دانشگاه اصفهان
دانشکده علوم ریاضی و کامپیوتر

Operating Systems

اصول سیستم‌های عامل

سری نخست تمرینات

نیم‌سال اول ۱۴۰۳-۱۴۰۴

اعضای گروه
محمد ملائی : ۴۰۱۴۰۲۳۰۴۲
علیرضا احمدی وشکی : ۴۰۱۴۰۱۳۰۰۷
یگانه رستگاری : ۴۰۱۴۰۱۳۰۴۰

نام استاد درس

مجتبی رفیعی

تمرین ۱

صحیح یا غلط بودن گزاره‌های زیر را مشخص کنید و دلیل خود را نیز بیان کنید:

(آ) سیستم عامل تنها یک واسط بین کاربر و سیستم کامپیوتری است.
این گزاره غلط است. سیستم‌های‌عامل تعاریف متعارفی ندارد و یک مفهوم کلی است. نسبت‌دادن یک نقش به چنین مفهوم جامعی نیز نمی‌تواند دقیق باشد. از لحاظ کاربردی هم سیستم‌های عامل نقش‌های زیادی را بازی می‌کند که یکی از آن‌ها می‌تواند بازی کردن نقش رابطه بین کاربر و سیستم کامپیوتری باشد. از نقش‌های دیگر آن می‌توان به موارد زیر اشاره کرد:

- مدیریت سیستم‌های کامپیوتری که یکی از ابعاد آن مدیریت کردنِ بهنیه‌ی منابع سیستمی است.
- محیطی برای اجرای برنامه‌های کاربردیست

(ب) هسته ی اصلی سیستم عامل، کرنل است که کنترل روی برخی از اجزای سیستم را در حافظه بر عهده دارد.
این گزاره نیز غلط است .با این فرض که سیستم عامل باید درستی سیستم را تضمین کند، این یک امر بدیهی است که کرنل کنترل بر روی تمامیِ اجزای سیستمی را داشته باشد. فرض کنیم اینطور نباشد. آنگاه بخش‌هایی وجود دارند که زیر نظر سیستم‌عامل فعالیت نمی‌کنند و توسط آن مدیریت نمی شوند. در این حالت کرنل، به عنوان هسته‌ی سیستم‌عامل، چون کنترلی بر روی این اجزا ندارد، پس نمی‌تواند درستیِ کارکرد کل سیستم کامپیوتری را نیز تضمین کند.

(ج) دستورالعمل خواندن ساعت سیستم، فقط در مد کرنل اجرا می شود.
این گزاره نیز غلط است. خواندن ساعت سیستم یک عملیات ممتاز نیست. در بیشتر سیستم‌های عامل، فرآیندها می‌توانند در حالتی به جز حالت کرنل(مثلاً حالت کاربر) به ساعت سیستم دسترسی داشته باشند. به اصطلاح این عملیات، یک عملیات غیرممتاز است. توجه شود که صرف خواندن ساعت سیستم، نباید با مدیریت و دستکاری در منبع ساعت سیستم(که امری حیاتی و حساس است) اشتباه گرفته شود.

(د) تنها تفاوت فرآیند با برنامه این است که فرآیند ماهیتی فعال دارد.
این گزاره صحیح است. در تعریف فرآیند آمده است که: به برنامه‌ی در حال اجرا، فرآیند می‌گویند. از این تعریف اینگونه برداشت می‌شود که اساسی‌ترین تفاوت بین فرآیندها و برنامه‌ها، ماهیتشان است. برنامه‌ها ماهیتی انفعالی(passive) و حالتی ثابت(static) دارند. در حالی که فرآیندها ماهیتی فعال(active) و حالتی منطعف(dynamic) دارند. این تفاوت اساسی شامل تفاوت‌های دیگر نیز بین فرآیند و برنامه می‌شود.

تمرین ۲

به سوالات زیر پاسخ کامل دهید.

(آ) تفاوت سیستم های چند پردازنده ای با سیستم های چندهسته ای چیست؟
اکثر سیستم‌های چندپردازنده‌ای از سیستم SMP استفاده می‌کنند بدین معنا که هر پردازنده register های خاص خودش و همینطور یک کش محلی دارد اما منابع سیستم را مانند BUS با دیگر پردازنده‌ها به اشتراک می‌گذارد.
در سیستم‌های چند هسته‌ای، تنها یک پردازنده وجود دارد که از چند هسته پردازشی تشکیل شده است. این سیستم‌ها می‌توانند از سیستم‌های چند پردازنده‌ای سریع‌تر باشند چراکه ارتباط درون تراشه‌ای سریع‌تر از ارتباط میان تراشه‌های مجزا است. از طرفی یک تراشه به مراتب انرژی کمتری مصرف میکند که موضوعی مهم برای دستگاه‌های موبایل و لپتاپ هااست. در این سیستم‌ها علاوه بر کش محلی یا L_1 کش‌های بیشتر و در لایه‌های مختلفی وجود دارد که از لحاظ اندازه و سرعت نیز متفاوت اند. برای مثال در یک پردازنده‌ی دو هسته‌ای کش L_2 یک حافظه‌ی مشترک بین این دو هسته خواهد بود.
موضوع دیگر مقیاس پذیری‌است که در سیستم‌های چندپردازنده‌ای می‌تواند مشکل‌ساز باشد چراکه با افزودن پردازنده رقابت بر سر منابع و به طور خاص BUS سیستم می‌تواند مشکلاتی را به وجود بیاورد و روی سرعت سیستم تاثیر گذار باشد که البته با تکنیک $NUMA$ تا حد مناسبی قابل رفع است.

(ب) PCB چیست و از چه قسمت هایی تشکیل شده است؟
بلاک کنترل فرآیند مهم‌ترین ساختمان داده در داخل هسته سیستم‌عامل است و شامل اطلاعات موردنیاز برای اجرای یک فرآیند است فرآیندهای سیستم‌عامل به وسیله آن نشان داده میشوند. این بلاک شامل بخش‌های زیر است:

- **وضعیت فرآیند:** وضعیت فرآیند میتواند یکی از موارد جدید، آماده، درحال اجرا، انتظار . یا مسدود باشد.
- **شمارنده برنامه:** شمارنده نشان دهنده‌ی آدرس دستورالعمل بعدی از فرآیند است.
- **ثبات‌های CPU:** به همراه شمارنده برنامه، مقدار ثبات‌های CPU باید ذخیره شوند تا پس از رخدادن وقفه یا اتفاقات دیگر، فرآیند بتواند به درستی کار خود را ادامه دهد.
- **اطلاعات زمان‌بندی CPU:** اطلاعاتی از قبیل اولویت فرآیند، اشاره‌گر به صف زمان‌بندی و پارامترهای مربوطه دیگر.
- **اطلاعات مدیریت حافظه:** حاوی اطلاعاتی مانند مقدار ثبات‌های پایه، جداول صفحه و...
- **اطلاعات حسابرسی:** حاوی اطلاعات میزان استفاده از پردازنده، محدودیت‌های زمانی و شماره فرآیند و غیره.
- **اطلاعات وضعیت I/O:** حاوی اطلاعات دستگاه‌های ورودی/خروجی تخصیص داده شده به فرآیند و لیست فایل‌های باز و ...

(ج) عمل تعویض متن با چه هدفی و توسط چه بخشی انجام می شود؟
به هنگام ایجاد وقفه در یک سیستم کامپیوتری، سیستم عامل وظیفهٔ جاری CPU را متوقف و یک روتین کرنل را اجرا می کند. چنین عملیاتی به طور مکرر در سیستم های همه منظوره اتفاق می افتد. هنگامی که یک وقفه در سیستم حادث می شود، سیستم عامل نیاز دارد تا محتوای فرآیند جاری در CPU را به نحوی ذخیره کند که در آینده بتواند برای از سرگیری اجرای فرآیند مذکور استفاده نماید.

(د) چرا مدیریت ورودی/خروجی در سیستم عامل پیچیده است؟
به علت نیازمندی‌های متفاوت و متنوع سیستم، دستگاه‌های I/O گستردگی زیاد دارند. علاوه بر این موضوع تاثیر مستقیم آنها بر قابلیت اطمینان و کارایی یک سیستم باعث پیچیده شدن مدیریت ورودی/خروجی در سیستم‌عامل میشود.

(ه) به کارگیری ویژگی DMA در سیستم های کامپیوتری برای پاسخ به کدام چالش موجود است و چگونه کار می کند؟
اگر دستگاه‌های ورودی خروجی بخواهند داده‌ها را از طریق CPU انتقال دهند با محدودیت‌های ظرفیت CPU مواجه میشوند که باعث کند شدن روند انتقال و همینطور سرباری برای سیستم میشود به همین دلیل از تکنیک DMA استفاده می‌کنیم که روشی برای انتقال داده از حافظه اصلی به دیگر بخش‌های سیستم کامپیوتری بدون درگیر کردن CPU است. عملکرد آن بدین صورت است که DC پس از انجام تنظیمات برای بافرها، اشاره‌گرها و شمارنده‌ها برای دستگاه I/O موردنظر، یک بلوک کامل از اطلاعات را از/به دستگاه و حافظه اصلی بدون دخالت CPU انتقال می دهد. بنابراین تنها یک وقفه برای کل بلوک تولید می شود.

تمرین ۳

با توجه به اینکه فرآیند های زامبی، منابع تخصیص یافته به خویش را رها کرده و عملا خاتمه یافته اند، آیا ممکن است داشتن فرآیند زامبی به تعداد زیاد در سیستم اخلالی ایجاد کند؟ (پاسخ خود را شرح دهید)
بله. تعداد زیاد فرآیندهای زامبی در سیستم کامپیوتری می‌تواند مشکل ساز شود. اگرچه فرآیندهای زامبی از منابعی مانند CPU و RAM استفاده نمی کنند. اما مشکلات دیگری ممکن است رخ بدهد. برخی از این مشکلات عبارت‌اند از:

- **هدر رفت PID:** می‌دانیم که هر فرآیند زامبی یک سطر در بلاک کنترل فرآیندها اشغال می‌کند. پس اگر تعداد این نوع فرآیندها زیاد شود، سطرهای بسیار زیادی را اشغال خواهد کرد. این امر موجب هدررفت PID های قابل دسترس می‌شود که درنهایت می‌تواند منجر به جلوگیری از ایجاد فرآیندهای جدید شود
- **عملکرد سیستمی:** سیستم‌عامل باید این فرآیندهای غیرمفید را زیر نظر بگیرد. و همین اضافه کاری می‌تواند روی عمل‌کرد سیستم کامپیوتری تأثیر منفی داشته باشد.
- **نشت حافظه:** فرآیندهای زامبی به طور مستقیم از حافظه‌ی اصلی استفاده نمی‌کنند، اما می‌تواند باعث بسته نشدن بعضی از فایل‌ها و بافرها شود. این حافظه‌ی اضافی باعث هدررفت حافظه و در مواردی باعث نشت حافظه میشود.

تمرین ۴

در ارتباط بین فرآیندی با استفاده از حافظه ی مشترک، نظر به محدود بودن حافظه در دنیای واقعی چه راهکارهایی برای نامحدود نگه داشتن بافر به نظر شما می رسد؟ توضیح دهید.
با توجه به محدود بودن فضای ذخیره سازی میتوانیم از Buffer حلقوی استفاده کنیم بدین صورت که مقدار ثابتی از حافظه را در نظر میگریم و تا پر شدن از آن استفاده می‌کنیم و سپس از ابتدای آن شروع به نوشتن می‌کنیم انگار که ابتدا و انتهای آن به هم متصل اند.
راه حل دیگر میتواند انتقال داده‌های کمتر استفاده شده به حافظه جانبی در زمان پرشدنش باشد، هرچند که این روش به ما یک Buffer نامحدود نمیدهد اما میتواند بدون نیاز به حذف داده‌ها حجم زیادی از داده‌ها را مدیریت کند.

تمرین ۵

چرخهٔ حیات یک فرآیند پدر و فرزندش که اجرای فرزند، غیر همروند با پدر و فضای حافظه ی آن نیز مستقل از والدش است را تشریح کنید (در صورت لزوم مفروضاتی به مسئله اضافه کنید.)
وقتی فرآیند فرزند توسط والد تشکیل می‌شود، این فرزند یک کپی از فضای حافظه‌ای والد را دریافت میکند، که شامل متن، پشته، هرم و بقیه‌ی بخش‌های ممکن می‌باشد. اگر فضاهای حافظه‌ای فرآیندها مجزا باشند، اجرا شدن یک فرآیند روی فضای حافظه‌ای فرآیند دیگر تأثیر ندارد و به اصطلاح فرآیندها مستقل از هم اجرا می‌شوند. اگر این دو فرآیند، حالت اجراییشان هم‌پوشانی نداشته باشد، آنگاه رقابت برای منابعی مانند CPU و RAM کاهش می‌یابد. و چون فضای حافظه‌ای متفاوت است، فرآیند والد و فرزند، به طور مستقیم به متغیرها و داده‌های یکدیگر دسترسی ندارند و باید از روش‌های ارتباطی غیرمستقیم (IPC) استفاده کنند.

دو نکته مهم:

اگر یک فرآیند فرزند خاتمه یابد، آنگاه تبدیل به یک فرآیند زامبی می‌شود تا وقتی که فرآیند پدر از شرایط فرزندش (معمولاً توسط $wait()$) باخبر شود. اگر فرآیند والد از دستور $wait()$ استفاده نکند، فرآیند فرزند زامبی خواهد ماند و یک بخش کوچکی از حافظه را آشغال خواهد کرد.
اگر فرآیند پدر زودتر خاتمه یابد، فرآیند فرزند یتیم خواهد شد و تا وقتی که سیستم‌عامل خاتمه‌اش ندهد یتیم خواهد ماند.

ویژگی	مدلِ فضای حافظه‌ای اشتراکی و همروند	مدلِ فضای حافظه‌ای مجزا و غیرهمروند
دسترسیِ حافظه و همگام‌سازی	فرآیند والد و فرزند به متغیرهای یکدیگر دسترسی مستقیم دارند. یعنی نیاز به روش‌های IPC نداریم. این ویژگی باعث افزایش سرعت ردوبدل شدن داده بین این دو فرآیند است. البته از طرفی به دلیل دسترسی به سلول‌های حافظه‌ای یکسان، همگام‌سازی نخ اجرایی یک نقش کلیدی بازی خواهد کرد.	نیازی به همگام‌سازی در اجرای فرآیندها نیست و انتقال داده در صورت لزوم به صورت غیرمستقیم صورت می‌گیرد.
عملکرد	اگر فرکانس انتقال بالا باشد، این روش بسیار می‌تواند مؤثر باشد، از این نظر که انتقال داده مستقیم است و همچنین عملیات تعویض متن ($context\ switch$) در این روش سریع‌تر است. زیرا منابع به صورت اشتراکی مورد استفاده قرار می‌گیرد و CPU نیاز به دوباره بارگذاری‌کردن منابع ندارد.	برای فرآیندهای که اهداف نسبتاً متفاوت و غیرموازی دارند میتواند گزینه‌ی مناسبی باشد. نکته‌ی مثبت در این حالت این است که همگام‌سازیِ فرآیندها، که در این مورد خاص به دلیل متفاوت بودن اهداف فرآیندها یک اضافه‌کاری (overhead) است، انجام نمیشود

امنیت و یکپارچگی داده‌ها	اختلال در فرآیند همگام‌سازی میتواند به مشکلاتی در این روش منجر شود. مثلاً از بین رفتن داده‌ها یا شرایط رقابتی که عمل‌کرد اجرای دستورات را پایین می‌آورد و حتی می‌تواند منجر به رقابت‌های بی‌پایان برای دسترسی به یک منبع خاص شود.	به دلیل مجزا بودن فضای حافظه‌ای، امکان بروز خطاهای رخ داده در روش دیگر وجود ندارد.
استفاده	معمولاً در برنامه‌های چندرشته‌ای، که فرآیندهای به هم وابسته‌اند و نیاز به انتقال بی‌درنگ و سریع داده‌ها است. مثال: web servers real time applications	وقتی فرآیندهای کارهای متفاوتی را انجام می‌دهند و وظایف مستقلی دارند. مثال: database management systems system-level operations

تمرین ۶

فرض کنید دو فرآیند p۱ و p۲ به صورت همروند وجود دارند. در این صورت کدام یک از موارد زیر نمی تواند خروجی اجرای این دو پردازه باشد؟ (پاسختان را شرح دهید)

(آ) $C^*A^*B^*D$

از آنجایی که D تنها یکبار چاپ شده‌است و C به تعدادی بیشتری چاپ شده‌است درحالی که هر دو در یک حلقه بوده‌اند پس این گزینه اشتباه است.

(ب) $(AB)^*(CD)^*$

این گزینه درست است زیرا حلقه اول اجرا شده و پس از آن حلقه دوم اجرا شده است.

(ج) $A(CD)^*B$

در این گزینه ابتدا A چاپ شده است سپس حلقه برنامه دوم و پس از پایان آن، ادامه حلقه اول که B است چاپ شده است پس میتواند خروجی برنامه باشد

(د) $BCDA$

این گزینه نیز نمی‌تواند خروجی این برنامه باشد زیرا B قبل از A چاپ شده است درحالی که در حلقه برعکس است.

تمرین ۷

فرض کنید در یک سیستم کامپیوتری سازوکار وقفه وجود ندارد، در این صورت به سوالات زیر پاسخ کامل بدهید

(آ) چه چالش هایی برای سیستم به وجود خواهد آمد؟
در این سیستم کامپیوتری، CPU همواره همه‌ی دستگاه‌ها را بررسی می‌کند تا درصورت نیاز به آنها رسیدگی کند. در این سیستم وقت CPU برای بررسی دستگاه‌ها استفاده میشود که به معنای تلف شدن این زمان است و اگر تعداد دستگاه‌ها زیاد باشد یا اکثر مواقع بیکار باشند این موضوع بسیار بیشتر خواهد بود. این موضوع میتواند باعث تاخیر در اجرای فرآیندهای مهم شود که احتیاج به اجرای آنی دارند و شاید این تاخیر باعث از دست رفتن فرآیندهای شود که باید در زمان کوتاهی اجرا شوند و CPU نتواند آنها را اجرا کند.

(ب) چنین سیستمی چه مزیت هایی خواهد داشت؟
این سیستم به نسبت سیستمی که از وقفه پشتیبانی میکند ساده‌تر است و پیاده‌سازی آن راحت‌تر و احتمالا کم‌هزینه‌تر است به علاوه به دلیل اینکه فاصله میان واکشی‌ها مشخص است پس زمانبندی این سیستم دقیق‌تر و پیش‌بینی پذیر است که در بعضی از دستگاه‌ها مورد نیاز است.

(ج) فرض کنید همان سیستم، مشغول یک عمل I/O است و همچنین فرآیندی بر روی آن در حال اجرا است که می دانیم عمل I/O پیش از این فرآیند به اتمام می رسد. اکنون برای مدیریت این سیستم سازوکاری ارائه دهید و اجرای آن روی فرآیند و عملیات موجود را گام به گام شرح دهید.
در این سیستم به دلیل وجود نداشتن وقفه، CPU باید در زمان‌های مشخص عملیات واکشی از دستگاه I/O را انجام دهد. اگر کار این دستگاه تمام شده باشد، به آن رسیدگی و عملیات را خاتمه میدهد و در غیر اینصورت اجرای فرآیند را از سرمیگیرد. البته میتوانیم یک صف نیز برای عملیات‌ها I/O یا به صورت کلی وقفه‌ها نیز ایجاد کنیم که CPU به جای بررسی هریک از اجزا این صف را بررسی و دستورالعمل‌های موردنیاز را انجام دهد. اجرای این سیستم روی فرآیند و عملیات به صورت زیر خواهد بود:

۱- اجرای فرآیند تا رسیدن به زمان واکشی وضعیت‌ها یا اتمام فرآیند

۲- اگر عملیات I/O به اتمام رسیده‌باشد، به آن خاتمه میدهد و منابع را آزاد میکند، در غیر اینصورت اجرای فرآیند را از سر می‌گیرد