



دانشکده مهندسی
کامپیوتر و فناوری اطلاعات

پروژه‌ی سوم

دکتر جوادی

سیستم‌های عامل

پروژه‌ی سوم

مقدمه

در این بخش قرار است با هم گروهی خود دو بخش جدید به سیستم عامل xv6 که در فاز اول و سپس دوم توسعه دادید اضافه کنید. دقت کنید که لازم است از فاز بخش دوم استفاده کنید. در ادامه لازم است تا به هر صورتی که برای دو نفر امکان دارد برای پیاده سازی بخش های لازم تقسیم کار کنید. در هر بخش نکات لازم برای پیاده سازی توضیح داده شده است و سپس نحوه ی پیاده سازی و اجرای فایل تست نیز مشخص شده است. برای تحویل کامل پروژه لازم است که فایل های تست به درستی اجرا شوند. **در صورتی که فاز دوم را هیچکدام از اعضا پیاده سازی نکرده اند آن را پیاده سازی کنید و یا میتوانید از کد فاز دوم هم کلاسی های خود کمک بگیرید.**

در ابتدا لازم است بررسی کنید که نحوه ی ذخیره memory page در سیستم عامل xv6 چگونه است و هر صفحه چه سازی بر حسب بایت دارد.

پیاده سازی در کرنل

میخواهیم سیستم کال fork را به گونه ای تغییر دهیم که به جای کپی کردن تمام صفحه های یک پردازش، فقط صفحه هایی که روی آن ها نوشته میشود را کپی کند. برای این کار ابتدا باید بفهمیم که چند پردازش از هر صفحه استفاده میکنند و از مکانیزم page fault برای تشخیص اینکه چه موقع روی صفحه نوشته میشود استفاده کنیم.

شمردن تعداد ارجاع ها به یک پیج

این سیستم عامل امکان به اشتراک گذاری صفحه ها را ندارد. هر گاه کار پردازش با یک صفحه تمام شد صفحه آزاد میشود. برای این که بتوان صفحه ها را به اشتراک گذاشت نیاز است تعداد پردازش هایی که از یک صفحه 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 ارجاعات هر صفحه را ذخیره کند. توابع کمکی ایجاد کنید که تعداد ارجاع را تغییر دهد (`increment`, `decrement`, `set`). سپس تعداد ارجاعات را در تابع `kfree` استفاده کنید تا فقط هنگامی که ارجاعات به صفر رسید صفحه آزاد شود.

به دلیل چند هسته ای بود سیستم عامل یک `spinlock` نیز در این ساختار تعریف کنید. در سیستم عامل قفل ها به صورت یک متغیر با نوع `struct spinlock` هستند که قبل از استفاده باید روی آن ها `initlock` صد زده شود. میتوانید این کار را در تابع `initlock` انجام دهید.

ایجاد فلگ `PTE_COW` برای تشخیص صفحه هایی که به صورت `copy on write` هستند:

تایپ `pte_t` در `xv6` یک `page table entry` را به ما نشان می دهد. در واقع ۱۰ بیت ابتدایی این متغیر یک `bit filed` است که هر بیت آن یک فلگ از صفحه است. در فایل `riscv.h` می توانید این فلگ ها را از جمله `pte_v` که نشان دهنده

پروژه سوم

valid بودن و PTE_W که نشان‌دهنده قابل نوشتن بودن صفحه است را مشاهده کنید. به‌طور پیش‌فرض xv6 از ۵ بین آن استفاده می‌کند. یک بین دیگر آن نیز برای PTE_COW قرار دهید تا مشخص شود که صفحه به‌صورت copy on write است یا نه.

تغییر رفتار تابع fork با صفحه‌ها

تابع fork با استفاده از تابع uvmcopy صفحه‌های پردازنده پدر را برای پردازنده ی فرزند کپی می‌کند. باید این تابع را به‌گونه‌ای تغییر دهیم که عملیات زیر را انجام دهد:

:

- برای صفحه‌هایی که قابل نوشتن هستند بیت PTE_W را صفر و بیت PTE_COW را یک کند.
- حافظه‌ی جدید تخصیص ندهد و صفحه‌ها کپی نکند.

مدیریت page fault

وقفه‌ها و ترم‌های مختلف در تابع usertrap در trap.c مدیریت می‌شوند. یک شرط بخ usertrap اضافه کنید که اگر شماره وقفه ۱۵ را خواند کد page fault handler ما را اجرا کند. از آنجایی که PTE_W را برای صفحه‌های fork شده صفر کردیم هنگام نوشتن بر روی آن قسمت‌های حافظه page fault خواهیم داشت. کد handler ما باید عملیات زیر را انجام دهد (به‌همراه چک کردن خطاهای احتمالی):

- ابتدا pte پردازنده ای که ایجاد وقفه کرده را می‌گیرد.
- با استفاده از kalloc یک صفحه جدید تخصیص می‌دهد
- فلگ PTE_W را در هر دو صفحه یک و فلگ PTE_COW را در هر دو صفحه صفر می‌کند.
- محتویات صفحه اولیه را در صفحه جدید کپی کند.
- صفحه قدیمی را free می‌کند. دقت کنید با انجام دادن مراحل قبل اگر پردازنده‌ی دیگری در حال استفاده از صفحه‌ای باشد که قرار است آزاد شود، صفحه در واقع آزاد نمی‌شود و پردازنده‌های دیگر همچنان می‌توانند از آن استفاده کنند.
- صفحه‌ی جدید را به پردازنده می‌دهد.

برنامه تست

در حالت عادی اگر پردازنده‌ای داشته باشیم که بیشتر از نصف حافظه در دسترس را استفاده کند، نمی‌توانیم آن را fork کنیم. برنامه ای بنویسید که با استفاده از sbrk مقدار دو سوم این حافظه را درخواست کند و سپس fork شود. در صورت صحیح بودن پیاده‌سازی کرنل شما باید بدون مشکل اجرا شود.

اضافه کردن memory statistics به دستور top

در این قسمت می خواهیم به دستور تاپی که در فاز قبلی پروژه پیاده سازی کرده ایم مقدار حافظه مورد استفاده هر پردازنده را به ستون های این دستور اضافه کنیم و در بخشی که اطلاعات سیستم را نشان می دهد مقدار حافظه ی خالی و استفاده ی شده ی کل سیستم را را نشان دهیم.

باید مقدار کل حافظه و مقدار حافظه ی استفاده شده را پیدا کنید و سپس آن ها را از هم کم کنید تا مقدار حافظه ی خالی را هم حساب کنید. در سیستم عامل xv6 عملیات مروط به حافظه همگی با واحد memory page انجام می شوند. این صفحه ها در قالب یک لیست پیوندی در سیستم عامل نگهداری می شوند. برای دسترسی به این لینکدلیست باید به ساختار kmem مراجعه کنید. واحد خروجی شما باید به بایت باشد پس نیاز است در فایل های سیستم عامل بگردید و اندازه ی یک صفحه به بایت را پیدا کنید.

دقت کنید که این ساختار توسط چند پردازنده در حال استفاده و تغییر است پس لازم است که عملیات روی آن را به صورت اتمیک انجام دهید.

برای محاسبه ی حافظه ی مورد استفاده هر پردازنده می توانید به ساختار آن مراجعه کنید.

تکمیل دستور top

در این بخش لازم است به دستور top یک ستون اضافه کنید. ستون mem% که درصد حافظه ی مورد استفاده توسط هر پردازنده از کل حافظه ی موجود را نشان می دهد. به بخشی که اطلاعات سیستم را نشان می دهد دو فیلد total memory و memory usage (این مقدار نشان دهنده ی مقدار حافظه ی مصرفی به درصد است) اضافه کنید.

پروژه‌ی سوم

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 می‌توانید هر یک از تدریس‌یاران را به پروژه‌ی خود اضافه کنید. دقت کنید که شما نبایستی برنامه‌های خود را با دیگر دانشجویان به اشتراک بگذارید.

توضیحات

- پروژه شما تحویل آنلاین 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 گروه درس در تلگرام مطرح کنید و ما در سریع‌ترین زمان ممکن به آنها پاسخ خواهیم داد.

آنچه که باید ارسال کنید:

- یک فایل زیپ با نام OS_P3_Sid1_Sid2.zip (که Sid1 را با شماره دانشجویی خود و دیگری را با شماره دانشجویی هم‌گروهی خود جایگزین کنید) که شامل مورد زیر است:
- پوشه‌ای که در آن کدهای شما وجود دارد. دقت کنید که **تنها و تنها فایل‌هایی را که تغییر داده‌اید یا اضافه کرده‌اید** را برای ما بفرستید. آپلود یک نفر از اعضا کافی است.

موفق باشید

تیم تدریس یاری درس سیستم‌های عامل