به نام خدا

سیستم‌های عامل پیشرفته
نیم‌سال دوم ۰۱-۰۲
استاد: دکتر اسدی

دانشکده مهندسی کامپیوتر

تمرین سری اول

- پاسخ تمرین‌های تئوری را به صورت فایل تایپ شده (به زبان فارسی) در فرمت PDF در قسمت مربوطه در سامانه CW بارگذاری نمایید.
- پرسش‌های خود را می‌توانید در فروم ایجاد شده در سایت درس مطرح کنید.

تمرینات تئوری

۱- سیستمی را فرض کنید که دربرگیرنده‌ی ۴ پردازنده‌ی اینتل که هر یک ۲۰ هسته دارد. هر هسته در این سیستم دارای ۲ ریسمان است بنابراین مجموعاً هر پردازنده ۴۰ ریسمان دارد. همچنین میزان حافظه‌ی اصلی این سیستم ۲۵۶ گیگابایت می‌باشد. بارکاری را به فرض نمایید که تعداد ۱ میلیون چانک حافظه‌ی اصلی با اندازه‌ی ۱۰۲۴ کیلوبایت را allocate می‌شود. سپس تمامی حافظه به صورت معکوس، deallocate می‌شود. در صورتی که جهت اجرای تکه کد زیر از ۲۵۶ عدد ریسمان به صورت موازی استفاده شود، تعداد TLB Shootdown IPI را محاسبه نمایید. آیا امکان کاهش در تعداد TLB Shootdown وجود دارد؟ ایده‌ی خود را به طور کامل شرح دهید.

```
Void *Func (void * args)
Void *Arr[1000000];
For (int  a = 0;  0 < 1000000; a++)
{
  For(int b= 0; b < 1000000; b++)
  Arr[b] = malloc(1048576);
  For(int c =0; c < 1000; c++)
  Free(Arr(999999-c));
}
```

تمرینات عملی

۱- تاثیر فراخوانی‌های سیستمی در تغییرات TLB

در یک برنامه چندریسمانه، ریسمان‌های یک پردازه به فضای آدرسی مشترکی دسترسی دارند. در نتیجه، کرنل باید انواع خاصی از تغییرات فضای آدرس را به TLB هر هسته‌ای که هر یک از آن ریسمان‌ها را اجرا می کند، ارتباط دهد. فراخوانی‌های سیستمی همچون mmap()، mprotect() و madvise() بر این تغییرات تاثیر دارند. استفاده از این دستورات با تعداد ریسمان بالا عموما باعث ایجاد TLB miss و TLB Shootdown می‌گردد.

در این بخش قصد داریم با اجرای تعدادی از این دستورات، تغییراتی در TLB پردازنده بوجود آورده و تاثیر آنها را بوسیله ابزار perf مشاهده کنیم.

۱,۱ دستور mprotect()

این فراخوانی سیستمی به جهت تعیین یا تغییر حفاظت مورد نیاز برای صفحات حافظه پردازه‌ها استفاده می‌شود. این صفحات حافظه شامل یک بخش یا تمام محدوده آدرس در بازه [addr , addr+l en-۱] می‌باشد.

برنامه‌ای بنویسید که شامل ۴ ریسمان بوده و هر کدام بر روی یک هسته اجرا می‌شوند. ابتدا ریسمان A به اندازه یک صفحه ۴ کیلوبایتی از حافظه اصلی را بوسیله دستور malloc() رزرو می‌کند. سپس ریسمان B در حین اجرای هر کدام از حالات زیر، بوسیله دستور mprotect() قسمت رزرو شده را فقط از **خواندن** محافظت می‌کند:

الف) هر دو ریسمان C و D بطور همزمان در حال خواندن از قسمت محافظت شده می‌باشند.

ب) ریسمان C در حال خواندن از قسمت محافظت شده و ریسمان D در حال نوشتن در قسمت محافظت شده می‌باشد.

ج) هر دو ریسمان C و D بطور همزمان در حال نوشتن در قسمت محافظت شده می‌باشند.

۱- با استفاده از ابزار perf، تعداد page fault ها، تعداد TBL miss ها و رویدادهای مربوط به TLB را در سه حالت بالا مشخص کرده و اسکرین شات بگیرید.

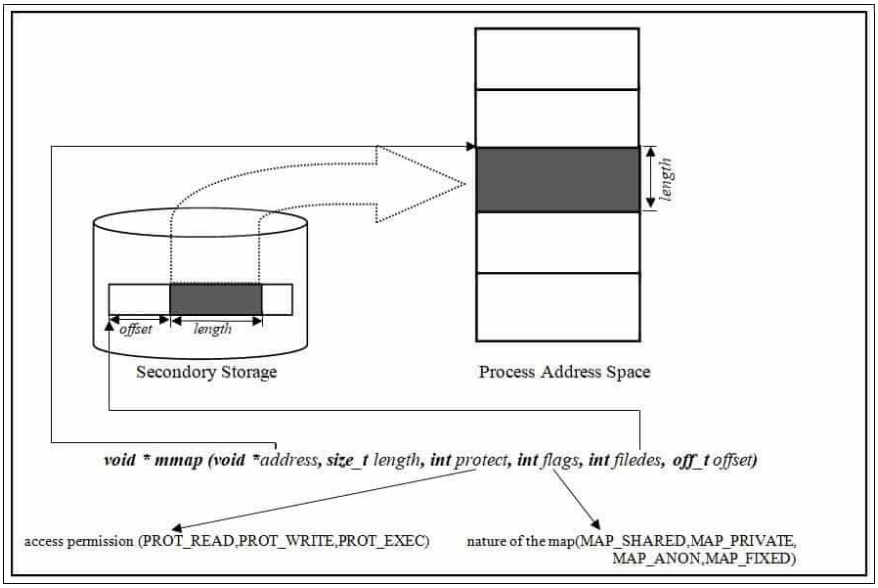
۲- نتایج بدست آمده از ابزار perf برای این سه حالت را مقایسه و تحلیل کنید.

۳- مشخص کنید در کدام یک از این سه حالت، خطای SIGSEGV رخ می‌دهد (خروجی در کد مشخص باشد).

۴- موارد ۱ و ۲ و ۳ را به ازای اندازه صفحه ۱ کیلوبایت نیز بررسی کنید.

۲,۱ دستور mmap()

این فراخوانی سیستمی برای نگاشت فضای آدرس پردازه و فایل‌های موجود در حافظه‌های جانبی استفاده می‌شود. هنگامی که یک فایل به فضای آدرس یک پردازه نگاشت می‌شود، می‌توان مانند یک آرایه به آن فایل در برنامه خود دسترسی داشته باشیم. بطور کلی، نحوه کارکرد این دستور را در شکل زیر مشاهده می‌کنید:



برنامه‌ای بنویسید که شامل ۴ ریسمان می‌باشد و هر کدام بر روی یک هسته اجرا می‌شوند. ابتدا ریسمان A قسمتی از حافظه جانبی را به حافظه اصلی نگاشت می‌کند. برای راحتی، این بخش از حافظه اصلی را بخش Exchanged نام گذاری می‌کنیم.

سپس ریسمان B در حین اجرای هر کدام از حالات زیر، بوسیله دستور ()mmap، قسمتی دیگر از حافظه جانبی را به بخش Exchange نگاشت می‌کند:

الف) هر دو ریسمان C و D بطور همزمان در حال خواندن از قسمت Exchange می‌باشند.

ب) ریسمان C در حال خواندن از قسمت Exchange و ریسمان D در حال نوشتن در قسمت Exchange می‌باشد.

ج) هر دو ریسمان C و D بطور همزمان در حال نوشتن در قسمت Exchange می‌باشند.

۱- با استفاده از ابزار perf، تعداد page fault ها، تعداد TBL miss ها و رویدادهای مربوط به TLB را در سه حالت بالا مشخص کرده و اسکرین شات بگیرید.

۲- نتایج بدست آمده از ابزار perf برای این سه حالت را مقایسه و تحلیل کنید.

۳- اگر ریسمان B قسمت Exchange را از حافظه اصلی برداشته و دوباره آن را به حافظه جانبی بازگرداند، به این عملیات نگاشت معکوس یا unmap می‌گویند که با دستور ()munmap تعریف می‌شود. اجرای این دستور توسط ریسمان B را در حین اجرای هر کدام از حالات الف – ب- ج فرض کرده و موارد ۱ و ۲ را بررسی کنید.

۳,۱ دستور ()madvise

این فراخوانی سیستمی در مورد نحوه مدیریت ورودی/خروجی صفحه‌بندی، اطلاعات و توصیه‌هایی در اختیار کرنل قرار می‌دهد. با این کار، یک برنامه می‌تواند به هسته اعلام کند چگونه انتظار دارد از برخی مناطق حافظه نگاشت شده یا مشترک استفاده کند، بطوری که هسته نیز می‌تواند تکنیک‌های مناسب پیشخوان و ذخیره را انتخاب کند.

یک برنامه بنویسید که با تعداد متغیری ریسمان، دستور ()madvise را به تعداد بسیار زیاد فراخوانی کند. هر ریسمان باید این دستور را در یک حلقه با حداقل تعداد ۱۰۰,۰۰۰ بار اجرا کند.

نکات زیر را در نوشتن برنامه در نظر داشته باشد:

۱) تعداد ریسمان‌ها را به عنوان ورودی از کاربر دریافت کنید. (حداکثر تعداد ریسمان انتخابی یکی کمتر از تعداد هسته‌های پردازنده باشد)

۲) هر ریسمان تنها روی یک هسته خاص اجرا شود.

۳) همزمانی یا سینک بودن ریسمان‌ها بوسیله pthread_barrier انجام شود.

۴) زمان اجرای دقیق برنامه به ازای تعداد ریسمان‌های متفاوت ثبت شود.

برنامه مذکور را به ازای چند ریسمان متفاوت اجرا کرده و موارد زیر را انجام دهید:

الف) با استفاده از ابزار perf، تعداد page fault ها، تعداد TBL miss ها و رویدادهای مربوط به TLB را در سه حالت بالا مشخص کنید.

ب) نموداری رسم کنید که نشان دهنده تاثیر تعداد ریسمان‌های متفاوت در میزان TLB miss، page fault و cache miss ها باشد.

ج) نموداری رسم کنید که نشان دهنده تاثیر تعداد ریسمان‌های متفاوت در تاخیر زمان اجرای برنامه می‌باشد.

نکات مهم تمرین:

- پاسخ سوالات به زبان فارسی نوشته شوند.
- تمامی برنامه‌های عملی بایستی به زبان C نوشته شوند.
- جهت اجرای مانیتورینگ تغییرات TLB، اسکریپت‌های ابزار perf را برای هر سوال نوشته و همراه کدها ارسال کنید.
- نکات ذیل میبایست در اسکرین شات‌های گرفته شده رعایت شود:
  - تصاویر واضح و شفاف باشند.
  - زمان و تاریخ کنونی سیستم (شمسی یا میلادی) مشخص باشد.
  - نام کاربری در محیط command line شامل نام و نام خانوادگی و شماره دانشجویی فرد به فرمت زیر باشد:

    firstName_lastName_studentID
- خروجی این تمرین، یک فایل PDF شامل تمام تصاویر و توضیحات هر سوال، به همراه کدهای نوشته شده می‌باشد.
- به دلیل محدودیت‌هایی که در اجرای دستورات سطح کرنل در ماشین مجازی وجود دارد، توصیه می‌شود حتما یک سیستم عامل لینوکس را روی سیستم خود نصب نمایید.
- تمامی کدهای نوشته شده (شامل برنامه‌های به زبان C و اسکریپت‌های perf) توسط تیم دستیاران بررسی و اجرا می‌شوند. بنابراین، حتما قبل از ارسال از کارکرد صحیح کدها و اسکریپت‌ها اطمینان حاصل نمایید.
- استفاده از کدهایی که ممکن است در اینترنت بیابید، مجاز نیست و شباهت کدهای شما با کدهای آماده، یا کدهای سایر دانشجوهای درس، به منزله‌ی تقلب و ثبت نمره‌ی **صفر** خواهد بود.

موفق باشید