OPERATING SYSTEM LAB

SESSION 5

SeyedMahdi HajiSeyedHossein	810100118
Amirali Shahriary	810100173
Alireza Hosseini	810100125

۱: راجع به مفهوم ناحیه مجازی(VMA)در لینوکس به طور مختصر توضیح داده و آن را با xv۶ مقایسه کنید.

به مجموعه حافظه های که در مموری اعم از ram و swap وssd و غیره اطلاق می شود را می گویند از مقایسه آن با توجه به این که xv6 برای اعمال آموزشی توسعه یافته است فلذا از نوشتن سیستم های مدیریتی shared memory و مانند آن اجتناب شده ولی در لینوکس به توجه به گسترده تر بودن کاربری های آن از راهکار های بیشتری بهره گرفته شده است.

۲: چرا ساختار سلسله مراتبی منجر به کاهش مصرف حافظه میگردد؟

نخستین علت آن این است که ما تنها آن بخشی از پردازه که میخواهد از اطلاعات آن استفاده شود را خوانده و قسمت هایی که استفاده نمیگردند را لود بلااستفاده نمیکنیم .
همچنین از آنجایی که فقط آدرس شروع صفحه هر پردازه را نگهداری می کنیم و قسمتی از صفحه پردازه را که دوباره صفحه بندی کردیم با یک آفست انتخاب کرده و به این صورت مقادیر عددی ذخیره شده هم کمتر خواهند شد .
به طور کلی پردازه ها و تسک ها به سادگی با اشتراک گذاری کد و داده ها توسط مپینگ پارت مناسب به صفحه های فیزیکی از هدر رفت حافظه جلوگیری می کند.

۳: محتوای هر بیت یک مدخل(۳۲ بیتی) در هر سطح چیست؟ چه تفاوتی میان آنها وجود دارد؟

در صفحه ورودی برای اشاره به سطح بعدی، از ۲۰ بیت استفاده می‌شود، همچنین ۱۲ بیت برای نگه‌داری سطح دسترسی محفوظ می‌گردد. این ۱۲ بیت در هر دو سطح (directory page و table page) وجود دارد، اما در سطح table page از ۲۰ بیت برای آدرس صفحه فیزیکی استفاده می‌شود. در بیت D (بیت dirty) این دو سطح با یکدیگر تفاوت دارند. در directory page، بیت D نشان‌دهنده این است که صفحه باید در دیسک نوشته شود تا تغییرات اعمال شود، در حالی که در table page این بیت معنایی ندارد.

۴: تابع() kalloc چه نوع حافظه ای تخصیص میدهد؟(فیزیکی یا مجازی)

این تابع یک حافظه فیزیکی به فضای ۴۰۹۶ بایتی اختصاص می دهد.
پوینتری را return میکنه که کرنل می تواند از آن استفاده کند همچنین صفر را return میکند در صورتی که که نتواند حافظه را تخصیص بدهد.

۵: تابع ()mappages چه کاربردی دارد؟

این تابع آدرسdirectory page ، آدرس یک خانه حافظه مجازی، آدرس یک خانه حافظه فیزیکی و سایز را میگیرد و صفحه موجود در حافظه فیزیکی را به توجه به آدرس و سایزی که به آن دادیم در آدرسی که در حافظه مجازی به آن دادیم بازگذاری میکند. این کار برای دسترسی به متغیرهای پردازه در حال اجرا است تا صفحه آن بتواند به درستی بارگذاری شود و تغییر داده شود همچنین این تابع صفحه جدید را به pgdir اضافه میکند و کلا حافظه مجازی را به فیزیکی متصل می کند.

۷: راجع به تابع ()walkpgdir توضیح دهید. این تابع چه عمل سخت افزاری را شبیه سازی میکند؟

تابع فوق ادرس directorypage و همچنین خانه ای از حافظه مجازی را گرفته و ادرس pagetable ای که رد حافظه مجازی داریم را از directoy page بر میگرداند و اگر لازم باشد جدول مورد نیاز را میسازد. این تابع عمل سخت افزاری ترجمه آدرس مجازی به فیزیکی را شبیه سازی میکند.

۸: توابع allocuvm و mappages که در ارتباط با حافظه ی مجازی هستند را توضیح دهید.

تابع allocuvm:

این تابع برای اختصاص حافظه مجازی به یک فرآیند (پروسه) در حال اجرا استفاده می‌شود.
زمانی که یک فرآیند نیاز به حافظه بیشتر دارد، تابع allocuvm برای تخصیص صفحات حافظه مجازی به آن فرآیند فراخوانی می‌شود.
این تابع با درخواست حافظه مجازی جدید، صفحات مجازی را در فضای آدرس فرآیند اختصاص می‌دهد و مرتباً به تعداد صفحات مجازی مورد نیاز را تخصیص می‌دهد.
تابع mappages:

تابع mappages برای اتصال حافظه مجازی به حافظه فیزیکی استفاده می‌شود.
زمانی که یک صفحه حافظه مجازی فرآیند باید به یک صفحه حافظه فیزیکی متناظرش نگاشت شود، تابع mappages به کار می‌رود.
این تابع با استفاده از مکانیسم‌های مدیریت حافظه سیستم عامل، مطابق با جدول صفحه‌ها (page table)، نگاشت مستقیم بین حافظه مجازی و حافظه فیزیکی فراهم می‌کند.
با ترکیب این دو تابع، سیستم عامل xv6 توانمندی را فراهم می‌کند تا به صورت ایمن و کارآمد حافظه مجازی را به فرآیندها اختصاص دهد و ارتباط مستقیم بین حافظه مجازی و حافظه فیزیکی را برقرار کند.

۹: شیوه ی بارگذاری برنامه در حافظه توسط فراخوانی سیستمی exec را شرح دهید.
نخست inode مربوط به path داده شده را یافته و در ip ذخیره می شود سپس elf header فایل مربوطه را چک و مطمئن می شویم که فایلی اجرایی دارای اعتبار باشد سپس مجموعه ای جدید از جدول های صفحه برای پراسس ها را با کمک تابع setup kvm ایجاد کرده و روی elfheader ها پیمایش انجام میدهیم . در ادامه به کمک readi هدر های خوانده شده را در ph ذخیره سازی کرده و فضای حافظه جدید برای این بخش process ها متناسب با مقدار نیاز تخصیص می دهیم و با کمک تابع loaduvm این بخش را به حافظه ای که گرفتیم

لود می کنیم. به طور کلی اول یه آدرس می سازیم سپس آدرس را چند بخش میکنیم (user space – kernel space – stack) پیج هایش را مپ کرده و page table هایش را ساخته و اضافه کنه به page directory .

Refrences :

https://www.cs.ucr.edu/~csong/cs153/19f/lab4.html
https://github.com/naelag/lab2-f17/
XV6 book