

سیستم‌های عامل دکتر زرندی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری هشتم

۲۱ آذر ۱۴۰۳

سوال اول

به سوالات زیر پاسخ دهید.

۱. برای هر یک از حالات زیر توضیح دهید کدام یک از روش‌های static linking و dynamic linking بهتر است انجام شود:

- تعدادی برنامه که از کتابخانه‌های مختلف استفاده می‌کنند.

پاسخ

برای برنامه‌هایی که هر کدام از کتابخانه‌های متفاوتی استفاده می‌کنند، روش static linking مناسب‌تر است. دلیل آن این است که هر برنامه با کتابخانه‌های مخصوص خود لینک می‌شود و در زمان اجرا نیازی به بارگذاری پویا و مدیریت نسخه‌های متفاوت کتابخانه‌ها نیست. همچنین از آنجا که کتابخانه‌ها مشترک نیستند، استفاده از dynamic linking صرفه‌جویی چندانی در حافظه ایجاد نمی‌کند و static linking پیکربندی و اجرا را ساده‌تر می‌سازد.

- تعدادی برنامه که همگی از یک کتابخانه استفاده می‌کنند.

پاسخ

برای برنامه‌هایی که همگی از یک کتابخانه‌ی مشترک استفاده می‌کنند، روش dynamic linking بهتر است. زیرا می‌توان تنها یک نسخه از کتابخانه را به صورت پویا در حافظه بارگذاری کرده و همه‌ی برنامه‌ها از آن استفاده کنند. این کار منجر به صرفه‌جویی در حافظه و سهولت در به‌روزرسانی کتابخانه می‌شود.

۲. تفاوت تکه‌تکه‌سازی خارجی و داخلی را توضیح دهید. در هر بخش زیر مشخص کنید کدام یک از تکه‌تکه‌سازی داخلی یا خارجی برای ما می‌تواند مشکل ایجاد کند؟

پاسخ

- External Fragmentation: زمانی رخ می‌دهد که فضای آزاد حافظه به صورت پراکنده بین بخش‌های اشغال‌شده پخش می‌شود و هرچند مجموع فضای آزاد برای اجرای برنامه‌ای جدید کافی است، اما به صورت یک بخش پیوسته در دسترس نیست.
- Internal Fragmentation: زمانی رخ می‌دهد که به یک فرایند، بلوکی بزرگ‌تر از نیاز واقعی‌اش اختصاص داده می‌شود و بخشی از آن بلوک استفاده‌نشده و هدر می‌رود.

- یک ماشین مدیریت حافظه ساده با استفاده از ثبات‌های base و limit و بخش‌بندی ایستا.

پاسخ

یک ماشین مدیریت حافظه ساده با استفاده از ثبات‌های $base$ و $limit$ و بخش‌بندی ایستا معمولاً منجر به Internal Fragmentation می‌شود. زیرا حافظه به بخش‌هایی با اندازه‌ی ثابت تقسیم شده و ممکن است اندازه‌ی بخش از نیاز واقعی فرایند بیشتر باشد، در نتیجه فضای داخلی هدر می‌رود.

- یک ماشین مشابه قسمت قبل با استفاده از بخش‌بندی پویا

پاسخ

یک ماشین با بخش‌بندی پویا (Dynamic Partitioning) با گذشت زمان و تخصیص و آزادسازی حافظه، دچار External Fragmentation می‌شود. در این حالت، شکاف‌های آزاد کوچک و پراکنده بین بخش‌های اشغال‌شده به وجود آمده و ممکن است علی‌رغم وجود فضای آزاد کافی، نتوان یک بخش پیوسته متناسب برای اجرای فرایند جدید یافت.

سوال دوم

فرض کنید در یک سیستم به سه فرآیند حافظه تخصیص داده شده است و مقادیر رجیسترهای پایه و حد آن‌ها به صورت زیر است:

۱. فرآیند ۱: ثبات پایه = ۵۰۰۰، ثبات حد = ۳۰۰۰

۲. فرآیند ۲: ثبات پایه = ۹۰۰۰، ثبات حد = ۲۰۰۰

۳. فرآیند ۳: ثبات پایه = ۱۲۰۰۰، ثبات حد = ۵۰۰۰

آدرس‌های زیر را بررسی کنید که آیا قانونی هستند یا خیر. همچنین اگر قانونی هستند مشخص کنید به کدام فرآیند مربوط هستند و در غیر اینصورت مشخص کنید چرا قانونی نیستند:

۱. آدرس ۶۵۰۰

۲. آدرس ۹۵۰۰

۳. آدرس ۱۴۰۰۰

۴. آدرس ۱۱۰۰۰

۵. آدرس ۱۲۵۰۰

پاسخ

محدوده آدرس‌های قانونی هر فرآیند به صورت زیر است:

۱ فرآیند : $[5000, 5000 + 3000 - 1] = [5000, 7999]$

۲ فرآیند : $[9000, 9000 + 2000 - 1] = [9000, 10999]$

۳ فرآیند : $[12000, 12000 + 5000 - 1] = [12000, 16999]$

حال آدرس‌های داده شده را بررسی می‌کنیم:

۱. آدرس ۶۵۰۰:

• در بازه $[5000, 7999]$ قرار دارد که مربوط به فرآیند ۱ است.

• نتیجه: قانونی و متعلق به فرآیند ۱.

۲. آدرس ۹۵۰۰:

• در بازه $[9000, 10999]$ قرار دارد که مربوط به فرآیند ۲ است.

• نتیجه: قانونی و متعلق به فرآیند ۲.

۳. آدرس ۱۴۰۰۰:

• در بازه $[12000, 16999]$ قرار دارد که مربوط به فرآیند ۳ است.

• نتیجه: قانونی و متعلق به فرآیند ۳.

پاسخ

۴. آدرس ۱۱۰۰۰:

- خارج از تمامی بازه‌های مشخص شده است.
- نتیجه: غیرقانونی، زیرا به هیچ فرآیندی تعلق ندارد.

۱. آدرس ۱۲۵۰۰:

- ۵. در بازه [۱۶۹۹۹، ۱۲۰۰۰] قرار دارد که مربوط به فرآیند ۳ است.
- نتیجه: قانونی و متعلق به فرآیند ۳.

سوال سوم

فرض کنید در یک سیستم حافظه مشخصات زیر داده شده است:

- آدرس منطقی: ۲۰ بیت
 - سایز صفحه: ۸ کیلوبایت (۸۱۹۲ بایت)
۱. تعداد صفحات منطقی موجود در فضای آدرس منطقی چقدر است؟

پاسخ

آدرس منطقی ۲۰ بیت است، پس فضای آدرس منطقی کل به صورت زیر محاسبه می‌شود:

$$2^{20} = 1,048,576 \text{ MByte}$$

سایز صفحه ۸ کیلوبایت است. تعداد صفحات منطقی از تقسیم فضای آدرس منطقی بر سایز صفحه به دست می‌آید:

$$\text{تعداد صفحات منطقی} = \frac{\text{فضای آدرس منطقی}}{\text{سایز صفحه}} = \frac{1,048,576}{8,192} = 128$$

۲. اگر آدرس منطقی 0X45F3A0 تولید شود شماره صفحه (Page Number) و offset داخل صفحه (Page Offset) را محاسبه کنید.

پاسخ

آدرس منطقی 0X45F3A0 به صورت عددی در مبنای ۱۰ برابر است با:

$$0X45F3A0 = 4,584,352$$

برای محاسبه شماره صفحه (Page Number) و Page Offset:

- اندازه یک صفحه بایت 8,192 است.
- شماره صفحه برابر با تقسیم آدرس منطقی بر سایز صفحه (بدون باقی‌مانده) است:

$$\text{Page Number} = \left\lfloor \frac{\text{آدرس منطقی}}{\text{سایز صفحه}} \right\rfloor = \left\lfloor \frac{4,584,352}{8,192} \right\rfloor = 560$$

- Page Offset برابر با باقی‌مانده تقسیم آدرس منطقی بر سایز صفحه است:

$$\text{Page Offset} = \text{آدرس منطقی} \bmod \text{سایز صفحه} = 4,584,352 \bmod 8,192 = 5024$$

بنابراین:

$$\text{Page Number} = 560, \quad \text{Page Offset} = 5024$$

۳. اگر جدول صفحات به صورت زیر باشد، آدرس فیزیکی متناظر با آدرس منطقی 0X45F3A0 را محاسبه کنید:

- Page 0 → Frame 7

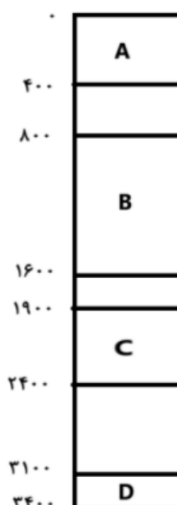
- Page 2 → Frame 3
- Page 5 → Frame 11
- Page 8 → Frame 6

پاسخ

طبق جدول، صفحه منطقی ۵۶۰ را پیدا می‌کنیم. از آنجا که در جدول صفحات داده شده Page 560 مشخص نشده است، نتیجه می‌گیریم که این آدرس منطقی به حافظه فیزیکی نگاشت نشده و دسترسی به آن باعث Page Fault می‌شود.

سوال چهارم

یک سیستم حافظه قطعه‌بندی شده را با حافظه تخصیص یافته مطابق شکل زیر در نظر بگیرید.



شکل ۱: حافظه قطعه‌بندی شده

فرض کنید اقدامات زیر رخ می‌دهد:

- فرآیند E شروع می‌شود و ۳۰۰ واحد حافظه درخواست می‌کند.
 - فرآیند A مقدار ۴۰۰ واحد حافظه دیگر درخواست می‌کند.
 - فرآیند B خارج می‌شود.
 - فرآیند F شروع می‌شود و ۸۰۰ واحد حافظه درخواست می‌کند.
 - فرآیند C خارج می‌شود.
 - فرآیند G شروع می‌شود و ۹۰۰ واحد حافظه درخواست می‌کند.
۱. وضعیت حافظه را پس از هر عمل با استفاده از الگوریتم اولین برآزش توصیف کنید.

پاسخ

(آ) E: درخواست ۳۰۰ واحد حافظه دارد. اولین فضای خالی که می‌تواند ۳۰۰ واحد را جای دهد، بخش B است (۸۰۰-۱۶۰۰). بنابراین:

- فرآیند E ۳۰۰ واحد اول را از این بخش می‌گیرد.
- بخش B به دو بخش تقسیم می‌شود: ۳۰۰ واحد اشغال شده (E) و ۵۰۰ واحد خالی.
- (ب) A: درخواست ۴۰۰ واحد حافظه دارد. اولین فضای خالی که ۴۰۰ واحد حافظه دارد، بخش خالی باقی‌مانده از B است (۵۰۰ واحد). بنابراین:
- فرآیند A ۴۰۰ واحد از این بخش را می‌گیرد.
- باقی‌مانده B: ۱۰۰ واحد خالی.

پاسخ

(ج) B: خارج می‌شود. این باعث می‌شود کل فضای B (۱۰۰ واحد خالی و ۴۰۰ واحد اشغال شده توسط B) به ۵۰۰ واحد خالی تبدیل شود.

(د) F: درخواست ۸۰۰ واحد حافظه دارد. اولین فضای خالی که می‌تواند این درخواست را برآورده کند، بخش C است (۱۹۰۰-۲۴۰۰). بنابراین:

• فرآیند F تمام این بخش را اشغال می‌کند.

(ر) C: خارج می‌شود. کل فضای C آزاد می‌شود (۵۰۰ واحد).

(ز) G: درخواست ۹۰۰ واحد حافظه دارد. اولین فضای خالی که می‌تواند این درخواست را برآورده کند، بخش D است (۳۱۰۰-۳۴۰۰). بنابراین:

• فرآیند G تمام بخش D را اشغال می‌کند.

۲. محتویات حافظه را پس از هر اقدام با استفاده از الگوریتم بهترین برازش توصیف کنید.

پاسخ

(آ) E: درخواست ۳۰۰ واحد حافظه دارد. بهترین فضای خالی، کوچک‌ترین فضای خالی است که درخواست را برآورده کند. بخش B (۸۰۰-۱۶۰۰) ۵۰۰ واحد دارد که بهترین گزینه است. نتیجه مشابه First Fit است.

(ب) A: درخواست ۴۰۰ واحد حافظه دارد. بهترین فضای خالی، بخش خالی باقی‌مانده B است (۵۰۰ واحد). نتیجه مشابه First Fit است.

(ج) B: خارج می‌شود. کل فضای B آزاد می‌شود (۵۰۰ واحد).

(د) F: درخواست ۸۰۰ واحد حافظه دارد. بهترین فضای خالی که می‌تواند این درخواست را برآورده کند، بخش C است (۵۰۰ واحد). نتیجه مشابه First Fit است.

(ه) C: خارج می‌شود. بخش C آزاد می‌شود.

(و) G: درخواست ۹۰۰ واحد حافظه دارد. بهترین فضای خالی، بخش D است. مشابه First Fit.

۳. الگوریتم، بدترین برازش حافظه را چگونه تخصیص می‌دهد؟

پاسخ

(آ) E: درخواست ۳۰۰ واحد دارد. بزرگ‌ترین فضای خالی موجود، بخش B (۸۰۰-۱۶۰۰) است. مشابه First Fit.

(ب) A: درخواست ۴۰۰ واحد دارد. بزرگ‌ترین فضای خالی باقی‌مانده نیز بخش B است. مشابه First Fit.

(ج) B: خارج می‌شود. کل فضای B آزاد می‌شود.

(د) F: درخواست ۸۰۰ واحد حافظه دارد. بزرگ‌ترین فضای خالی، بخش C است. مشابه First Fit.

(ه) C: خارج می‌شود. بخش C آزاد می‌شود.

(و) G: درخواست ۹۰۰ واحد حافظه دارد. بزرگ‌ترین فضای خالی، بخش D است. مشابه First Fit.