



سیستم‌های عامل

تمرین سری اول

مهلت ارسال ۱ مهر ۱۳۹۳

نیم‌سال اول ۹۴-۱۳۹۳

مدرس: دکتر اسدی

دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

هدف از تمرین: آشنایی با هسته‌ی سیستم‌عامل و فراخوانی‌های سیستم

نحوه‌ی تحویل: پاسخ‌های تمرین خود را حداکثر تا ساعت ۲۳:۵۹ روز ۱ مهر ۱۳۹۳ به صورت فایل فشرده در فرمت ZIP و یا RAR بر روی [صفحه‌ی درس](#) در بخش مربوط به Assignment1 بارگذاری کنید. فایل فشرده شما باید شامل پاسخ سوالات تئوری به صورت تایپ شده و یا اسکن شده، کدهای مربوط به سوالات برنامه‌نویسی و گزارشی شامل توضیح کد (در سوالات برنامه‌نویسی) و نحوه‌ی اجرای آن باشد. توجه کنید که تمامی کدها و فایل‌های مورد نیاز اجرای آن برای هر سوال در پوشه‌ای مجزا با نام‌گذاری به صورت Q# (به طور مثال Q5) قرار گیرد.

نکات ضروری: تمامی بخش‌های تمرین و کدهای شما باید توسط خود شما نوشته شده باشد، در غیر این صورت، مطابق قوانین کلاس برخورد خواهد شد. در صورت دیرکرد در ارسال، حداکثر تا دو روز بعد از می‌توانید فایل تمرین خود را بر روی صفحه‌ی درس بارگذاری کنید.

در صورتی که هر پرسشی در رابطه با تمرین دارید، تنها از طریق [فروم درس](#) و در بخش مربوطه اقدام کنید.

سوال ۱

نصب و کامپایل هسته‌ی Linux

در این قسمت هدف کامپایل و نصب هسته‌ی سیستم‌عامل لینوکس است. بدین منظور شما باید در ابتدا سیستم‌عامل لینوکس را بر روی یک ماشین مجازی نصب کنید. برنامه‌ی مورد نیاز برای مجازی سازی و همچنین فایل iso سیستم‌عامل مورد نظر از طریق [صفحه‌ی درس](#) در اختیار شما قرار گرفته است.

زمان کامپایل هسته‌ی سیستم‌عامل بستگی به عوامل مختلفی دارد، شما در این تمرین باید سعی کنید این زمان را تا حد امکان کاهش دهید. نصب سیستم‌عامل در حالت کمینه و همچنین تنظیمات قبل از کامپایل در اینجا نقش بسزایی دارند. هسته‌ی سیستم‌عامل لینوکس شامل تنظیمات و برنامه‌های غیرضروری بسیاری (همانند درایورهای مختلف، پشتیبانی از فرمت‌های مختلف و ...) است که ما در طول این درس به آنها نیازی نخواهیم داشت. پس از نصب سیستم‌عامل و آماده سازی تنظیمات مورد نیاز برای کامپایل، شما باید شماره دانشجویی خود را به اسم هسته‌ی سیستم‌عامل اضافه کنید، نام هسته‌ی سیستم‌عامل شده شما باید به صورت زیر باشد:

3.15.3 – 91234567

بعد از ایجاد تغییرات در هسته شما باید آن را کامپایل کنید و سپس فایل تولیدی را نصب کنید. دستورات مورد نیاز برای نصب و کامپایل در اختیار شما قرار گرفته است.

توجه: قسمت بعد نیز نیاز به کامپایل هسته دارد، و می‌توانید بعد از انجام قسمت بعدی، کامپایل هسته را انجام دهید ولی این کار توصیه نمی‌شود. زیرا این امکان وجود دارد که تغییرات شما در تنظیمات پیش از کامپایل

دارای مشکل باشد، و با توجه به اینکه کدهایی که به هسته اضافه می‌کنید امکان بروز خطا در هنگام کامپایل را افزایش می‌دهد، باعث دشوارتر شدن عیب‌یابی شود!

اضافه کردن فراخوانی سیستم^۱ جدید

در این قسمت شما باید به هسته‌ی سیستم عامل لینوکس دو فراخوانی سیستم جدید اضافه کنید. به وسیله‌ی این دو فراخوانی سیستم، شما قادر خواهید بود مقدار متغیری که در هسته‌ی سیستم عامل تعریف می‌کنید تغییر دهید و یا دریافت کنید. بدین منظور باید متغیری جدیدی با نام `os_dummy_var` از نوع `int` تعریف کنید. همچنین باید دو فراخوانی سیستم با مشخصاتی که در آمده جدول ۱ است، ایجاد کنید. توجه کنید که فراخوانی‌های سیستمی شما نباید شماره‌ی متفاوتی داشته باشد.

جدول ۱ - مشخصات فراخوانی‌های سیستمی

Syscall Name	Syscall No.
<code>get_os_dummy_var</code>	357
<code>set_os_dummy_var</code>	358

توابع شما باید به صورتی که در ادامه آمده است تعریف شده باشد.

```
int get_os_dummy_var(int * value)
```

```
int set_os_dummy_var(int value)
```

همچنین توجه داشته باشید که در فراخوانی‌های سیستمی، مقداری که تابع تعریف شده بر می‌گرداند تنها باید نشان دهنده درست و یا غلط اجرا شدن آن تابع باشد و نباید برای انتقال داده بین سطح کاربر و سطح هسته استفاده شود. به طور مثال در این سوال در تابع `get_os_dummy_var()` شما باید مقدار متغیر `os_dummy_var` را از طریق آرگومان ورودی تابع (`value`) به سطح کاربر منتقل کنید و برای نشان دادن موفقیت و یا عدم موفقیت انجام عملیات از مقداری که تابع `return` می‌کند استفاده کنید. برای انتقال داده بین سطح هسته و کاربر باید از توابع `put_user` و `get_user` استفاده کنید.

برای تعریف فراخوانی‌های سیستمی جدید باید ابتدا و مشخصات فراخوانی‌ها را در این فایل، همانند سایر فراخوانی‌ها قرار دهید:

```
linux – 3.17.1/arch/x86/syscalls/syscall_32.tbl
```

سپس باید تعریف فراخوانی‌های جدید خود را در فایلی تعریف کنید، می‌توانید این کار را با ایجاد یک فایل جدید در پوشه‌ی زیر انجام دهید:

```
linux – 3.17.1/kernel/
```

¹ System Call

برای اینکه فایل اضافه شده در هنگام کامپایل کرنل شناسایی شود باید Makefile موجود در پوشه (که در بالا آمده است) را تغییر دهید و مشخصات فایل اضافه شده را باید به آن اضافه کنید. پس از اضافه کردن فراخوانی‌های سیستمی جدید و کامل کردن توابع مربوط به آن، شما باید هسته‌ی لینوکس را دوباره کامپایل و نصب کنید و سپس ماشین را با هسته‌ی جدید دوباره راه‌اندازی کنید.

برای بررسی صحت عملکرد فراخوانی‌های اضافه شده باید برنامه‌ای در سطح کاربر بنویسید، این برنامه باید هر دو تابع *set* و *get* را مورد ارزیابی قرار دهد. نمونه‌ای از برنامه سطح کاربر که فراخوانی سیستمی فرضی را صدا می‌کند در کد ۱ آمده است.

کد ۱ - نمونه فراخوانی سیستمی در سطح کاربر

```
#include <stdio.h>
#include <errno.h>

#define TEST_SYSCALL 357

int main(int argc, char ** argv) {
    printf("Calling...\n");
    syscall(TEST_SYSCALL, 1);
    return 0;
}
```

سوال ۲

سه ساختار سیستم عامل ریزهسته^۲، لایه‌ای^۳ و یکپارچه^۴ (ساده) را در نظر بگیرید و برای هرکدام از معیارهای طراحی زیر آن‌ها را با هم مقایسه کرده و دلایل خود را نیز شرح دهید.

- ا. قابلیت اطمینان
- ب. اشکال‌زدایی و اعتبارسنجی
- ج. کارایی
- د. نگهداری و توس
- ه. نحوه تعامل برنامه‌های سطح کاربر با خدمات سیستم عامل

^۲ Micro kernel

^۳ Layered

^۴ Monotonic