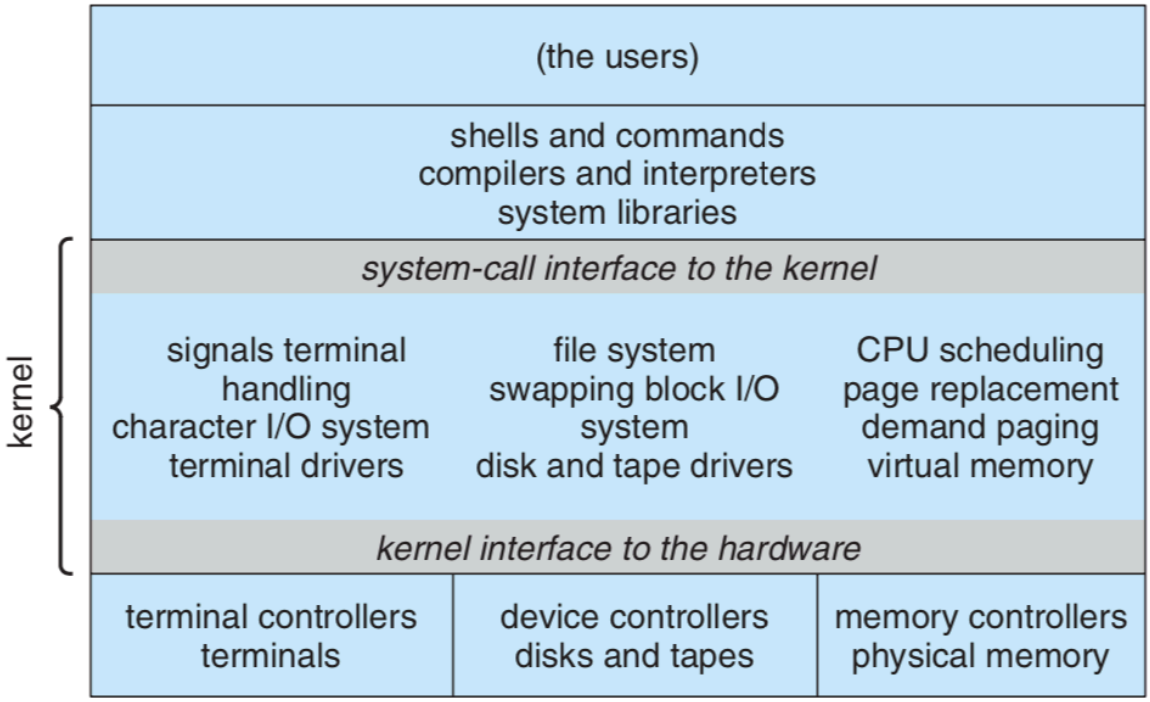
سیستم عامل

تمرین دوم

علیرضا حیدری - ۹۴۳۱۰۵۷

۱. برای برقراری ارتباط با kernel نیازمند یک api هستیم تا ارتباط با آن برقرار شود و بتوانیم دستورات را به آن بخش انتقال کنیم زیرا به صورت مستقیم به آن دسترسی نداریم و به این صورت امن‌تر است. برای برقراری این ارتباط از طریق system call ها و طریقه‌ی استفاده ی آن میتوانیم این ارتباط را برقرار کنیم. این لایه در بالای لایه ی فیزیکی و kernel قرار دارد.



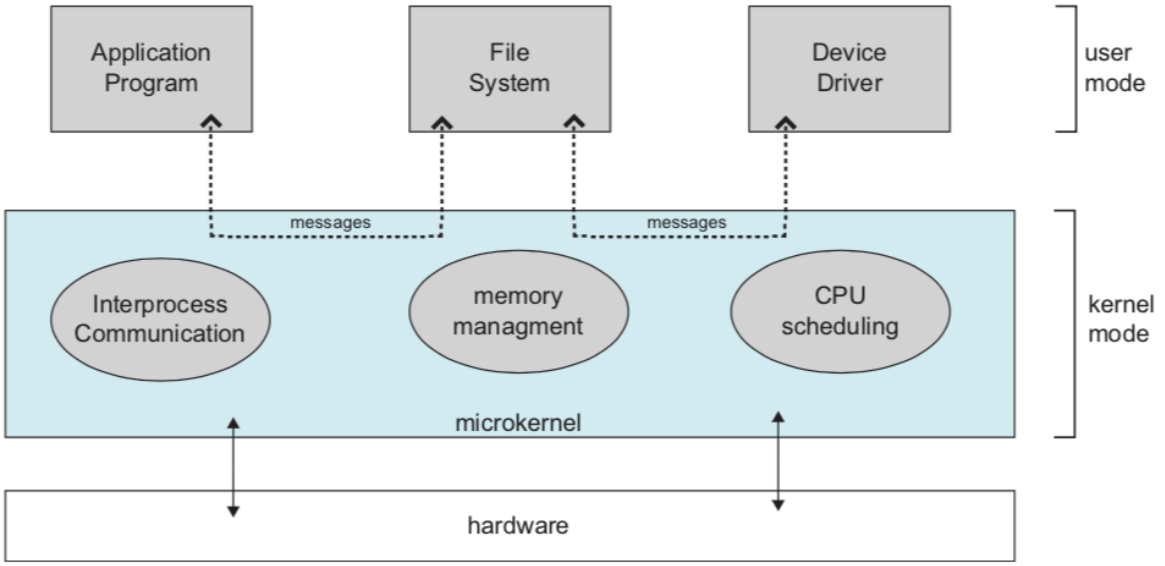
۲.سیستم monolothic به صورت یکپارچه عمل کرده و تمام برنامه در یک فایل قرار میگیرد. بنابر این در صورت بروز خطا یا bug کل برنامه دچار مشکل شده و انجام عملیات دچار توقف شده و یا به طور کامل دچار مشکل می شود. بنابراین این نوع سیستم باعث می شود تا نتوان برنامه را در حالت ادامه نگه داشت و درصورت بروز مشکل کوچکی کل برنامه نابود می شود.

۳.مدل لایه ای به این صورت است که هر لایه ممکن است به سری ای از داده ها نیاز داشته باشد. برای این ممکن باید این داده ها در لایه های پایین تر قرار داده شده باشد تا در دسترس باشد. بنابر این هر متغییر و یا داده عمومی تری در لایه های پایین تر قرار میگیرد. برای مدیریت این لایه ها در زیرین ترین لایه یک مدیریت کامل باید انجام شود چون این لایه نزدیکترین لایه به سخت افزار نیز هست.این مدیریت به دلیل انتقال داده ها از لایه های بالایی به پایینی و توالی این انتقال ها و زیاد بودن آن ها باعث شده تا این مدل از سیستم به دلیل overhead زیاد کاربرد کمتری داشته باشد.

۴.در مدل لایه ای هر لایه تنها به عملیات و توابع و داده های لایه ی پایینی دسترسی دارد.بنابر این برای دیباگ تنها لازم است به لایه های زیرین لایه فعلی که در حال دیباگ کردن است مراجعه کرد.بنابر این اولین لایه و

پایین ترین لایه بدون مشکل دیباگ خواهد شد و به همین ترتیب میتوان به لایه های بعدی رفته و مراحل دیباگ کردن را ادامه داد.

۵.در میکروکرنل بسیاری از عملیات ها در سطح کاربری و user انجام می شود و عملیات ها را به سمت سطح کاربری برده و در این حالت در صورت رخ دادن مشکلی در سطح کاربر تنها آن قسمت دچار مشکل شده و نابود می شود و بقیه سیستم به کار خود ادامه می دهد و این کار باعث بقای بیشتر کار در سیستم عامل می شود.



در عکس بالا میتوانید تقسیم کارهای میکروکرنل به دو قسمت سمت کرنل و سمت یوزر را مشاهده کنید.

۶. فعالیت ها و عملکردهای میکروکرنل در حدی که ممکن بوده به صورتی طراحی شده تا در سمت کاربر انجام شود و این کار باعث میشود تا کرنل تغییرات خاصی نیاز نداشته باشد و درصورت نیاز بودن به این تغییرات به مقدار کمی باشند. پس انتقال میکروکنترل به دلیل کار بیشتری در سمت کاربر راحت تر منتقل می شود و نیازی به بررسی و تغییر کرنل اصلی نداریم.

۷.انتقال و ارتباطات داده ها در میکروکرنل به صورت message passing است. پس اگر کاربر فایلی را خواست باید با فایل سرور ارتباط برقرار کند و قسمت کلاینت برنامه و سرور هیچگاه به صورت مستقیم با یکدیگر در تماس نیستند.در صورتی که ارتباطات module ها به صورت داخلی بوده و به دلیل این داخلی بودن سرعت بیشتری را دارا هستند و باعث انتقال سریعتر داده ها میشوند. پیاده سازی های ماژول و میکروکنترل شبیه هم است اما در ماژول به دلیل نیاز نداشتن ماژول به برقراری ارتباط به صورت message passing و انتقال داده ها به صورت داخلی به صرفه تر است از این مدل استفاده کرد.