Repository Link: <https://github.com/SamanEN/Operating-System-Lab-Projects>

Latest Commit Hash: <hash>

# آشنایی با سیستم‌عامل xv6

سیستم عامل xv6 مشابه Unix v6 نوشته شده و معماری و ساختاری شبیه به آن دارد. این سیستم عامل برای پردازنده‌های مبتنی بر x86 نوشته شده (مطابق با داکیومنت این سیستم عامل؛ xv6-rev11). علاوه بر آن در دفاع از این سخن می‌توان به فایل x86.h اشاره کرد که از دستورات پردازنده‌های x86 استفاده شده است. در دیگر فایل‌های “basic headers”، نظیر asm.h و mmu.h نیز می‌توان اشاراتی به معماری x86 مشاهده کرد.

معماری کلی سیستم عامل Unix بصورت زیر می‌باشد:

همانطور که گفته شد، معماری xv6 نیز از Unix پیروی می‌کند. این موضوع از دسته‌بندی فایل‌ها که شامل file systems، system calls، user-level و... می‌شود نیز قابل مشاهده است.

یک پردازه در xv6 از حافظۀ فضای کاربری (user-space) (شامل دستورات، داده‌ها و استک)، و وضعیت پردازه که فقط برای هسته قابل رؤیت است تشکیل شده است.

xv6 زمان را بین پردازه‌ها تقسیم می‌کند و به صورت نامحسوس پردازنده‌ها را برای اجرای دستورات به پردازه‌ها اختصاص می‌دهد. هروقت یک پردازه قرار است اجرا نشود، سیستم‌عامل register های CPU که حاوی مقادیر مورد نیاز آن پردازه بوده را ذخیره می‌کند تا دفعه بعدی که آن پردازه قرار است اجرا شود، آنها را بازگرداند.

هسته xv6 به هر پردازه یک شناسه یکتا PID (Process Identifier) اختصاص می‌دهد. با استفاده از system call ء getpid() می‌توان PID پردازۀ کنونی را دریافت کرد.

# اضافه کردن یک متن به Boot Message

# اضافه کردن چند قابلیت به کنسول xv6

# اجرا و پیاده‌سازی یک برنامه سطح کاربر

# مقدمه‌ای درباره سیستم‌عامل و xv6

# کامپایل سیستم‌عامل xv6

# اجرا بر روی شبیه‌ساز QEMU

# مراحل بوت سیستم‌عامل xv6

## اجرای بوت‌لودر

## اجرای هسته xv6

## اجرای نخستین برنامه سطح کاربر

# اشکال زدایی

# روند اجرای GDB

# کنترل روند اجرا و دسترسی به حالت سیستم

# اشکال زدایی در سطح کد اسمبلی