

مقاله تحقیقاتی محدودیت‌های سیستم‌های عامل سنتی را از نظر عملکرد، انعطاف‌پذیری و کارایی برنامه‌ها مورد بحث قرار می‌دهد و معماری سیستم عامل exokernel را به عنوان راه حل پیشنهاد می‌کند. معماری exokernel با ارائه مدیریت منابع فیزیکی از سطح برنامه، این محدودیت‌ها را برطرف می‌کند، امکان سفارشی‌سازی abstractionهای سیستم عامل سنتی را با گسترش، بهبود یا جایگزینی کتابخانه‌ها فراهم می‌کند. مقاله اندازه‌گیری‌هایی را ارائه می‌دهد که برتری سیستم exokernel را نسبت به سیستم‌های عامل سنتی از نظر پیش عملیات‌های کرنل، حافظه مجازی و ارتباط بین فرآیندها نشان می‌دهد.

exokernel به طور امن منابع سخت‌افزاری را به کتابخانه‌های ناامن سیستم‌عامل صادر می‌فرستد، همچنین این امکان را می‌دهد که آن‌ها System objects و Policies را از طریق رابط سطح پایین (low-level interface) پیاده‌سازی کنند و مراقبت از منابع را از مدیریت جدا می‌کند. مقاله نشان می‌دهد که پیش عملیات‌های کرنل در سیستم exokernel ۱۰ تا ۱۰۰ برابر سریع‌تر از سیستم عامل UNIX است. علاوه بر این، پروتوتایپ exokernel حداقل پنج برابر سریع‌تر از پیاده‌سازی‌های حال حاضر برای عملیات‌هایی مانند exception dispatching و ارتباط بین فرآیندها است.

علاوه بر این، مقاله بحث می‌کند که چگونه abstractionهای سیستم عامل قدیمی می‌تواند عملکرد و آزادی پیاده‌سازی برنامه‌ها را محدود کند، و چگونه معماری exokernel اجازه مدیریت منابع در سطح برنامه را می‌دهد. همچنین تأکید می‌کند که دادن کنترل منابع ماشین به برنامه‌ها برای پیاده‌سازی abstractionها در سطح بالاتر نتایج را بهبود می‌بخشد. سپس با استناد به مطالعات قبلی نشان می‌دهد چگونه برنامه‌ها می‌توانند از داشتن کنترل بیشتر بر منابع ماشین استفاده کنند و معیارهایی مانند عملکرد و کارایی را بهبود ببخشند.

مقاله اطلاعاتی را در مورد اصول کلیدی معماری exokernel ارائه می‌دهد، مانند امنیت منابع سخت‌افزاری، افشای names, allocations, و revocation of resources، همچنین جداسازی امنیت منابع از مدیریت. همچنین در مورد پیاده‌سازی و عملکرد Aegis، به عنوان یک exokernel، و ExOS، به عنوان یک کتابخانه سیستمی، بر روی DECstations مبتنی بر MIPS صحبت می‌کند، که در این پیاده‌سازی multiplexing کارا منابع سخت‌افزار، exception dispatching، مدیریت حافظه مجازی، و انتقالات کنترل محافظت شده نشان داده می‌شود. علاوه بر این، مقاله آزمایش‌هایی را ارائه می‌دهد که کارایی IPC abstractions را بر روی ExOS نشان می‌دهد و سیستم عامل سنتی را از نظر زمان ارتباط بین فرآیندها با استفاده از pipeها، حافظه مشترک و ... از منظر سرعت می‌کند.

نظر شخصی :

معماری exokernel که در مقاله تحقیقاتی ارائه شده است، در واقع یک رویکرد نوین برای مقابله با محدودیت های سیستم عامل های قدیمی ارائه می دهد. با این حال، مانند تمام تحقیقات، با مجموعه ای از چالش های جدید روبرو میشویم.

چالش ها

۱. امنیت : در حالی که معماری exokernel حفاظت منابع را از مدیریت جدا می کند، اطمینان از امنیت زمانی که منابع سخت افزاری به کتابخانه های سیستم عاملی ناامن نمایش داده می شود، می تواند یک چالش قابل توجه باشد.

۲. سازگاری : معماری exokernel اجازه می دهد تا abstraction های سیستم عامل سنتی سفارشی سازی شود. با این حال، این قابلیت می تواند منجر به مشکلات سازگاری با نرم افزارهای موجود شود که انتظار دارند abstraction های خاصی را داشته باشد.

۳. پیچیدگی : مدیریت منابع فیزیکی در سطح برنامه می تواند پیچیدگی توسعه برنامه را افزایش دهد، زیرا توسعه دهندگان باید منابعی را مدیریت کنند که معمولاً توسط سیستم عامل اداره می شود.

راه حل های پیشنهادی برای تحقیقات آینده

۱. مکانیسم امنیتی پیشرفته : تحقیقات می تواند به سمت توسعه مکانیسم های امنیتی قوی سوق داده شود که می توانند به طور موثری منابع سخت افزاری را در یک سیستم exokernel حفاظت کنند.

۲. استاندارد سازی : برای مقابله با مشکلات سازگاری، استانداردهایی می توانند برای سفارشی سازی abstraction های سیستم عامل توسعه یابند.

۳. API های ساده شده مدیریت منابع : برای کاهش پیچیدگی توسعه برنامه، API های ساده شده می توانند برای مدیریت منابع در سطح برنامه توسعه یابند.

در مورد امکان پذیری پیاده سازی معماری exokernel در مقیاس بزرگ، بیشتر به مورد استفاده خاص بستگی دارد. برای برنامه هایی که نیاز به عملکرد بالا و انعطاف پذیری دارند و جایی که توسعه دهندگان دارای تخصص برای مدیریت منابع در سطح برنامه هستند، معماری exokernel می تواند مناسب باشد. با این حال، برای محاسبات معمولی (و روتین)، پیچیدگی به نسبت افزایش میابد که میتواند کار توسعه را سخت کند.