



به نام خدا
آزمایشگاه سیستم عامل



پروژه پنجم: مدیریت حافظه
(پیاده سازی mmap در xv6)

گروه ۶

اعضای گروه :

محمد سعادت‌ی ۸۱۰۱۹۸۴۱۰

محمد عراقی ۸۱۰۱۹۸۴۳۶

سید عماد امامی ۸۱۰۱۹۸۵۲۷

مقدمه

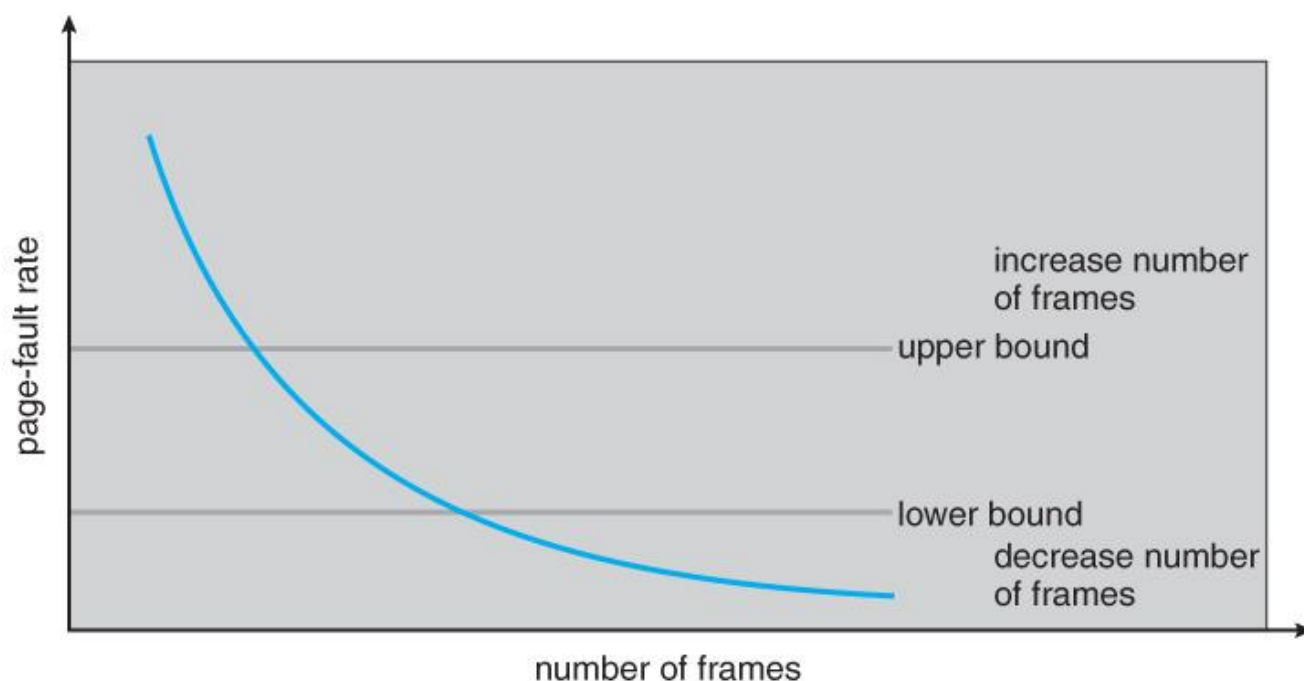
(۱) چرا ساختار سلسله مراتبی منجر به کاهش مصرف حافظه می گردد؟

در این ساختار پردازش ها و task ها با اشتراک گذاری کد ها و اطلاعات به کمک mapping بخش مناسب به صفحات فیزیکی ، مانع مصرف بیش از نیاز حافظه و در نتیجه کاهش مصرف حافظه می شوند.

(۲) روشی برای تخمین فرکانس دسترسی به صفحه های حافظه به کمک بیت های مدخل جدول صفحه ارائه دهید. (راهنمایی: می توان از شیوه بهره گیری از بیت Accessed در لینوکس الگوبرداری کرد).

می توانیم از Page Fault Frequency برای کنترل تعداد فریم های تخصیص یافته به یک فرآیند استفاده کنیم.

با افزایش تعداد فریم ها، ابتدا تعداد خطاهای صفحه به سرعت کاهش می یابد، سپس به نقطه ای می رسد که افزودن فریم های بیشتر به سختی سرعت صفحه بندی را تغییر می دهد. ما مرزهای بالا و پایین را تنظیم می کنیم و فریم هایی را اضافه یا حذف می کنیم تا در آنها باقی بمانند.



یک فرآیند باید فرکانس خطای صفحه ایده آل داشته باشد. اگر اغلب خطا می کند، باید فریم های صفحه فیزیکی بیشتری به آن بدهید. اگر به ندرت ایراد می گیرد، فریم های صفحه فیزیکی زیادی دارد و باید مقداری از آن برداشته و به فرآیندهای دیگر بدهد. هنگامی که همه فرآیندها به طور مکرر خطا می کنند، یکی را برای تعویض انتخاب کنید.

کد مربوط به ایجاد فضاهای آدرس در Xv6

۳) تابع `kalloc()` چه نوع حافظه ای تخصیص می دهد؟ (فیزیکی یا مجازی)

یک صفحه ۴۰۹۶ بایتی از حافظه فیزیکی اختصاص می دهد. اشاره گر را برمی گرداند که هسته می تواند از آن استفاده کند. اگر حافظه قابل تخصیص نباشد ۰ را برمی گرداند.

۴) تابع `mappages()` چه کاربردی دارد؟

از این تابع برای اتصال حافظه مجازی به حافظه فیزیکی و اضافه کردن صفحه جدید به `pgdir` استفاده می شود. PTE هایی برای آدرس های مجازی که از `va` شروع می شوند ایجاد می کند که به آدرس های فیزیکی با شروع `pa` اشاره می کنند. `va` و اندازه ممکن است با صفحه تراز نباشند.

۵) راجع به `walkpgdir()` توضیح دهید. این تابع چه عمل سخت افزاری را شبیه سازی می کند؟

این تابع آدرس PTE را در جدول صفحه `pgdir` که با آدرس مجازی `va` مطابقت دارد، برمی گرداند. اگر `alloc!=0`، هر صفحه جدول صفحه مورد نیاز را ایجاد می کند. در واقع `walkpgdir()` شبیه سازی عمل سخت افزاری ترجمه آدرس مجازی به فیزیکی را انجام می دهد.

شرح آزمایش

۶) دو نقص نگاشت فایل در حافظه نسبت به خواندن عادی فایل ها را بیان کنید.

دلیل اصلی انتخاب ورودی/خروجی فایل نگاشت شده حافظه، عملکرد است. با این وجود، ممکن است معاوضه هایی وجود داشته باشد. روش استاندارد ورودی/خروجی به دلیل سربار تماس سیستم و کپی کردن حافظه پرهزینه است. روش نگاشت حافظه هزینه خود را در خطاهای جزئی صفحه دارد - زمانی که بلوکی از داده ها در حافظه پنهان صفحه بارگذاری می شود، اما هنوز در فضای حافظه مجازی فرآیند نگاشت نشده است. در برخی شرایط، ورودی/خروجی فایل نگاشت شده حافظه می تواند به طور قابل ملاحظه ای کندتر از ورودی/خروجی فایل استاندارد باشد.

یکی دیگر از اشکالات فایل های نگاشت حافظه مربوط به فضای آدرس یک معماری معین است: یک فایل بزرگتر از فضای آدرس پذیر می تواند تنها بخش هایی را در یک زمان نگاشت داشته باشد که خواندن آن را پیچیده می کند. به عنوان مثال، یک معماری ۳۲ بیتی مانند IA-32 اینتل فقط می تواند به طور مستقیم به ۴ گیگابایت یا بخش های کوچکتر از فایل ها رسیدگی کند. حتی مقدار کمتری از فضای آدرس پذیر برای برنامه های جداگانه در دسترس است - معمولاً در محدوده ۲ تا ۳ گیگابایت، بسته به هسته سیستم عامل. اما این اشکال در معماری مدرن ۶۴ بیتی تقریباً حذف شده است.

mmap همچنین نسبت به ابزار استاندارد ورودی/خروجی فایل، مقیاس پذیری کمتری دارد، زیرا بسیاری از سیستم عامل ها، از جمله لینوکس، روی تعداد هسته هایی که خطاهای صفحه را مدیریت می کنند، محدودیت دارند. دستگاه های بسیار سریع، مانند SSD های مدرن NVMe Express، می توانند سربار را به یک نگرانی واقعی تبدیل کنند.

فقط معماری های سخت افزاری با MMU می توانند فایل های دارای نقشه حافظه را پشتیبانی کنند. در معماری های بدون MMU، سیستم عامل می تواند کل فایل را در هنگام درخواست نقشه برداری در حافظه کپی کند، اما اگر فقط به مقدار کمی از فایل دسترسی داشته باشید، این بسیار ضایع و کند است و فقط برای فایل ها کار می کند. که در حافظه موجود جای می گیرد.

نکات منفی mmap:

- هزینه های راه اندازی و تخریب کاملاً قابل توجه. و منظورم قابل توجه است.

چیزهایی مانند دنبال کردن جداول صفحه برای حذف نقشه همه چیز است

تمیز این حسابداری برای حفظ فهرستی از همه موارد است

نقشه برداری ها این تراش TLB است که پس از برداشتن نقشه چیزها مورد نیاز است.

- خطای صفحه گران است. به این ترتیب نقشه برداری پر می شود،

و بسیار کند است

نتیجه گیری: اگر به داده ها به طور تصادفی دسترسی دارید، آن ها را برای مدت طولانی نگه دارید، یا اگر می دانید می توانید آن ها را با سایر فرآیندها به اشتراک بگذارید، از نقشه های حافظه استفاده کنید (MAP_SHARED اگر اشتراک گذاری واقعی وجود نداشته باشد، خیلی جالب نیست). اگر به صورت متوالی به داده ها دسترسی دارید یا پس از خواندن آن ها را دور می اندازید، فایل ها را به طور معمول بخوانید. و اگر هر یک از روش ها برنامه شما را پیچیده تر می کند، این کار را انجام دهید. برای بسیاری از موارد دنیای واقعی هیچ راه مطمئنی برای نشان دادن سریع تر بودن برنامه بدون آزمایش برنامه واقعی شما وجود ندارد و معیار نیست.