

در این تمرین قصد داریم رفتار TLB Shutdown هایی را بررسی کنیم که توسط اجرای فراخوانی های سیستمی و دستورات لینوکسی بر روی یک میزبان با ۲ ماشین مجازی همزمان اجرا میشوند و تاثیرات استفاده از مجازی سازی را تحلیل نماییم.

وظیفه شما، طراحی و پیاده سازی برنامه هایی چند ریسمانه است که بر روی هر یک از ماشین های مجازی اجرا شده و از دستورات و فراخوانی هایی استفاده میکند که باعث تشدید TLB Shutdown میگردد. در نهایت میبایست بوسیله ابزار perf گزارش هایی از TLB بدست آورید.

پیکربندی ماشین های مجازی

همانطور که گفته شد، در این تمرین میبایست ۲ ماشین مجازی بطور همزمان بر روی سیستم خود بالا آورده و دستورات و روندهایی که در ادامه گفته میشود را بر روی دو ماشین اجرا نمایید. مشخصات حداقلی هر ماشین باید بصورت زیر باشد:

- 1) پردازنده با دو ریسمان
- 2) ۴ گیگابایت حافظه اصلی
- 3) ۲۰ گیگابایت حافظه ذخیره سازی (ترجیحا SSD)
- 4) یک توزیع از سیستم عامل لینوکس (ترجیحا نسخه های قبل تر Ubuntu به دلیل سبک تر بودن)

دستورات و روندها

۱- دستور madvise()

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 تعداد بسیار زیاد حداقل تعداد ۱۰۰,۰۰۰ بار اجرا کند.

نکات زیر را در نوشتن برنامه در نظر داشته باشد:

۱) تعداد ریسمان ها را به عنوان ورودی از کاربر دریافت کنید (حداکثر تعداد ریسمان انتخابی یکی کمتر از تعداد هسته های پردازنده باشد)

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

۳) همزمانی یا سینک بودن ریسمان ها بوسیله pthread_barrier انجام شود.

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

الف- این برنامه را بر روی ۲ ماشین بصورت همزمان اجرا کنید. با استفاده از ابزار perf، تعداد page fault ها، تعداد TLB miss ها و رویدادهای مربوط به TLB را در حین اجرای برنامه در هر یک از ماشین ها بدست آورده و گزارش کنید.

ب) این برنامه را در خود لینوکس (بدون ماشین مجازی) اجرا کنید و با استفاده از ابزار perf، تعداد page fault ها، تعداد TBL miss ها و رویدادهای مربوط به TLB را مشخص کنید. نتایج بدست آمده از این قسمت را با نتایج به دست آمده از دو ماشین (قسمت الف) مقایسه کنید

۲- دستور **msync()** این فراخوانی سیستمی به جهت همگام‌سازی صفحات حافظه اصلی (RAM) با فایل‌های موجود در حافظه جانبی (Disk) استفاده میشود. این دستور میتواند با آپشن‌هایی همچون **MS_SYNC**، **MS_ASYNC** یا **MS_INVALIDATE** بکار گرفته شود تا رفتارهای مختلف همگام‌سازی بدست آید

برنامه‌ای بنویسید که با ساخت ۲ ریسمان، بصورت همزمان بر روی یک فایل memory-mapped بنویسد و با فراخوانی دستور **msync()** آپشن **MS_SYNC**، تغییرات را با حافظه جانبی همگام‌سازی کنند.
(راهنما: میتوانید با دستور **mmap()** یک فایل memory-mapped بسازید)

الف) ابتدا این برنامه را بر یک سیستم Bare Metal (بدون استفاده از مجازی‌سازی) به تنهایی اجرا کرده و با استفاده از ابزار perf، تعداد page fault ها، تعداد TLB miss ها و رویدادهای مربوط به TLB را در حین اجرای برنامه بدست آورده و گزارش کنید.
ب- سپس این برنامه را بر روی ۲ ماشین بصورت همزمان اجرا کنید و با استفاده از ابزار perf، تعداد page fault ها، تعداد TBL miss ها و رویدادهای مربوط به TLB را در حین اجرای برنامه در هر یک از ماشین‌ها بدست آورده و گزارش کنید.

ج- نتایج بدست آمده از قسمت الف و ب را با هم مقایسه کنید.

د - (امتیازی) - همین روند را برای آپشن **MS_ASYNC** طی کرده و قسمت الف و ب را انجام دهید. سپس اجرای دستور با این دو آپشن را از نظر تعداد و میزان TLB shutdown مقایسه کنید.

نکات مهم :

- تمامی برنامه‌ها بایستی به زبان C نوشته شوند.
- جهت اجرای مانیتورینگ تغییرات TLB، اسکریپت‌های ابزار perf را برای هر سوال نوشته و همراه کدها ارسال کنید.
- خروجی، یک فایل PDF شامل تمام تصاویر و توضیحات هر سوال، به همراه کدهای نوشته شده می‌باشد.