

(1)

الف) در سیستم paging اصلاً external fragmentation وجود ندارد. [ع] 15

ب) معماری سلسله مراتبی به دلیل نخست بودن page table در حافظه، از نظر مدیریت حافظه کارآمدتر است. [ص] 15

اگر از نظر سرعت مقایسه شده باشد، پاسخ مخ هم نمره کامل می گیرد. (باید اشاره شود که در معماری سلسله مراتبی گذر است.)

ج) از آنجایی که حجم برنامه های موجود در RAM از نظر منطقی، همیشه بیش تر از حافظه فیزیکی

است، تعداد فریم های فیزیکی همواره کمتر از تعداد صفحات منطقی خواهد بود. [ص] 15
برای مثال یک برنامه 50GB که در 8GB RAM اجرام شود تعداد page های بیش تری نسبت به تعداد کل frame ها دارد.

$$\begin{array}{l} \text{اگر هر frame را} \\ \text{page 9} \Rightarrow \frac{50GB}{128KB} > \frac{8GB}{128KB} \\ \text{128KB بگیریم} \end{array}$$

2) براساس ام در حال اجرا است و نیاز به IO دارد. این فرآیند چون به صورت DMA انجام می شود، OS از آن اطلاعی ندارد و اگر در آینده این فرآیند swap شود، مستقوی IO به (تساو) بر روی براسس جدید نوشته می شود. 25

راه حل این این است که ابتدا مستقوی IO را در فضای OS بنویسیم و سپس OS در زمان مناسب، داده را منتقل کند.

که نوشتن این جمله بارم نداشته و با مثال زدن نمره داده شده است.

$$\frac{80}{100} = \frac{h(20+50) + (1-h)(20+50+50)}{50+50} \Rightarrow h = 80\%$$

30 (3)

برای دسترسی به دوبار memory access نیاز داریم که در صورت hit شدن یک بار کم شود.

(1) الف) با کوچک کردن اندازه frame و تعداد frame های پرونده افزایش می یابد که این موضوع سبب افزایش

اندازه page من شود. [15] 15
table

ب) این به ازای هر پرونده یک page در نظر بگیریم از جهت کارایی و انعطاف پذیری بهتر است و امروزه در سیستم های فعلی این روش استفاده می شود. (وقت کمینه در روش paging توضیح داده شده در این فصل،

هر پرونده page مشخص به خود را دارد. [15] 15
table

ج) معماری سلسله مراتبی نه تنها حجم کمتری مصرف نمی کند، بلکه مقداری overhead از نظر حافظه نیز دارد.

دست این جمله که از نظر حجم مصرف بسیار به صرفه تر است، غلط است. [15] 15

(2) در این معماری page نام هر پرونده ها در یک جدول اند و برای تعایز قابل شدن بین آنها برای جست و جو نیاز داریم تا از pid استفاده کنیم. 15

ب) جهت صرف page های به واقع در حافظه فیزیکی قرار دارند، در این جدول هستند پس اندازه page کم شود و از نظر مدیریت حافظه بهتر است. 15

(3)

$$\frac{\text{استفاده از صفحات اشتراکی}}{\text{عدم استفاده از صفحات اشتراکی}} = \frac{(3 + 15 \times 1) \times 128 \text{ KB}}{15 \times 4 \times 128 \text{ KB}} = \frac{18}{60} = 0.3$$

جواب $\frac{46}{60}$ هم نمره کامل دریافت کرده است.
 اگر صفحات اشتراکی نباشند سه صفحه برای کد و یک صفحه برای داده داریم و هر کاربر ۴ صفحه استفاده می کند.

25

اگر از صفحات اشتراکی استفاده کنیم، همان ۳ صفحه استفاده می کنند و هر کدام یک صفحه برای داده به صورت جداگانه دارند.