



1. توضیح دهید چگونه تعامل میان File-system manipulation، I/O operations و Resource allocation می‌تواند بر عملکرد کلی سیستم تأثیر بگذارد (برای پاسخ به نقش هرکدام در کارایی سیستم اشاره کنید).
2. فرض کنید در یک سیستم چندکاربره خطایی در مازول Error Detection رخ دهد. تحلیل کنید که این خطا چگونه ممکن است به نقض Protection and Security منجر شود.
3. اگر سیستم عامل فاقد مازول Accounting Logging باشد، چه پیامدهایی برای مدیریت منابع و امنیت خواهد داشت؟ (از دیدگاه مدیریتی و فنی را بررسی کنید).
4. تفاوت‌های طراحی بین رابط‌های کاربری CLI، GUI و Touch Screen را از نظر نوع کاربر هدف، بار شناختی و ارتباط با سیستم توضیح دهید.
5. انواع روش‌های پیاده‌سازی shell را نام برده و توضیح دهید. در پیاده‌سازی پروژه‌های آزمایشگاه شما کدام روش را در پیش گرفته‌اید؟
6. ارتباط بین System Call Interface، API و Kernel را به صورت تحلیلی توضیح دهید. چرا جداسازی این لایه‌ها برای پایداری سیستم حیاتی است؟
7. کامپایلرها نظیر gcc و msvc هر دو پارامترها را به شکل متفاوتی پاس می‌دهند، توضیح دهید هر کدام معمولاً از چه روشی استفاده می‌کنند؟ (برای راحتی می‌توانید از godbolt.org استفاده نمایید).
8. مقایسه کنید که چگونه نحوه Passing Parameters to System Calls می‌تواند روی امنیت و کارایی سیستم عامل تأثیر بگذارد.
9. تحلیل کنید که چرا مدل ارتباطی Shared Memory با وجود سرعت بالا، به لحاظ مفهومی پیچیده‌تر از Message Passing است؟ مثال بزنید که در چه شرایطی هرکدام مناسب‌تر است.
10. آیا در سیستم عامل‌های Distributed نیز می‌توان از Shared Memory استفاده کرد؟ اگر بله، چگونه؟ آیا واقعاً همان مفهوم حافظه مشترک محلی را دارد یا نوعی شبیه‌سازی است؟
11. در بحث Protection، چگونه System Calls می‌توانند به صورت غیرمستقیم در افزایش امنیت شبکه موثر باشند؟

12. مقایسه کنید که چگونه System Service و System Calls مکمل یکدیگرند و نبود هر کدام چه تاثیری بر تجربه کاربر و توسعه‌دهنده دارد.
13. تابع printf در زبان C مستقیماً یک System Call نیست و از طریق لایه‌ی کتابخانه استاندارد عمل می‌کند؛ توضیح دهید چرا این لایه‌ی واسط بین برنامه و System Call طراحی شده و چه مزیتی نسبت به فراخوانی مستقیم System Call دارد؟
14. در این فصل سه API اصلی شامل Windows API، POSIX API و Java API معرفی شده‌اند. با توجه به مثال‌های موجود، ساختار و نام‌گذاری توابع در ویندوز و لینوکس را مقایسه کنید و توضیح دهید به نظر شما کدام سیستم در طراحی API‌ها بهتر عمل کرده است و چرا؟
15. بررسی کنید چگونه مفهوم User View از سیستم‌عامل در اثر تنوع System Services ممکن است با Actual System Structure تفاوت داشته باشد؟
16. تفاوت میان Linking ایستا و پویا را توضیح داده و تحلیل کنید چگونه Dynamic Linking می‌تواند هم کارایی و هم خطرات امنیتی را افزایش دهد.
17. با توجه به نقش Loader در فرآیند اجرا، توضیح دهید چگونه این جزء می‌تواند در امنیت اجرای برنامه‌ها نقشی حیاتی داشته باشد.
18. در سیستم‌های چند پردازشی، چرا Loader باید از مکانیزم Relocation استفاده کند؟
19. زنجیره‌ی تبدیل «Source Code → Executable → Running Process» را از منظر تعامل بین Linker ، Loader و Kernel تحلیل کنید. نقش هرکدام در مدیریت حافظه و آدرس‌دهی را توضیح دهید.
20. توضیح دهید چگونه ترکیب نادرست System Call هایی create و delete در یک سیستم چند کاربره می‌تواند باعث نشت منابع (Resource Leak) شود.
21. سه راه اجرا و در دسترس نمودن یک نرم‌افزار بر روی سیستم‌عامل‌های مختلف را نام ببرید و توضیح دهید.
22. مفهوم Application Binary Interface چیست و آیا می‌تواند مشکل cross-platform compatability را به طور کامل حل کند؟ توضیح دهید.
23. اهداف سطح بالای طراحی سیستم‌عامل را بیان کنید. برای اهداف کاربر و اهداف سیستم در سطح پایین‌تر چند مثال ارائه دهید.
24. دو مفهوم mechanism و policy را شرح دهید و آن‌ها را با استفاده از یک مثال از یک‌دیگر تفکیک کنید. توضیح دهید چگونه تفکیک این دو مفهوم منجر به انعطاف‌پذیری سیستم می‌شود.
25. معماری monolithic را تشریح کنید، مزایا و معایب آن را بررسی کنید و یک نمونه سیستم‌عامل با این معماری مثال بزنید.

26. طراحی لایه‌ای سیستم‌عامل را توضیح دهید و بیان کنید استفاده از این طراحی چه فواید و معایبی به همراه دارد.
27. در مورد معماری microkernel به سوالات زیر پاسخ دهید:
- a. ساختار کلی این معماری را تشریح کنید. چه وظایفی در این معماری همچنان در kernel mode قرار می‌گیرند؟ چند نمونه سیستم‌عامل با این طراحی را نام ببرید.
- b. کارکرد اصلی کرنل در این معماری چیست و به چه طریق انجام می‌پذیرد؟
- c. مزایا و معایب استفاده از این طراحی را شرح دهید.
- d. این طراحی را با طراحی monolithic به طور کلی مقایسه کنید.
28. طراحی ماژولار سیستم‌عامل و مفهوم LKM را شرح دهید و توضیح دهید چگونه این معماری از مزایای دو روش layered و microkernel استفاده می‌کند.
29. ساختار hybrid یک سیستم‌عامل را شرح و به طور مختصر در مورد چگونگی hybrid بودن سیستم‌عامل‌های Windows و Linux توضیح دهید.
30. در مورد سیستم‌عامل Darwin به سوالات زیر پاسخ دهید:
- a. ساختار و معماری این سیستم‌عامل را شرح دهید.
- b. مفاهیم Kernel Abstraction و Kernel Extension را در این سیستم‌عامل شرح دهید.
- c. توضیح دهید در این سیستم‌عامل از چه راه‌حلهایی برای کم‌رنگ‌تر کردن معایب معماری microkernel استفاده شده است.
31. ساختار، اجزا و طراحی سیستم‌عامل به کار رفته در MacOS/iOS و Android را شرح دهید و با یک‌دیگر مقایسه کنید.
32. سه روش ممکن برای پیاده‌سازی و پیکربندی سیستم‌عامل را شرح دهید و آن‌ها را از نظر اندازه، تعمیم‌پذیری و انعطاف‌پذیری در برابر تغییرات مقایسه کنید.
33. چهار مرحله پروسه Boot سیستم‌عامل‌ها را به طور کلی بیان کنید؛ همچنین پروسه بوت شدن سیستم‌عامل لینوکس را به طور کامل شرح دهید.
34. BIOS و UEFI هر دو نقش مهمی در فرآیند بوت دارند، به‌طور خلاصه بگویید مهم‌ترین مزیت UEFI نسبت به BIOS چیست؟
35. وقتی کرنل در اجرا با شکست روبه‌رو می‌شود، آن را Crash می‌نامیم. با توجه به پیچیده بودن دیباگ کردن کرنل، در مواقع مواجهه با کرش، از چه ساز و کاری برای یافتن مشکل استفاده می‌شود؟

36. Counting و Tracing دو دسته روش مانیتور کردن عملکرد سیستم می‌باشند. آن‌ها را شرح دهید و با

یک‌دیگر مقایسه کنید و برای هر دو حالت per-process و system-wide آن‌ها یک مثال بیاورید.

37. در پردازنده‌ها قابلیت به نام single step وجود دارد که اجرای دستور به‌دستور را ممکن می‌کند؛ این ویژگی

بیشتر در چه شرایطی کاربرد دارد؟

38. در مورد Virtual Machine به سوالات زیر پاسخ دهید:

a. در مورد ساختار کلی آن توضیح دهید.

b. در سیستم‌های Virtual Machine آیا هر یک از processها، از hardwareهای جدا استفاده می‌کنند؟

آیا این processها به Hardwareهای یک‌دیگر دسترسی دارند؟

c. در Virtual Machine آیا امکان این را داریم که برای هر process سیستم‌عامل جداگانه فراهم کنیم؟

چه برنامه‌ای این جداسازی را برای ما فراهم می‌کند؟

d. معایب و مزایای VM را بیان کنید. توضیح دهید چگونه استفاده از VM می‌تواند سبب تفاوت در زمان

اجرا شود.