به نام خدا

آزمایشگاه سیستم عامل

آشنایی، اجرا و اشکال‌زدایی هسته سیستم عامل xv6

(بخش اول: آشنایی با xv6)

مقدمه

سیستم عامل xv6 یک سیستم عامل آموزشی است که در سال ۲۰۰۶ توسط محققان دانشگاه MIT به وجود آمده است. این سیستم عامل به زبان C و با استفاده از هسته Unix Version 6 نوشته شده و برروی معماری Intel x86 قابل اجرا می‌باشد. سیستم عامل xv6 علی‌رغم سادگی و حجم کم، نکات اساسی و مهم در طراحی سیستم عامل را دارا است و برای مقاصد آموزشی بسیار مفید می‌باشد. تا پیش از این، در درس سیستم عامل دانشگاه تهران از هسته سیستم عامل لینوکس استفاده می‌شد که پیچیدگی‌های زیادی دارد. در ترم پیشرو، دانشجویان آزمایشگاه سیستم عامل بایستی پروژه‌های مربوطه را برروی سیستم عامل xv6 اجرا و پیاده‌سازی نمایند. در این پروژه، ضمن آشنایی به معماری و برخی نکات پیاده‌سازی سیستم عامل، آن را اجرا و اشکال‌زدایی خواهیم کرد و همچنین برنامه‌ای در سطح کاربر خواهیم نوشت که برروی این سیستم عامل قابل اجرا باشد.

آشنایی با سیستم‌عامل xv6

کدهای مربوط به سیستم عامل xv6 از لینک زیر قابل دسترسی است:

https://github.com/mit-pdos/xv6-public

همچنین مستندات این سیستم عامل و فایلی شامل کدهای آن در صفحه درس بارگزاری شده است. برای این پروژه، نیاز است که فصل‌های ۰ و ۱ از مستندات فوق را مطالعه کرده و به سوالات زیر پاسخ دهید. پاسخ این سوالات را در قالب یک گزارش آپلود خواهید کرد.

- سوال ۱ : معماری سیستم عامل xv6 چیست؟ چه دلایلی در دفاع از نظر خود دارید؟
- سوال ۲ : یک پردازه[1] در سیستم‌عامل xv6 از چه بخش‌هایی تشکیل شده است؟ این سیستم‌عامل به طور کلی چگونه cpu را به پردازه‌های مختلف اختصاص می‌دهد.
- سوال ۳ : فراخوانی‌های سیستمی fork و exec چه عملی انجام می‌دهند؟ از نظر طراحی، ادغام نکردن این دو چه مزیتی دارد؟
- سوال ۴: مفهوم file descriptor در سیستم عامل‌های مبتنی بر Unix چیست؟ عملکرد pipe در سیستم‌عامل xv6 چگونه است و به طور معمول برای چه هدفی استفاده می‌شود؟

اجرا و اشکال‌زدایی

در این بخش به اجرای سیستم عامل xv6 خواهیم پرداخت. علی‌رغم این که این سیستم عامل قابل boot شدن مستقیم بر روی سیستم است، به دلیل آسیب‌پذیری بالا و رعایت مسائل ایمنی، از این کار اجتناب می‌کنیم و سیستم عامل را به کمک امولاتور qemu روی سیستم عامل لینوکس اجرا می‌کنیم. برای این منظور لازم است که کدهای مربوط به سیستم عامل را از لینک ارائه شده clone و یا دانلود کنیم. در ادامه با اجرای دستور make در دایرکتوری دانلود، سیستم عامل کامپایل می‌شود. در نهایت با اجرای دستور make qemu سیستم عامل برروی ماشین‌مجازی[2] اجرا میشود (توجه شود که فرض شده qemu از قبل برروی سیستم شما نصب بوده است. در غیر این صورت ابتدا آن را نصب نمایید).

[1] Process

[2] Emulator

اضافه کردن یک متن به Boot Message

در این بخش، شما باید نام اعضای گروه را پس از بوت شدن سیستم‌عامل روی ماشین مجازی qemu، در انتهای پیام‌های نمایش داده شده در کنسول نشان دهید.
تصویر این اطلاعات را در گزارش خود قرار دهید.

اضافه کردن چند قابلیت به کنسول xv6

پس از اجرای سیستم‌عامل بر روی qemu، مشاهده می‌کنید که در صورت استفاده از کلیدهای: shift +]، shift + [و Ctrl + c معادل کاراکتری آن‌ها در کنسول نمایش داده می‌شود. در این قسمت برای بهتر کردن کار با کنسول می‌خواهیم قابلیت‌هایی را به آن اضافه کنیم.

- در صورتی که کاربر دستور shift +] را وارد کرد باید نشانه‌گر[3] به ابتدای خط برود.
- در صورتی که کاربر دستور shift + [را وارد کرد باید نشانه‌گر به انتهای خط برود.
- در صورتی که کاربر دستور Ctrl + c را وارد کرد باید همه اطلاعات نوشته شده توسط کاربر پاک شوند.

توجه داشته باشید که در صورتی که نشانه‌گر در انتهای خط باشد استفاده از دستور shift + [نباید تغییری در جایگاه نشانه‌گر بدهد و همچنین در صورتی که نشانه‌گر در ابتدای خط باشد استفاده از دستور shift +] نباید جایگاه نشانه‌گر را تغییر دهد. در صورتی که در کنسول چیزی نوشته نشده باشد استفاده از این دستورات نباید عمل خاصی انجام دهند. در همه این قابلیت‌ها باید کاراکترها به خوبی شیفت پیدا کنند و در صورتی که نشانه‌گر در ابتدای خط بود در صورت نوشتن کاراکتری باید همه کاراکترها به خوبی شیفت پیدا کنند.

اجرا و پیاده‌سازی یک برنامه‌ی سطح کاربر

در این قسمت شما باید یک برنامه‌ی سطح کاربر و به زبان C بنویسید به برنامه‌های سطح کاربر سیستم‌عامل اضافه کنید. اسم این برنامه‌ی سطح کاربر cpt[4] می‌باشد. دو حالت کلی برای اجرای این برنامه وجود دارد.

زمانی که تنها یک ورودی اولیه به برنامه‌ی سطح کاربر داده شود، برنامه در هنگام اجرا منتظر خواندن یک خط از کاربر می‌ماند و در انتها آن خط را در فایل داده شده به عنوان ورودی می‌نویسد. توجه داشته باشید در صورتی که فایل گفته شده وجود نداشته باشد باید این فایل ساخته شود و اگر فایل موجود باشد متن روی آن بازنویسی می‌شود.

```
$ cpt first_file.txt
Hello World!
```

زمانی که دو ورودی به برنامه داده می‌شود باید محتوای فایل txt اول در فایل txt دوم عیناً نوشته شود. در صورتی که فایل اول موجود نباشد باید خطای مناسبی را نمایش دهید و در صورتی که فایل دوم موجود نبود باید فایل متناظر با آن را بسازید و اگر فایل موجود باشد متن روی آن بازنویسی می‌شود.

[3] Cursor
[4] Copy Text

```
$ cpt first_file.txt second_file.txt
$ cat second_file.txt
Hello World!
```

از فراخوانی‌های سیستمی open، read، write و close استفاده کنید که برای باز کردن، خواندن، نوشتن و بستن فایل‌ها استفاده می‌شود.

نکات مهم

- در نهایت کدهای خود را در کنار گزارش با فرمت pdf در یک فایل zip آپلود نمایید.
- به تمامی سوالاتی که در صورت پروژه از شما پرسیده شده‌است پاسخ دهید و آن‌ها را در گزارش‌کار خود بیاورید.
- همه‌ی اعضای گروه باید به پروژه‌ی آپلود شده توسط گروه خود مسلط باشند و لزوماً نمره‌ی افراد یک گروه با یکدیگر برابر نیست.
- در صورت مشاهده‌ی هرگونه مشابهت بین کدها یا گزارش دو گروه، نمره‌ی ۰ به هر دو گروه تعلق می‌گیرد.
- تمامی سؤالات را در **کوتاه‌ترین اندازه ممکن** پاسخ دهید.