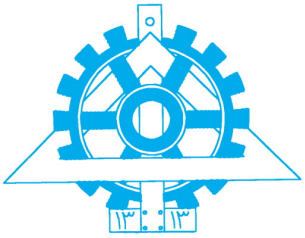

به‌نام‌خدا

سیستم‌عامل پیشرفته - بهار ۱۴۰۳

تمرین سوم: آشنایی با محیط سیستم‌عامل xv6

استاد درس: دکتر ناصر یزدانی

دستیار آموزشی: حسین افکار

ایمیل: hossein.afkar@ut.ac.ir

مقدمه

سیستم‌عامل xv6 توسط دانشگاه MIT برای دوره سیستم‌های عامل این دانشگاه طراحی شده است. در این تمرین ما سعی می‌کنیم با این سیستم‌عامل آشنا شویم. سپس در ادامه یک فراخوانی سیستمی را به این سیستم‌عامل اضافه می‌کنیم. جدیدترین نسخه این سیستم‌عامل برای معماری riscv نوشته شده است که این معماری نسبت به بقیه معماری‌ها مانند x86 و ARM ساده‌تر بوده و نشان‌گر پیشرفت‌های حوزه پردازنده‌ها می‌باشد. در این تمرین و تمرین‌های آتی کمترین درگیری با معماری پردازنده در نظر گرفته شده است ولی لازم است در مواقعی به راهنمای برنامه‌نویس این معماری رجوع کنید.
در ادامه و در تمرین آتی شما بیشتر با این سیستم‌عامل درگیر خواهید شد و این تمرین مقدمه‌ای برای این کار خواهد بود.

سوال ۱ (۴۰ نمره)

هدف این سوال این است که شما بتوانید این سیستم‌عامل را کامپایل و اجرا کنید. ابتدا لازم است که سورس این سیستم‌عامل را دریافت کنید. برای این کار از دستور زیر استفاده کنید.

```
1    git clone https://github.com/mit-pdos/xv6-riscv
```

دریافت سورس سیستم‌عامل xv6

سپس لازم است که پکیج‌های زیر را نصب کنید. این پکیج‌ها برای توزیع ubuntu می‌باشند. در صورت استفاده از توزیع دیگر می‌توانید معادل آن‌ها را به راحتی پیدا کنید.

```
1    apt install gcc-riscv64-unknown-elf qemu-system-riscv64 gdb-multiarch
```

پکیج‌های مورد نیاز

سپس می‌توانید این سیستم‌عامل را با دستور زیر در شبیه‌ساز qemu اجرا کنید.

```
1    make qemu
```

اجرای سیستم‌عامل

برای دیباگ کردن این کرنل نیاز دارید که دستور زیر را اجرا کنید.

```
1    make qemu-gdb
```

اجرای سیستم‌عامل در حالت دیباگ

سپس در یک پنجره جدا در مسیر سورس دستورات زیر را اجرا کنید.

```
1    gdb-multiarch -x .gdbinit
2    file kernel/kernel
```

تنظیمات دیباگر

در ادامه به سوالات زیر پاسخ دهید.

ملاحظات ارسال سوال ۱

- از اجرای تمامی مراحل بالا عکس و گزارش تهیه کنید.
- در خط ۱۶ فایل main.c با استفاده از gdb بریک‌پوینت بگذارید و رجیسترهای سیستم را گزارش کنید. مقدار ساختار pr که در خط ۲۴ فایل printf.c نیز تعریف شده است را نیز گزارش کنید. سپس با دستور step در gdb در سورس جلو بروید و تا زمانی که لاک به صورت کامل در اختیار فرآیند printf قرار می‌گیرد ادامه دهید. در این حین تغییرات ساختار pr را نیز مانیتور کنید. از مراحل تصویر تهیه کرده و آن را گزارش کنید.
توضیح دهید که چرا از spinlock در این بخش استفاده شده و آن را با spinlock معرفی شده در تمرین قبلی مقایسه کنید و عملکرد آن را کاملا توضیح دهید. سپس memory ordering را در سخت‌افزار riscv توضیح دهید و ذکر کنید که با معماری x86 چه تفاوت‌هایی دارد.
- ساختار حافظه که بخشی از آن در فایل memlayout.h تعریف شده است را توضیح دهید.
برای این کار به کتاب این سیستم‌عامل که در منابع این تمرین آمده است رجوع کنید. به نظر شما و با رجوع به ساختار حافظه برای رزرو کردن بازه‌ای از حافظه مجازی که با بقیه کارکردهای این سیستم‌عامل تداخل نداشته باشد باید چه کار کنیم؟
این بازه را نیز مشخص کنید.

سوال ۲ (۶۰ نمره)

در این سوال ما سعی می‌کنیم که یک فراخوان سیستمی جدید را به این سیستم‌عامل اضافه کنیم. این فراخوانی سیستمی با نام sys_poweroff به شبیه‌ساز qemu دستور خاموش کردن سیستم را می‌دهد.
سپس با استفاده از یک برنامه سطح کاربر به نام poweroff می‌توانیم از این فراخوان سیستمی استفاده کرده و سیستم را خاموش کنیم.
برای این کار نیاز است که در خانه 0x100000 حافظه فیزیکی مقدار 0x5555 نوشته شود که در ادامه در مورد آن بیشتر توضیح می‌دهیم.
برای این کار ابتدا لازم است به سیستم مدیریت صفحه مجازی سیستم‌عامل بفهمانیم که آدرس مجازی 0x100000 را به خانه‌ای از حافظه با همین آدرس نگاشت کند. برای این کار باید به تابع kvmmake در فایل vm.c رجوع کنید که در آن تعدادی از نگاشت‌های مستقیم حافظه وجود دارد کافی است که یک kvmmap جدید به این تابع اضافه کنید.
تابع kvmmap به جدول صفحه یا page table یک نگاشت اضافه می‌کند. کافی است برای هدف ما آدرس 0x100000 را به خودش نگاشت کنیم. لازم به ذکر است که سایز این صفحه باید با سایز صفحه سیستم‌عامل یکسان باشد. همچنین اجازه‌های PTE_R و PTE_W را باید به این نگاشت صفحه بدهید که در تابع kvmmap در نظر گرفته شده است.
سپس باید فراخوانی سیستمی sys_poweroff را به این سیستم‌عامل اضافه کنیم. برای این کار باید فایل‌های زیر را بررسی کنید و مشابه فراخوانی‌های سیستمی دیگر این کار را انجام دهید:

- syscall.h
- syscall.c
- user.h
- usys.pl

در این فراخوانی سیستمی لازم است که مقدار مورد نظر را در آدرس حافظه‌ای که نگاشت کردیم بنویسیم.
برای راحتی کد c زیر برای نوشتن مقدار در آدرس مورد نظر داده شده است.

```c
(*(volatile uint32 *) 0x100000) = 0x5555;
```

طریقه نوشتن مقدار در خانه حافظه با استفاده از پوینتر

سپس باید برنامه سطح کاربر poweroff را به این سیستم‌عامل اضافه کنید. برای این کار به Makefile رجوع کنید و متغیر UPROGS را تغییر دهید. سپس یک فایل به نام poweroff.c در پوشه user بسازید و از فراخوانی سیستمی که تعریف کرده‌اید در آن استفاده کنید.

ملاحظات ارسال سوال ۲

- در این بخش باید فراخوانی سیستمی sys_poweroff و برنامه سطح کاربر poweroff که از این فراخوانی استفاده می‌کند به این سیستم‌عامل اضافه شود. در خط دستور این سیستم‌عامل با نوشتن poweroff باید این برنامه فراخوانی شده و سیستم را خاموش کند.
- گزارش باید شامل توضیحات در مورد تمامی بخش‌های نوشته شده و کدهایی که به سیستم اضافه کرده‌اید باشد.
- کدهای نوشته شده حتما باید قابل کامپایل و اجرا کردن باشد.

ملاحظات ارسال

- گزارش سوالات باید عملکرد اسکریپت‌ها و برنامه‌های نوشته شده در تمرین را شرح دهد و همچنین تصاویری از عملکرد آن ارائه دهد. این کار حتما باید به صورت دقیق انجام شود. کیفیت توضیحات و تصاویر ارائه شده در گزارش از اهمیت زیادی در نمره‌دهی برخوردار است. تمامی سوالاتی که در بخش الزامات تحویل مطرح شده‌اند باید به دقت پاسخ داده شوند.
- گزارش باید به صورت lastname_studentnumber و با فرمت pdf باشد.
- فایل گزارش و سورس کدهای نوشته شده باید به فرمت فشرده در سامانه آپلود شود. نام فایل باید به صورت lastname_studentnumber باشد.
- مهلت ارسال تمرین ساعت ۲۳:۵۹ دقیقه روز ۲۸ اردیبهشت می‌باشد
- برای هر تمرین یک هفته مهلت ارسال با تاخیر در نظر گرفته می‌شود که فرمول محاسبه تاخیر به صورت دنباله فیبوناچی خواهد بود و هر تاخیر هر روز با روز قبل جمع خواهد شد. به عنوان مثال دو روز اول ۱ درصد تاخیر و روز دوم ۲ درصد تاخیر و روز سوم ۳ درصد تاخیر و روز چهارم ۵ درصد تاخیر دارد که به همین ترتیب ادامه پیدا می‌کند. در صورتی که چهار روز تاخیر داشته باشید. ۷ درصد جریمه تاخیر لحاظ می‌شود که اگر تا روز آخر که یک هفته است تاخیر وجود داشته باشد ۳۳ درصد جریمه تاخیر لحاظ می‌شود.
- ورژن سیستم‌عامل و توزیع لینوکسی و یا macOS مورد استفاده باید در گزارش درج شود.

موفق باشید.

References

[1] xv6 book. https://pdos.csail.mit.edu/6.1810/2023/xv6/book-riscv-rev3.pdf.

[2] GDB Quick Reference. https://users.ece.utexas.edu/~adnan/gdb-refcard.pdf.

[3] Debugging with GDB. https://ftp.gnu.org/old-gnu/Manuals/gdb/html_node/gdb_toc.html.

[4] RISC-V Instruction Manual. https://riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf.