



به نام خدا

آزمایشگاه سیستم عامل

آشنایی، اجرا و اشکال زدایی هسته سیستم عامل xv6

(بخش اول: سطح کاربر)



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 بر روی 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 بر روی 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 بر روی این سیستم عامل قابل اجرا باشد.

آشنایی با سیستم عامل 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.

- معماری سیستم عامل xv6 چیست؟ چه دلایلی در دفاع از نظر خود دارید؟
- مفهوم file descriptor در سیستم عامل‌های مبتنی بر Unix چیست؟ این مساله چه مزیتی برای طراح سیستم عامل دارد؟
- فراخوانی‌های سیستمی fork و exec چه عملی انجام می‌دهند؟ از نظر طراحی، ادغام نکردن این دو چه مزیتی دارد؟

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 بر روی امولاتور اجرا می‌شود (توجه شود که فرض شده qemu از قبل بر روی سیستم شما نصب بوده است. در غیر این صورت ابتدا آن را نصب نمایید).

اضافه کردن متن به Boot Message

در این بخش، شما بایستی با تغییرات مختصری در کد، کاری کنید که پس از بوت شدن سیستم عامل روی qemu، در پایان پیام‌هایی که روی کنسول نمایش داده می‌شود، نام اعضای گروه شما نیز نوشته شود. تصویری از این پیام را در گزارش خود قرار دهید.

حلقه حافظه در کنسول

پس از اجرای سیستم عامل بر روی qemu، مشاهده می‌کنیم که در صورت استفاده از کلیدهای ↑ و ↓ معادل ASCII این دو کاراکتر در کنسول نوشته می‌شوند. همانطور که از تجربه‌ی استفاده از ترمینال لینوکس می‌دانید، در آنجا استفاده از این دو کلید موجب می‌شود که دستوراتی که قبلاً در ترمینال وارد شده روی کنسول ظاهر شوند. در این قسمت از پروژه، باید این ویژگی را به کنسول xv6 اضافه کنید. برای این منظور کافی است تنها ۵ دستور وارد شده آخر را نگهداری کنید. همچنین دقت داشته باشید پیاده‌سازی شما به نحوی باشد که دستور اجرا شده دقیقاً همان دستوری باشد که بر روی کنسول نوشته شده است. پیاده‌سازی باید به نحوی باشد که استفاده از کلید ↑ موجب شود دستور وارد شده قبلی روی کنسول نمایش داده شود و استفاده از ↓ باعث شود که دستور بعدی در حلقه حافظه روی کنسول ظاهر شود.

اشکال‌زدایی

یکی از مسائل مهم در بررسی و تحلیل سیستم‌های عامل، اشکال‌زدایی است. یکی از ابزارهای مشهور برای این کار اشکال‌زدای gdb است. در این بخش از پروژه، باید پس از اجرای xv6، ابتدا آدرس مربوط به start_ را در هسته پیدا کنید. این قسمت مربوط به شروع اجرای هسته است. به کمک دستور `nm kernel | grep _start` می‌توانید آدرس شروع را پیدا کنید. سپس لازم است که به کمک دستور `make qemu-gdb` سیستم عامل را در وضعیت اشکال‌زدایی اجرا کنید. پس از اجرا در وضعیت اشکال‌زدایی، به کمک دستورات gdb باید یک breakpoint در آدرس موردنظر ایجاد کرده و اجرا را تا ابتدای آن دستور ادامه دهید. سپس گزارشی از وضعیت رجیسترها در آن مرحله ارائه دهید. همچنین در گزارش خود توضیح دهید که در معماری x86 رجیسترهای esp و ebp نشانگر چه چیزی هستند؟ در شرایط فعلی مقدار آن‌ها چیست و چه نتیجه‌ای از مقدار کنونی آن‌ها می‌گیرید؟

اجرای یک برنامه سطح کاربر

بخش پایانی این پروژه به اجرای یک برنامه در سطح کاربر می‌پردازد. برای این کار، شما بایستی یک برنامه در سطح کاربر و به زبان C بنویسید که ۷ عدد را از ورودی دریافت کرده و پس از به دست آوردن میانه آن‌ها، خروجی را در فایل به اسم `result.txt` که در سیستم ایجاد می‌کنید، بنویسد. توجه شود که الگوریتم بدست آوردن میانه اهمیتی ندارد ولی فایل گفته شده باید توسط کد شما به وجود آید. برای ایجاد فایل باید از فراخوانی سیستمی `open` و برای نوشتن در فایل از فراخوانی سیستمی `write` استفاده کنید. همچنین در این کد لازم است که به کمک فراخوانی سیستمی `getpid`، شماره پردازش مربوطه را گرفته و در کنسول چاپ کنید.

نکات مهم

- در نهایت کدهای خود را در کنار گزارش با فرمت pdf در یک فایل zip آپلود نمایید. (دقت کنید تمامی فایل‌های سیستم عامل را آپلود نکنید و تنها فایل‌هایی را آپلود کنید که در آن‌ها تغییری به وجود آورده‌اید).
- تمامی 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.