سوال ۱

یک مدل از hardware support که اطمینان میدهد که حملات buffer overflow اتفاق نیوفتند این است که از اجرای کدی که در بخش stack از فضای آدرس پردازه قرار دارد ، جلوگیری کند.
میدانیم که حملات buffer overflow با سر ریز کردن buffer در داخل stack frame به وقوع میپیوندد. درواقع با overwriteکردن آدرس برگشتی یک تابع و سپس در نتیجه‌ی آن پریدن به قسمت دیگری از stack frame که برنامه بدخواه و خراب‌کار در آن قرار دارد که در نتیجه‌ی buffer overflow در آنجا قرار گرفته، به وقوع میپیوندند. با جلوگیری از اجرای کد در stack segment این مشکل حل میشود.
رویکرد دیگری که از یک better programming methodology استفاده میکنند به صورت معمول با توجه به استفاده‌ی bounds-checking ساخته میشوند تا در برابر buffer overflow محافظت کنند.
Buffer overflow معمولا در زبان‌هایی مثل جاوا که هر دسترسی آرایه‌ای تضمین میشود در محدوده‌ای که نرم افزار میتواند چک کند است، رخ نمیدهد. این ویکردها به پشتیبانی سخت افزاری نیاز ندارند ولی خب هزینه‌ی اجرایی دارد که مرتبط با اجرای bounds-checkingمیباشد.

سوال ۲

در ابتدا برای دسترسی به TLB به ۱۰ نانوثانیه زمان نیاز داریم.
اگر TLB Miss رخ دهد در این صورت باید ابتدا از حافظه و از Page Table آدرس فیزیکی را لود کنیم که خود ۱۰۰ نانوثانیه زمان نیاز دارد. سپس دسترسی به این آدرس به ۱۰۰ نانوثانیه دیگر زمان نیاز دارد.
اگر hit رخ دهد در این صورت فقط به یک ۱۰۰ نانوثانیه دیگر زمان نیاز داریم.
EAT = 0.95*(10 + 100) + 0.05*(100 + 100 + 10) = 115 ns

سوال ۳

اولین مزیت watchdog این است که یک سیستم مرکزی (centralized) است و زمانی که برای تمامی فایل ها اعمال شود اگر بخواهیم تغییری در آن ایجاد کنیم به راحتی قابل انجام است و نیازی به فعال سازی دوباره برای تک به تک فایل ها نیست.
مزیت دوم آن این است که چون اولین برنامه ایی است که برای دسترسی به فایل با آن مواجه میشویم, اگر بتوان اطمینان حاصل کرد که تکنیک های به کار رفته در آن قابل اعتماد هستند این برنامه میتواند گارانتی بکند که تمامی فایل های ما در وضعیت امن قرار خواهند داشت.

اولین عیب watchdog دقیقا به مزیت دوم بازمیگردد. روی طرف دیگر اولین مکانیزم امنیتی بودن این است که این مکانیزم تبدیل به bottleneck میشود و هر دسترسی باید از این مکانیزم عبور کند که میتواند سربار اضافی داشته باشد و برای هر بار دسترسی به فایل باید از این گلوگاه عبور کرد.
دومین و اصلی ترین عیب آن این است که اگر این برنامه و تکنیک هایش به درستی پیاده سازی نشده باشند و دارای حفره امنیتی باشند هیچ سیستم پشتیبانی برای file protection وجود ندارد و این مشکل امنیتی به دلیل مرکزی بودن watchdog برای تمامی فایل هایی که با این سیستم امن شده اند وجود خواهد داشت.

سوال ۴

در ابتدا حافظه مانند شکل زیر است.

2KB
7KB
5KB
8KB
30KB
10KB
10KB

اگر از first fit برای تخصیص حافظه به برنامه‌ها استفاده کنیم، داریم:

2KB
7KB
5KB
Program B
Program A 5KB
Program C 1KB
10KB

اگر از best fit برای تخصیص حافظه به برنامه‌ها استفاده کنیم، داریم:

2KB
7KB
5KB
Program B
Program A 5KB
Program C 1KB
10KB

اگر از worst fit برای تخصیص حافظه به برنامه‌ها استفاده کنیم، داریم:

2KB
7KB
5KB
8KB
Program A 5KB
Program B 2KB
Program C 1KB

سوال ۵

الف) نصب یک پردازنده سریعتر باعث می‌شود بهره‌وری سیستم کاهش یابد، زیرا کارها در زمان کمتری روی پردازنده قرار می‌گیرند.
ب) نصب یک دیسک بزرگتر برای paging فضای ذخیره سازی pageها در حافظه را افزایش می‌دهد و از بهره‌وری paging disk می‌کاهد.
ج) افزایش اندازه‌ی multiprogramming سیستم تعداد page faultها را افزایش می‌دهد، بنابراین از بهره‌وری CPU می‌کاهد.
د) کاهش اندازه‌ی multiprogramming سیستم تعداد page faultها را کاهش می‌دهد، بنابراین به بهره‌وری CPU می‌افزاید.
هـ) افزایش حافظه‌ی main memory از تعداد page faultها کاسته و بهره‌وری CPU را افزایش می‌دهد.
و) نصب یک دیسک سریعتر زمان page swapها را کاهش داده و به بهره‌وری پردازنده می‌افزاید.
ز) افزایش page size از تعداد page faultها کاسته و به زمان page swapها می‌افزاید، بنابراین با توجه به شرایط می‌تواند باعث افزایش یا کاهش بهره‌وری پردازنده شود.