

سیستم‌های عامل
دکتر زرندی



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری سوم

۲۶ مهر ۱۴۰۳

سوال اول

به سوالات زیر در رابطه با محیط‌های محاسباتی (Computing environment) پاسخ دهید.

۱. مدل‌های Client-server و Peer to peer را تعریف و با یکدیگر مقایسه کنید.

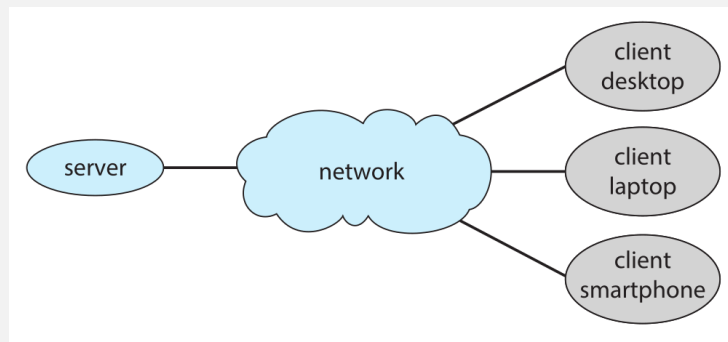
پاسخ

(آ) معماری Client-server:

در این مدل، سیستم به دو بخش اصلی تقسیم می‌شود: کلاینت (Client) و سرور (Server). سرور یک سیستم قدرتمند است که خدمات یا منابع خاصی مانند پایگاه داده، فایل‌ها، یا اپلیکیشن‌ها را ارائه می‌دهد و کلاینت‌ها دستگاه‌هایی هستند که از این منابع استفاده می‌کنند. ارتباط معمولاً به این شکل است که کلاینت‌ها درخواست‌هایی به سرور می‌فرستند و سرور پاسخ می‌دهد. برای مثال زمانی که ما از مرورگر خود به وب‌سایتی دسترسی پیدا می‌کنیم، مرورگر به عنوان کلاینت و وب‌سرور به عنوان سرور عمل می‌کند.

مطابق با توضیحات صفحه ۴۳ کتاب Silberschatz می‌توان گفت معماری شبکه‌های امروزی معمولاً به این شکل است. به این مدل از سیستم‌های توزیع‌شده، سیستم کلاینت-سرور می‌گویند. همچنین سرورها به دو دسته تقسیم می‌شوند: سرورهای محاسباتی و سرورهای فایل. که چون در صورت سوال توضیحات سرور ها خواسته نشده است، توضیحات آن را نمی‌نویسیم.

شکل زیر که از کتاب Silberschatz آورده شده است، ساختار مدل کلاینت-سرور را نشان می‌دهد.

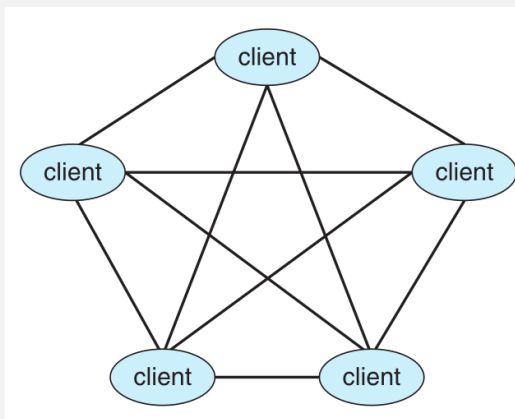


شکل ۱: ساختار سیستم‌هایی با مدل کلاینت-سروری

(ب) معماری Peer to peer:

مدل دیگری که برای سیستم‌های توزیع‌شده وجود دارد، مدل P2P است. در این مدل، تفاوتی بین کلاینت‌ها و سرورها وجود ندارد یعنی هر گره برحسب خدمتی که ارائه می‌دهد، می‌تواند هم به عنوان کلاینت و هم به عنوان سرور عمل کند. شکل زیر نمونه‌ای از این معماری است:

پاسخ



شکل ۲: ساختار سیستم‌هایی با مدل P2P

سیستم‌های P2P نسبت به سیستم‌های کلاینت-سرور مزیتی دارند. در سیستم کلاینت-سرور، سرور می‌تواند یک گلوگاه باشد؛ اما در سیستم P2P خدمات توسط چندین گره که در شبکه توزیع شده‌اند ارائه می‌شوند. همچنین Scaleability در مدل P2P بهتر است، چون با افزایش تعداد دستگاه‌ها، شبکه قوی‌تر می‌شود. در حالی که در مدل Client-server، با افزایش تعداد کلاینت‌ها، سرور ممکن است تحت فشار قرار گیرد. از نظر امنیت مدل کلاینت سرور راحت‌تر مدیریت می‌شود چون یک سرور مرکزی می‌تواند کنترل کامل بر روی داده‌ها و دسترسی داشته باشد، در حالی که در مدل P2P امنیت پیچیده‌تر است چون داده‌ها در میان بسیاری از دستگاه‌ها پخش می‌شوند.

اسکایپ نمونه‌ای از مدل P2P است.

۲. Virtualization و Emulation را تعریف کنید و تفاوت‌های آن‌ها را ذکر کنید.

پاسخ

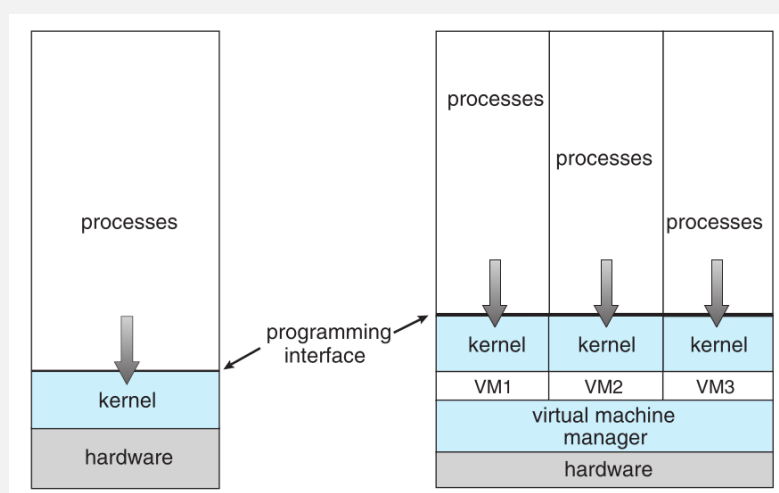
مطابق با تعریف آقای Silberschatz در صفحه ۳۴ کتابشان، می‌توان گفت که: مجازی‌سازی یا Virtualization یک فناوری است که به ما این امکان را می‌دهد تا سخت‌افزار یک کامپیوتر (مانند پردازنده، حافظه، دیسک‌های سخت، کارت‌های شبکه و غیره) را به چندین محیط اجرایی مختلف تجزیه کنیم و به این ترتیب، این حس برای کاربر ایجاد شود که هر محیط به‌صورت مجزا روی یک کامپیوتر اختصاصی خود در حال اجرا است. این محیط‌ها می‌توانند به‌عنوان سیستم‌عامل‌های مختلف (مثل ویندوز و یونیکس) در نظر گرفته شوند که ممکن است هم‌زمان اجرا شوند و با هم تعامل داشته باشند. کاربر یک ماشین مجازی می‌تواند بین این سیستم‌عامل‌های مختلف به همان شکلی که بین فرآیندهای مختلف در یک سیستم‌عامل واحد جابه‌جا می‌شود، جابه‌جا شود.

مجازی‌سازی به سیستم‌عامل‌ها این امکان را می‌دهد که به‌عنوان برنامه‌هایی در درون دیگر سیستم‌عامل‌ها اجرا شوند. در نگاه اول، ممکن است این قابلیت چندان کاربردی به نظر نرسد، اما صنعت مجازی‌سازی بسیار گسترده و در حال رشد است که نشان‌دهنده اهمیت و کاربرد فراوان آن است.

به‌طور کلی، نرم‌افزارهای مجازی‌سازی یکی از اعضای گروهی از نرم‌افزارها هستند که شبیه‌سازی (Emulation) نیز در آن قرار می‌گیرد. Emulation به معنای شبیه‌سازی سخت‌افزار کامپیوتر در نرم‌افزار است و معمولاً زمانی استفاده می‌شود که نوع پردازنده مبدأ با پردازنده هدف متفاوت باشد. به عنوان مثال، زمانی که شرکت اپل پردازنده‌های IBM Power خود را به پردازنده Intel x86 برای کامپیوترهای دسکتاپ و لپ‌تاپ خود تغییر داد یک قابلیت شبیه‌سازی به نام Rosetta ارائه داد که به برنامه‌هایی که برای پردازنده IBM نوشته شده بودند، اجازه می‌داد روی پردازنده Intel اجرا شوند.

پاسخ

این مفهوم می‌تواند توسعه یابد تا به یک سیستم‌عامل کامل که برای یک پلتفرم خاص نوشته شده، اجازه دهد روی یک پلتفرم دیگر اجرا شود. با این حال، شبیه‌سازی هزینه بالایی دارد، زیرا هر دستورالعمل در سطح ماشین که به‌طور بومی روی سیستم مبدأ اجرا می‌شود، باید به دستورالعمل معادل در سیستم هدف ترجمه شود، که اغلب منجر به چندین دستورالعمل در سیستم هدف می‌شود. اگر پردازنده‌های مبدأ و هدف سطح عملکرد مشابهی داشته باشند، ممکن است کد شبیه‌سازی شده بسیار کندتر از کد بومی اجرا شود. شکل زیر نمونه ای از یک کامپیوتر معمولی و یک کامپیوتر مجازی شده را نشان می‌دهد:



شکل ۳: ساختار یک ۳ عدد VM

۳. سه نمونه از دسته سرویس‌های ابری را نام ببرید و به صورت مختصر توضیح دهید.

پاسخ

(آ) Software as a service (SaaS):

یک یا چند برنامه (مانند پردازشگرهای کلمه یا spreadsheets) که از طریق اینترنت در دسترس هستند.

(ب) Platform as a service (PaaS):

یک پشته نرم‌افزاری آماده برای استفاده از طریق اینترنت (به عنوان مثال، یک سرور پایگاه داده).

(ج) Infrastructure as a service (IaaS):

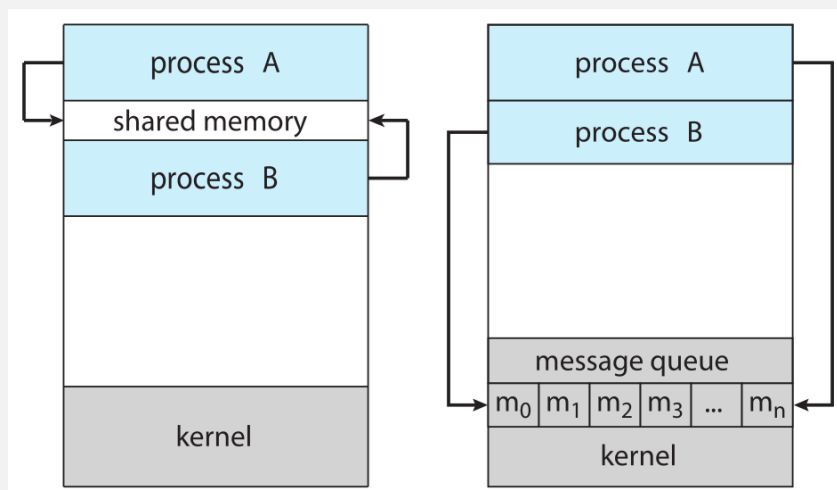
سرورها یا فضای ذخیره‌سازی که از طریق اینترنت قابل دسترسی هستند (برای مثال، فضای ذخیره‌سازی برای تهیه نسخه‌های پشتیبان از داده‌های تولید).

سوال دوم

دو روش برای ارتباط میان فرایندها Message passing و Shared memory است. آن‌ها را تعریف و با یکدیگر مقایسه کنید.

پاسخ

در مدل Message passing یک بخش از حافظه که بین فرایندهای همکار به اشتراک گذاشته می‌شود، ایجاد می‌گردد. سپس فرایندها می‌توانند با خواندن و نوشتن داده‌ها در این بخش مشترک، اطلاعات را مبادله کنند. در مدل Message passing، ارتباط از طریق پیام‌هایی که بین فرایندهای همکار رد و بدل می‌شود، انجام می‌گیرد. تفاوت این دو مدل ارتباطی در شکل زیر نشان داده شده است.



شکل ۴: مدل‌های برقراری ارتباط میان فرایندها

هر دو مدل ذکر شده در سیستم‌عامل‌ها رایج هستند و بسیاری از سیستم‌ها هر دو را پیاده‌سازی می‌کنند. Message passing برای تبادل مقادیر کوچکتر داده‌ها مفید است، زیرا نیازی به جلوگیری از تداخلات نیست. Shared memory می‌تواند همچنین در سیستم‌های توزیع‌شده نسبت به حافظه مشترک آسان‌تر پیاده‌سازی می‌شود. Message passing سریع‌تر از پیام‌رسانی باشد، زیرا سیستم‌های Message passing معمولاً از طریق فراخوانی‌های سیستمی پیاده‌سازی می‌شوند و بنابراین نیازمند مداخله هسته هستند که زمان‌بر است. در سیستم‌های Shared memory فراخوانی سیستمی تنها برای ایجاد بخش‌های حافظه مشترک لازم است. پس از ایجاد حافظه مشترک، تمام دسترسی‌ها مانند دسترسی‌های معمولی به حافظه انجام می‌شود و نیازی به کمک هسته نیست.

سوال سوم

از میان عملیات‌هایی که نیاز به System Call دارند ۳ مثال نام ببرید و توضیح دهید که اگر هر عملیات در لایه User انجام نشود چه مشکل‌هایی را می‌تواند به وجود بیاورد.

پاسخ

۱. عملیات خواندن و نوشتن فایل‌ها (File Read/Write)

برای دسترسی به سیستم فایل و خواندن یا نوشتن داده‌ها در فایل‌ها، نیاز به System Call داریم. دسترسی مستقیم کاربران به هارد دیسک یا سیستم فایل از طریق User mode بدون استفاده از System Call بسیار خطرناک است. زیر می‌تواند مشکلات زیر را ایجاد کند:

- عدم کنترل دسترسی: اگر User ها به صورت مستقیم و بدون استفاده از سیستم‌عامل به فایل‌ها دسترسی پیدا کنند، امنیت فایل‌ها به خطر می‌افتد و احتمال خرابی داده‌ها وجود دارد.
- مدیریت ضعیف منابع: بدون واسطه‌ای مانند System Call امکان مدیریت مناسب منابع و جلوگیری از استفاده نادرست یا بیش‌ازحد منابع وجود نخواهد داشت.

۲. تخصیص و آزادسازی حافظه (Memory Allocation/Deallocation)

فرآیند تخصیص حافظه به برنامه‌ها توسط System Call‌هایی مانند malloc یا new انجام می‌شود. سیستم‌عامل کنترل می‌کند که چه بخشی از حافظه به برنامه اختصاص داده شود و چه زمانی حافظه باید آزاد شود. مشکلات این دسته به‌صورت زیر عنوان می‌شود:

- خرابی یا دسترسی غیرمجاز به حافظه: اگر کاربر مستقیماً به حافظه دسترسی پیدا کند، ممکن است به بخش‌هایی از حافظه دسترسی داشته باشد که به برنامه‌های دیگر اختصاص داده شده‌اند، که این می‌تواند منجر به خرابی سیستم یا دسترسی غیرمجاز به اطلاعات شود.
- مدیریت نادرست حافظه: بدون System Call، امکان نشت حافظه (memory leak) و استفاده بی‌رویه از منابع حافظه وجود دارد، چرا که سیستم‌عامل نمی‌تواند حافظه‌ای که دیگر مورد استفاده نیست را آزاد کند.

۳. اجرای فرآیند جدید (Process Creation)

عملیات ایجاد یک فرآیند جدید توسط سیستم‌عامل انجام می‌شود که از طریق System Call‌هایی مانند fork یا exec در سیستم‌عامل‌های یونیکسی انجام می‌شود. این System Call‌ها امکان اجرای برنامه‌های جدید در محیط سیستم‌عامل را فراهم می‌کنند. مشکلات این مثال به صورت زیر معرفی می‌شود:

- تداخل در مدیریت فرآیندها: اگر کاربران بتوانند به‌طور مستقیم فرآیند جدید ایجاد کنند، سیستم‌عامل نمی‌تواند به درستی فرآیندها را زمان‌بندی کند و منابع را به طور عادلانه بین آن‌ها توزیع نماید.
- نقض امنیت: ایجاد فرآیندهای جدید بدون کنترل سیستم‌عامل می‌تواند منجر به اجرای کدهای مخرب یا دسترسی‌های غیرمجاز به منابع سیستم شود، که امنیت کل سیستم را به خطر می‌اندازد.

سوال چهارم

انواع مدل‌های طراحی سیستم‌های عامل را نام ببرید و به صورت مختصر ساختار آن‌ها را نیز توضیح دهید.

پاسخ

سوال پنجم

به سوالات زیر در رابطه با مدل‌های سیستم‌های عامل پاسخ دهید.

۱. سیستم‌های عامل اولیه از چه مدلی پیروی می‌کردند و دو مورد از معایب این مدل را توضیح دهید.

پاسخ

۲. فواید کاهش ساینز kernel (در مدل Microkernel) چیست؟ سه نمونه از عملیات‌هایی که در Kernel ننگه داشته می‌شوند را نام ببرید.

پاسخ

۳. تفاوت میان مدل لایه ای و مدل ماژولار چیست و چه عاملی باعث برتری مدل ماژولار می‌شود؟

پاسخ