

پاسخنامه تمرین سوم درس سیستم‌های عامل

دکتر زرندی

پاییز ۱۴۰۰

۱- همانطور که می‌دانید توابع مربوط به رابط‌های برنامه‌نویسی (*API*) خود از فراخوانی‌های سیستم (*system calls*) استفاده می‌کنند. به‌طور مثال تابع *createprocess()* که برای ساخت یک فرایند استفاده می‌شود، خود *NTCreateProcess()* را فراخوانی می‌کند که یک فراخوانی سیستم است. توضیح دهید که یک برنامه‌نویس چرا باید رابط‌های برنامه‌نویسی را به استفاده مستقیم از فراخوانی‌های سیستم ترجیح دهد.

برنامه‌نویسانی که از رابط‌های برنامه‌نویسی استفاده می‌کنند، می‌توانند برنامه خود را بر روی هر سیستمی که از آن رابط برنامه‌نویسی را پشتیبانی می‌کند، کامپایل و اجرا کنند و لازم نیست نگران تفاوت‌های معماری آن سیستم‌ها باشند.

همچنین فراخوانی‌های سیستم معمولاً جزئیات فراوانی دارند و کار کردن با آن‌ها نسبت به یک رابط برنامه‌نویسی بسیار سخت‌تر است.

۲- در مورد ساختارهای سیستم‌عامل به سوالات زیر پاسخ دهید.

الف) معایب ساختار میکروکرنل که باعث کاهش عملکرد (*performance*) در سیستم عامل های مبتنی بر این نوع معماری می‌شود را شرح دهید؟ در کرنل *Mach* چگونه این مشکل حل شده است؟

معماری میکروکرنل می‌تواند به دلیل سربار زیاد توابع سیستمی باعث کاهش کارایی شود. هنگامی که دو سرویس سطح کاربر (*user-level services*) می‌خواهند با هم ارتباط برقرار کنند، پیام‌ها باید بین آن‌ها کپی شود که در فضاهای آدرس جداگانه‌ای قرار دارند. علاوه بر این، سیستم‌عامل ممکن است مجبور شود که از یک فرآیند به فرآیند بعدی جابجا شود تا بتواند پیام‌ها را منتقل کند. سرباری که از کپی کردن پیام‌ها و جابجا شدن بین فرآیندها ناشی می‌شود، بزرگ‌ترین مانع رشد سیستم‌های عامل میکروکرنل بوده است.

برای حل مشکل کارایی در کرنل *Mach*، همه‌ی افزونه‌های کرنل از جمله کیت *I/O* در یک فضای آدرس واحد قرار گرفته‌اند. بنابراین در کرنل *Mach* همچنان زیرسیستم‌ها در فضای کاربر اجرا شده و انتقال پیام نیز انجام می‌شود اما دیگر نیازی به کپی کردن پیام‌ها نیست زیرا همه‌ی سرویس‌ها به یک فضای آدرس یکسان دسترسی دارند.

ب) ساختار ماژولار چه شباهتی به ساختارهای لایه‌ای (*layered approach*) و میکروکرنل دارد و چگونه مشکلات آن‌ها را برطرف کرده است؟

ساختار ماژولار از این لحاظ که هر بخش کرنل یک تعریف مشخص و رابط حفاظت‌شده‌ای دارد، مشابه ساختار لایه‌ای می‌باشد اما از آن انعطاف‌پذیری بیشتری دارد زیرا در ساختار ماژولار بر خلاف ساختار لایه‌ای هر ماژول می‌تواند هر ماژول دیگری را صدا بزند.

این ساختار به ساختار میکروکرنل نیز شبیه است زیرا ماژول اصلی فقط شامل کارکردهای اصلی بوده و می‌داند چگونه ماژول‌های دیگر را بارگذاری کرده و با آن‌ها ارتباط برقرار کند. اما ساختار ماژولار کارایی بیشتری نسبت به ساختار میکروکرنل دارد چون ماژول‌ها برای ارتباط با یکدیگر نیازی به انتقال پیام ندارند.

۳- درباره کرنل NT تحقیق کرده به سوالات زیر پاسخ دهید. (می‌توانید از این [لینک](#) نیز کمک بگیرید).

الف) این کرنل در کدام دسته از ساختارهای سیستم‌عامل قرار می‌گیرد و چرا؟ مایکروسافت چه نام دیگری را برای این نوع سیستم‌عامل استفاده می‌کند؟

کرنل NT یکی از نمونه‌های برجسته کرنل ترکیبی (hybrid) می‌باشد. این کرنل ترکیبی تحت عنوان یک میکروکرنل تغییر یافته (modified microkernel) طراحی شده و از میکروکرنل Mach نیز الهام گرفته است اما از تمامی ویژگی‌های میکروکرنل پیروی نمی‌کند.

کرنل NT یک کرنل ترکیبی محسوب می‌شود زیرا زیرسیستم‌های آن در حالت کاربر اجرا شده و بسیاری از اهداف طراحی آن با اهداف طراحی کرنل Mach مشابهت دارند. از طرف دیگر NT میکروکرنل نیست به دلیل اینکه بیشتر بخش‌های سیستم در فضای آدرس یکسانی با خود کرنل اجرا می‌شوند که با طراحی یکپارچه (monolithic) مطابقت دارد.

مایکروسافت این نوع میکروکرنل تغییر یافته را ماکروکرنل (macrokernel) نیز می‌نامد.

ب) ساختار میکروکرنل دو ویژگی اصلی دارد. توضیح دهید که کرنل *NT* کدام ویژگی را رعایت کرده و کدام را رعایت نکرده است.

سیستم‌های عامل از نوع میکروکرنل بر مبنای دو اصل طراحی می‌شوند. اصل اول ماژولار بودن، کپسوله‌سازی و پنهان کردن داده‌ها می‌باشد. در این جنبه از طراحی، هر قابلیت سیستم عامل فقط به عهده یک بخش از سیستم بوده و همه بخش‌های دیگر از طریق رابط‌های خوش تعریف به این قابلیت دسترسی پیدا می‌کنند. توابع در جای دیگری تکرار نشده و جایگزینی برای آن‌ها وجود ندارد. همچنین تمام دسترسی به ساختمان‌های داده فقط از طریق رابط‌های نرم‌افزاری صورت می‌گیرد. این رویکرد به ما اجازه می‌دهد که یک ماژول را به طور کامل جایگزین کنیم یا ارتقا دهیم بدون اینکه مزاحم سایر بخش‌ها شویم.

اصل دوم در طراحی میکروکرنل به اصل اول مرتبط است اما بیشتر جنبه پیاده‌سازی دارد. این اصل می‌گوید که بخش زیادی از سیستم عامل که به طور سنتی در کرنل و حالت ممتاز (*privileged mode*) اجرا می‌شدند، حالا باید در حالت کاربر اجرا شوند و تنها خود میکروکرنل به همراه کدهای مرتبط با سخت‌افزار در حالت کرنل اجرا می‌شوند.

سیستم‌های عاملی که هر دوی این اصل‌ها را رعایت کنند، میکروکرنل خالص (*pure microkernel*) نامیده می‌شوند. سیستم‌های عاملی که فقط از اصل اول (ماژولار بودن و کپسوله‌سازی) پیروی کنند و اما اصل دوم را رعایت نکنند، کرنل تغییر یافته (*modified microkernel*) یا ماکروکرنل (*macrokernel*) نامیده می‌شوند.

کرنل *NT* جزو دسته دوم بوده و ماکروکرنل محسوب می‌شود. بنابراین فقط اصل اول را رعایت کرده و اصل دوم را رعایت نمی‌کند.