

در لینوکس فضای آدرس های مجازی یک فرایند به بخش های به نام vma تقسیم می شود.هر vma بازه ای از آدرس های مجازی و attribute هایی را توصیف می کند که می توانیم مثلا با استفاده از آن ها بفهمیم آیا ناحیه برای فایل خاصی است یا مربوط به دیتا های پویاست

همانطور که می دانید لینوکس از جدول صفحات برای نگاشت آدرس های مجازی به آدرس های فیزیکی استفاده می کند.vma شامل ورودی هایی از جدول صفحات می باشد که تعیین می کند هر آدرس مجازی به کدام بخش ازحافظه ی فیزیکی اشاره دارد.

در xv6 فضای آدرس هر فرایند به چند بخش دیتا استک وکد تقسیم می شود .مانند لینوکس ما اینجا نیز جدول صفحات داریم ولی سیاست های ساده تری برای مدیریت حافظه استفاده می گردد.



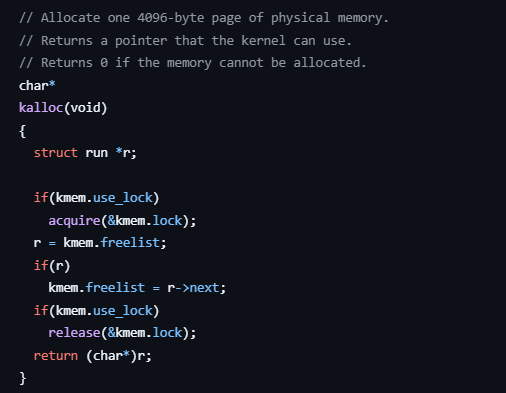
در ساختار سلسله مراتبی ما lazy allocation داریم.به این معنی که همه ی جداول لزوما در حافظه ی فیزیکی ایجاد یا نگهداری نمی شوند و فقط بخش هایی که مورد دسترس است در حافظه نگهداری می شود.

همچنین در ساختار خطی ما به فضای بزرگی برای ذخیره سازی page table نیازمندیم درحالی که وقتی ساختار سلسله مراتبی است هر جدول به چند سطح کوچکتر تقسیم می شود.



در سطح جدول صفحات 20 بیت داریم که مربوط به آدرس بیس حافظه ی فیزیکی است.12 بیت دیگر بیت های کنترلی هستند.مثلا بیتی به اسم present داریم که نشان می دهد آیا این صفحه در حافظه ی فیزیکی وجود دارد یا نه یا مثلا بیت read /write داریم که به ما می گوید اجازه ی نوشتن خواندن داریم یا نه.





این تابع حافظه ی فیزیکی را تخصیص می دهد. این تابع از free list یک پیج خالی با اندازه 4096 بایت پیدا می کند و پوینتری به آن را بر می گرداند.

در این تابع ابتدا چک می کنیم که آیا از لاک استفاده شده و اگر شده آن راaqcuire می کنیم.سپس در free list دنبال صفحه ی خالی می گردد و در صورت وجود پوینتر را به صفحه ی خالی بعدی می دهد.اگر از لاک استفاده کردیم آن را ریلیز می کنیم و سپس پوینتر را برمی گردانیم.