/mkfs fs.img README user/.cat user/_echo user/_forktest user/_grep user/_init user/_kill user/_ln user/_ls user/_mkdir user/_rm user/_sh user/_stress user/_userstest user/_grind user/_wc user/_zombie user/_logstress user/_forphan user/_dorphan user/_hello user/_activitymon user/_cpuburn user/_memtest
nmeta 47 (boot, super, log blocks 31, inode blocks 13, bitmap blocks 1) blocks 1953 total 2000
balloc: first 1051 blocks have been allocated
balloc: write bitmap block at sector 46
qemu-system-riscv64 -machine virt -bios none -kernel kernel/kernel -m 128M -smp 3 -nographic -global virtio-mmio.force-legacy=false -drive file=fs.img,if=none,format=raw,id=x0 -device virtio-blk-device,drive=x0,bus=virtio-mmio-bus.0
xv6 kernel is booting
Welcome Miguel Gauss

hart 2 starting
hart 1 starting
init: starting sh
$ memtest
Allocating memory...
Writing into memory...
Memory access OK
$
```

Capture d'écran : Test de la lazy allocation avec memtest

Difficultés rencontrées (Question 4)

L'implémentation de la lazy allocation a entraîné plusieurs erreurs de compilation. En particulier, l'utilisation de return sans valeur dans la fonction **usertrap()**, qui retourne un uint64, a provoqué des erreurs bloquantes avec l'option -Werror. La correction a consisté à respecter le flot d'exécution original de xv6 et à laisser la fonction retourner uniquement la valeur satp à la fin.

```
miguelgauss@miguelgauss-X202E: ~/xv6-riscv
.o kernel/proc.c
riscv64-unknown-elf-gcc -g -c -o kernel/swtch.o kernel/swtch.S
riscv64-unknown-elf-gcc -g -c -o kernel/trampoline.o kernel/trampoline.S
riscv64-unknown-elf-gcc -Wall -Werror -Wno-unknown-attributes -O -fno-omit-frame-pointer -ggdb -gdwarf-2 -MD -mcmode=medany -ffreestanding -fno-common -nostdlib -fno-builtin-strncpy -fno-builtin-strncmp -fno-builtin-strlen -fno-builtin-memset -fno-builtin-memmove -fno-builtin-memcmp -fno-builtin-log -fno-builtin-bzero -fno-builtin-strchr -fno-builtin-exit -fno-builtin-malloc -fno-builtin-putc -fno-builtin-free -fno-builtin-memcpy -Wno-main -fno-builtin-printf -fno-builtin-fprintf -fno-builtin-vprintf -I. -fno-stack-protector -fno-pie -no-pie -c -o kernel/trap.o
.o kernel/trap.c
kernel/trap.c: In function 'usertrap':
kernel/trap.c:77:5: error: 'return' with no value, in function returning non-void [-Werror{return-type}]
  77 |     return;
     |     ^
kernel/trap.c:38:1: note: declared here
  38 | usertrap(void)
     | ^
kernel/trap.c:84:5: error: 'return' with no value, in function returning non-void [-Werror{return-type}]
  84 |     return;
     |     ^
kernel/trap.c:38:1: note: declared here
  38 | usertrap(void)
     | ^
kernel/trap.c:94:5: error: 'return' with no value, in function returning non-void [-Werror{return-type}]
  94 |     return;
     |     ^
kernel/trap.c:38:1: note: declared here
  38 | usertrap(void)
     | ^
kernel/trap.c:97:3: error: 'return' with no value, in function returning non-void [-Werror{return-type}]
  97 |     return; // fault traité avec succès
     |     ^
kernel/trap.c:38:1: note: declared here
  38 | usertrap(void)
     | ^
kernel/trap.c: At top level:
cc1: note: unrecognized command-line option '-Wno-unknown-attributes' may have been intended to silence earlier diagnostics
cc1: all warnings being treated as errors
make: *** [ : kernel/trap.o] Erreur 1
miguelgauss@miguelgauss-X202E:~/xv6-riscv$
```

Capture d'écran : Erreur rencontrée avec la fonction *usertrap()*,

Conclusion

Ce projet a permis d'approfondir la compréhension des mécanismes internes d'un système d'exploitation, notamment les appels système, l'ordonnancement des processus et la gestion de la mémoire. Les différentes difficultés rencontrées et corrigées ont constitué une expérience formatrice, rapprochant le fonctionnement de xv6 de celui des systèmes d'exploitation réels.