

به نام خدا



دانشگاه صنعتی امیرکبیر  
( پلی تکنیک تهران )

سیستم‌های عامل (بهار ۱۴۰۲)

## فاز سوم پروژه

استاد درس:

دکتر جوادی

مهلت نهایی ارسال پروژه:

با توجه به مهلت نهایی ثبت نمرات متعاقبا اعلام خواهد شد

نکته مهم: مهلت ارسال پروژه تا آخرین زمان ممکن قبل از مهلت ثبت نمرات است و امکان تمدید آن وجود نخواهد داشت.

# فاز سوم

## پیاده سازی در کرنل

می‌خواهیم سیستم کال fork را به گونه‌ای تغییر دهیم که به جای کپی کردن تمام پیج‌های یک پردازش، فقط پیج‌هایی که روی آنها نوشته می‌شود را کپی کند. برای این کار ابتدا باید بفهمیم هر پیج را چند پردازش استفاده می‌کنند و از مکانیزم Page Fault برای تشخیص اینکه چه موقع روی یک پیج نوشته می‌شود استفاده کنیم.

## شمردن تعداد ارجاع‌ها به یک پیج

سیستم عامل 6xv امکان به اشتراک گذاری پیج‌ها را ندارد هرگاه کار یک پردازش با یک پیج تمام شد پیج آزاد می‌شود. برای اینکه بتوان پیج‌ها را به اشتراک گذاشت نیاز است که تعداد پردازش‌هایی که از یک پیج استفاده می‌کنند را نگه داریم تا هرگاه این تعداد به صفر رسید پیج آزاد شود.

در `kalloc.c` یک ساختار `kref` تعریف کنید که یک آرایه در این ساختار تعداد رفرنس‌های هر پیج را نگه دارد. توابع کمکی ایجاد کنید که تعداد رفرنس‌ها را تغییر دهد (`Increment`, `Decrement`, `Set`). سپس تعداد رفرنس‌ها را در تابع `kfree` استفاده کنید تا فقط هنگامی که رفرنس‌ها به صفر رسید پیج آزاد شود.

به دلیل چند هسته‌ای بودن سیستم عامل، یک `spinlock` نیز در این ساختار تعریف کنید. در سیستم عامل 6xv قفل‌ها به صورت یک متغیر با تایپ `struct spinlock` هستند که قبل از استفاده باید روی آنها تابع `initlock` صدا زده شود. می‌توانید این کار را در تابع `initlock` انجام دهید.

## ایجاد فلگ `PTE_COW` برای تشخیص پیج‌هایی که به صورت `Copy On Write` هستند

تایپ `pte_t` در 6xv یک `Page Table Entry` را به ما نشان می‌دهد. در واقع ۱۰ بیت ابتدایی این متغیر یک `bit field` است که هر بیت آن یک فلگ از پیج است. در فایل `riscv.h` می‌توانید این فلگ‌ها را از جمله `PTE_V` که نشان دهنده `Valid` بودن پیج و `PTE_W` که نشان دهنده قابل نوشتن بودن پیج است مشاهده کنید. به طور پیش فرض 6xv از ۵ بیت آن استفاده می‌کند. یک بیت دیگر آن را نیز برای `PTE_COW` قرار دهید تا مشخص شود که پیج به صورت `Copy On Write` است یا نه.

## تغییر رفتار تابع fork با پیج ها

تابع fork با استفاده از تابع `uvmcopy` پیج های پردازش پدر را برای پردازش فرزند کپی می کند. باید این تابع را به گونه ای تغییر دهیم که عملیات زیر را انجام دهد:

- برای پیج هایی که قابل نوشتن هستند بیت `PTE_W` را صفر و بیت `PTE_COW` را یک کند.
- حافظه جدید تخصیص ندهد و پیج را کپی نکند.

## مدیریت Page Fault

اینترپت ها و ترمپ های مختلف در تابع `usertrap` در `trap.c` مدیریت می شوند. یک شرط به `usertrap` اضافه کنید که اگر شماره اینترپت ۱۵ را خواند کد `Page Fault Handler` ما را اجرا کند. از آنجایی که `PTE_W` را برای پیج های fork شده صفر کردیم هنگام نوشتن بر روی آن قسمت های حافظه `Page Fault` خواهیم داشت. کد هندلر ما باید عملیات زیر را انجام دهد (به همراه چک کردن خطاهای احتمالی):

- ابتدا `pte` پردازش ای که ایجاد ترمپ کرده را می گیرد.
- با استفاده از `kalloc` یک پیج جدید تخصیص می دهد.
- فلگ `PTE_W` را در هر دو پیج یک و فلگ `PTE_COW` را در هر دو پیج صفر می کند.
- محتویات پیج اولیه را در پیج جدید کپی می کند.
- پیج قدیمی را `Free` می کند. دقت کنید با انجام دادن مراحل قبل اگر پردازش (های) دیگری در حال استفاده از پیجی که قرار است آزاد شود باشد پیج در واقع آزاد نخواهد شد و پردازش های دیگر همچنان می توانند از آن استفاده کنند.
- پیج جدید را به پردازش می دهد.

## برنامه تست

در حالت عادی اگر پردازش ای داشته باشیم که بیشتر از نصف حافظه در دسترس را استفاده کند نمی توانیم آن را fork کنیم. در فاز اول تشخیص دادید که میزان کل حافظه ی در دسترس چقدر است. برنامه ای بنویسید که با استفاده از `sbrk` مقدار  $\frac{2}{3}$  این حافظه را درخواست کند و سپس fork شود. در صورتی صحیح بودن پیاده سازی کرنل شما، برنامه باید بدون ارور اجرا شود.

از شما درخواست داریم که یک **private repository** در گیت هاب درست کنید و تغییرات کد خود را مرحله به مرحله Commit کنید و در صورت تمایل می توانید هر یک از تدریس یاران را به پروژه ی خود اضافه کنید. دقت کنید که شما نبایستی برنامه های خود را با دیگر دانشجویان به اشتراک بگذارید.

## توضیحات

- پروژه شما تحویل اسکایی خواهد داشت بنابراین از استفاده از کدهای یکدیگر یا کدهای موجود در وب که قادر به توضیح دادن عملکرد آنها نیستید، پرهیزید.
- ابهامات خود را در بات سوالات درس در تلگرام مطرح کنید و ما در سریع ترین زمان ممکن به آنها پاسخ خواهیم داد.
- به دلیل اینکه چرا از شما خواسته شده که یک کار را انجام دهید خوب فکر کنید. هنگام ارائه از شما در این مورد سوال خواهد شد.

## آنچه که باید ارسال کنید:

یک فایل زیپ با نام Sid1\_Sid2\_hw3.zip ( که Sid را با شماره دانشجویی خود و هم گروهی خود جایگزین کنید) که شامل دو مورد زیر است:

- گزارش خیلی مختصر از آنچه که انجام داده اید تا قابلیت Copy On Write Paging به سیستم عامل 6xv اضافه شود.
- پوشه ای که در آن کدهای شما وجود دارد. دقت کنید که **تنها و تنها فایل هایی را که تغییر داده اید یا اضافه کرده اید** را برای ما بفرستید.

موفق باشید

تیم تدریسیاری درس سیستم های عامل