1) برای پیدا کردن یک آدرس فیزیکی یک صفحه، ابتدا 10 بیت اول آدرس، page directory را انتخاب میکند. اگر page directory موجود بود، 10 بیت بعدی، page table را انتخاب میکند (اگر موجود باشد). این دو لایه کردن سرچ، باعث میشود که اگر مثلا page directory وجود نداشت، بازه گسترده ای از آدرس ها همان اول به عنوان خطا شناخته میشوند و نیازی به چک کردن آن ها نیست بنابراین دسترسی به حافظه به طور چشمگیری کاهش پیدا میکند.

2) page directory، 1024 رفرنس به page table ها دارد. (1024 آدرس 4 بایتی). هر page table 1024 تا pte دارد. بنابراین همانطور که در سوال یک گفته شد، 10 بیت پرارزش آدرس مجازی، page directory را انتخاب میکند و اگر موجود بود 10 بیت بعدی page table را انتخاب میکند.

2) 4096 بایت حافظه فیزیکی اختصاص میدهد و اشاره گر به آن صفحه را برمیگرداند. اگر تخصیص حافظه موفقیت آمیز نبود، 0 برمیگرداند.

3) این تابع یک pte برای آدرس مجازی va میسازد که متناظر است با آدرس فیزیکی pa. همچنین اندازه این صفحه را هم به عنوان ورودی میگیرد.

4) این تابع آدرس pte را در pgdir (page directory) بر میگرداند. پس برای این کار به آدرس مجازی نیاز دارد که به عنوان آرگومان میگیرد. در واقع معادل سخت افزاری آن یک look up table میتواند باشد که آدرس میگیرد و عدد متناظر آن را خروجی میدهد.

5) این داده ساختار اشاره گری به لیست صفحه های آزاد دارد و همچنین یک lock جهت محافظت از این لیست. لیست صفحه های آزاد را به صورت یک لینک لیست نگه میدارد. (هر نود، اشاره گری به نود بعدی دارد.

6) shmat: این سیستم کال، قسمتی (segment) از حافظه مشترک را که با یکی از ورودی هایش مشخص شده است، به فضای آدرس پردازه متصل میکند.

Shmget: حافظه مشترکی متناظر با یکی از آرگومان هایش را برمیگرداند. اگر وجود نداشت، یک صفحه جدید، allocate میکند.