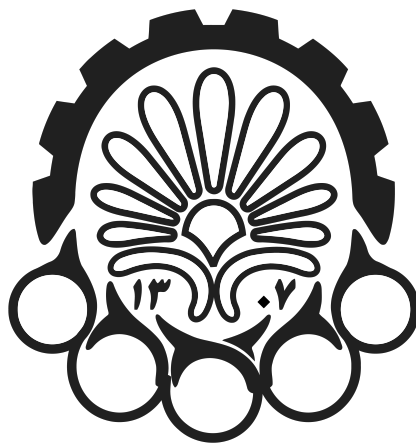


سیستم‌های عامل  
دکتر زرندی



**دانشگاه صنعتی امیرکبیر**  
( پلی تکنیک تهران )  
دانشکده مهندسی کامپیوتر

رضا آدینه پور ۴۰۲۱۳۱۰۵۵

تمرین سری اول

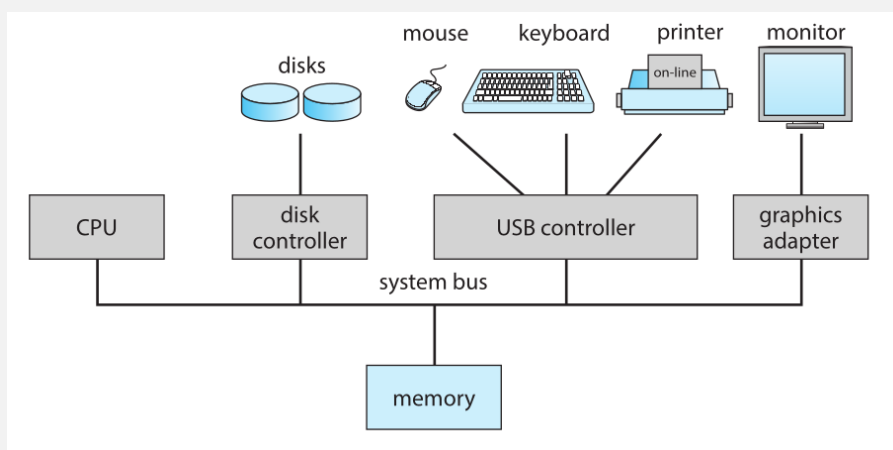
۴ مهر ۱۴۰۳

## سوال اول

به سوالات زیر در مورد دستگاه‌های ورودی و خروجی و نحوه انتقال اطلاعات از آنها پاسخ دهید.

۱. وظیفه کنترلر و درایور دستگاه‌ها چیست؟ تعامل این دو قسمت با یکدیگر و با دستگاه مربوط به خودشان از آغاز تا پایان یک عملیات I/O به چه صورت است؟

کنترلر، سخت‌افزاری است که به طور مستقیم با دستگاه I/O تعامل دارد. این قطعه، دستورات و سیگنال‌های کنترلی را از سیستم عامل دریافت می‌کند و آن‌ها را به زبان و فرمت قابل فهم برای دستگاه تبدیل می‌کند. همچنین اطلاعات را از دستگاه به سیستم برمی‌گرداند. کنترلر عملاً مانند واسطه‌ای است که باعث می‌شود سیستم عامل بتواند با دستگاه‌های جانبی ارتباط برقرار کند. در شکل زیر نقش کنترلرهای I/O برای ارتباط با کامپیوتر مشهود است.



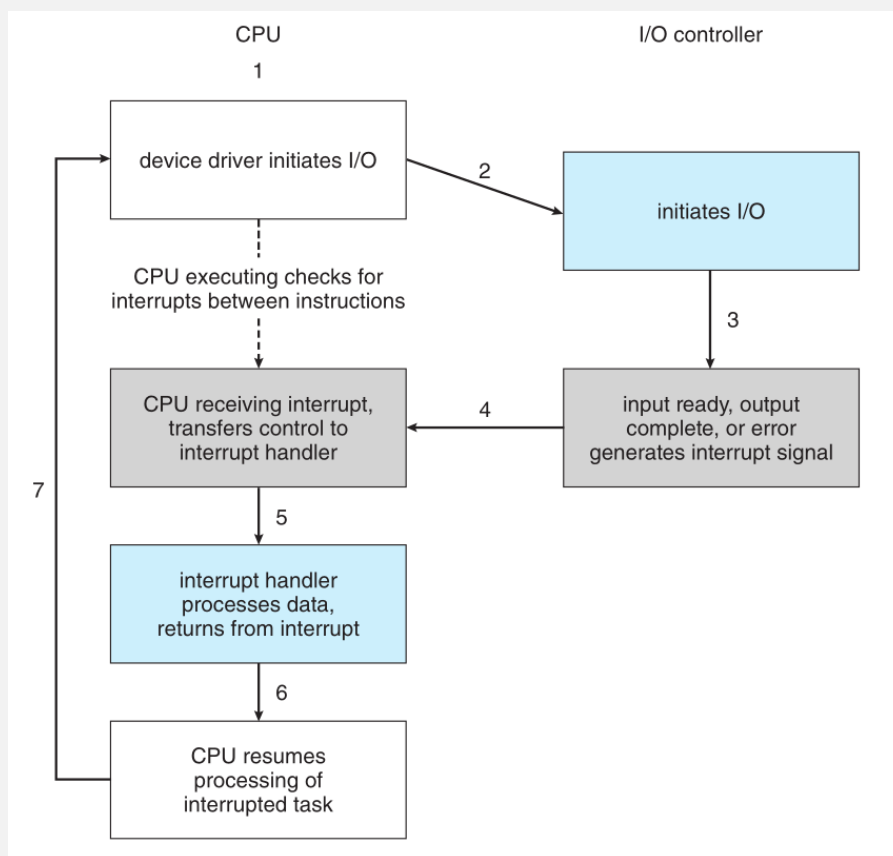
شکل ۱: نقش کنترلر‌ها در سیستم

درایورها نرم‌افزارهایی هستند که وظیفه آن‌ها ایجاد هماهنگی بین سیستم عامل و کنترلر است. درایور، سیگنال‌ها و دستورات سیستم عامل را به فرمت مورد نیاز کنترلر و دستگاه ترجمه می‌کند و به کنترلر ارسال می‌کند. در واقع، بدون وجود درایور، سیستم عامل نمی‌تواند مستقیماً با کنترلر و در نتیجه دستگاه ارتباط برقرار کند. [مرجع]

برای مثال یک برنامه که عملیات I/O را انجام می‌دهد فرض کنید. برای شروع یک عملیات I/O، درایور دستگاه رجیسترهای مناسب را در کنترلر دستگاه بارگذاری می‌کند. کنترلر دستگاه به نوبه خود محتوای این ثبات‌ها را بررسی می‌کند تا تعیین کند که چه اقدامی انجام دهد (برای مثال «خواندن یک کاراکتر از صفحه‌کلید»). کنترلر انتقال داده از دستگاه به بافر محلی خود را آغاز می‌کند. هنگامی که انتقال داده تکمیل شد، کنترلر دستگاه به درایور دستگاه اطلاع می‌دهد که عملیات خود را به پایان رسانده است. سپس درایور دستگاه کنترل را به سایر بخش‌های سیستم عامل واگذار می‌کند و احتمالاً داده‌ها یا اشاره‌گری به داده‌ها را در صورت انجام عملیات خواندن برمی‌گرداند.

## پاسخ

برای سایر عملیات‌ها، درایور دستگاه اطلاعات وضعیت مانند «نوشتن با موفقیت انجام شد» یا «دستگاه مشغول است» را بازمی‌گرداند. کنترلر از طریق interrupt به درایور دستگاه اطلاع می‌دهد که عملیات خود را به پایان رسانده است. این فرایند در شکل زیر آورده شده است:



شکل ۲: فرایند ارتباط I/O با CPU

۲. میدانیم یک روش انتقال داده بین دستگاه‌های ورودی و خروجی و پردازنده، مبتنی بر وقفه هاست. عیب این روش چیست و چگونه در سیستم‌های کامپیوتری امروزی رفع شده است؟

یکی از عیب‌های عمده روش انتقال داده بین دستگاه‌های I/O و CPU این است که وقفه‌ها به صورت مکرر توسط دستگاه‌های مختلف متصل به سیستم رخ می‌دهند. بنابر این این وقفه‌ها باید مدیریت شوند تا از تداخل وقفه‌ها با یکدیگر جلوگیری شود. این مدیریت در کامپیوترهای امروزی به صورت زیر رفع شده است:

(آ) کامپیوترهای امروزی قادر به تاخیر انداختن مدیریت وقفه‌ها در طول پردازش‌های بحرانی و مهم هستند. یعنی به وقفه‌ها اولویت داده می‌شود و وقفه‌هایی با اولویت بالا تر در اولویت سرویس‌دهی قرار دارند. اکثر CPU ها دو خط درخواست وقفه دارند. یکی وقفه nonmaskable interrupt است که برای رویدادهایی مانند خطاهای حافظه غیرقابل بازیابی رزرو شده است. خط وقفه دوم maskable است. این خط می‌تواند توسط CPU قبل از اجرای توالی‌های بحرانی دستورالعمل‌ها که نباید وقفه‌ای در آن‌ها ایجاد شود، خاموش شود. وقفه maskable توسط کنترل‌کننده‌های دستگاه برای درخواست سرویس استفاده می‌شود.

## پاسخ

(ب) مشکل بعدی این است که درخواست‌های بالای سرویس‌دهی وقفه می‌تواند پردازنده را مدام مشغول رسیدگی به درخواست‌های وقفه کند بنابر این بخش زیادی از توان پردازشی پردازنده صرف رسیدگی به این درخواست‌ها می‌شود. برای رفع این مشکل، در کامپیوترهای امروزی برای انتقال حجم بالای داده بین دستگاه‌های I/O و حافظه بدون نیاز به مداخله پردازنده، از تکنیکی به نام DMA استفاده می‌شود. در این روش، یک کنترلر خاص به نام کنترلر DMA عملیات انتقال داده را مدیریت می‌کند و پس از اتمام کار، تنها یک وقفه به پردازنده ارسال می‌کند. این کار باعث کاهش تعداد وقفه‌ها و در نتیجه آزادسازی پردازنده برای سایر وظایف می‌شود.

## سوال دوم

تصور کنید یک کامپیوتری دارای چندین دستگاه ورودی/خروجی (I/O) مانند کیبورد و اسکنر است. میدانیم که این دستگاه‌ها برای ارسال و دریافت اطلاعات از CPU نیاز به مدیریت دارند. حال توضیح دهید زمانی که کاربر کلیدی را بر روی کیبورد فشار می‌دهد و یا می‌خواهد عکسی با حجم کم را اسکن کند، چه فرآیند و مراحل میان دستگاه I/O، Device Controller، CPU، Memory طی می‌شود.

وقتی کاربر کلیدی را بر روی کیبورد فشار می‌دهد یا عکسی را اسکن می‌کند، فرآیند تبادل داده بین دستگاه I/O، کنترلر دستگاه، CPU و حافظه طی مراحل مشابه به هم انجام می‌شود. که این مراحل را در ادامه برای کیبورد و اسکنر به طور جداگانه توضیح خواهیم داد:

### ۱. فشردن کلید کیبورد

- (آ) ارسال سیگنال از کیبورد به کنترلر دستگاه:  
وقتی کاربر کلیدی را روی کیبورد فشار می‌دهد، یک سیگنال الکتریکی به کنترلر کیبورد ارسال می‌شود. این سیگنال نشان‌دهنده‌ی کد آن کلید خاص است که به صورت یک کد اسکی توسط کیبورد تولید می‌شود.
- (ب) ارسال وقفه به CPU:  
کنترلر کیبورد سیگنال را دریافت کرده و یک وقفه به CPU ارسال می‌کند. این وقفه به CPU اطلاع می‌دهد که یک داده جدید از کیبورد برای پردازش وجود دارد.
- (ج) سرویس‌دهی به وقفه:  
CPU به وقفه پاسخ داده و اجرای فرآیند جاری را متوقف می‌کند. سپس CPU به سراغ دستورالعمل‌های مدیریت وقفه (Interrupt Handler) می‌رود. این دستورالعمل‌ها مشخص می‌کنند که وقفه از کیبورد است و باید داده‌ی کیبورد خوانده شود.
- (د) دریافت داده از کنترلر:  
CPU به کنترلر کیبورد پیام می‌فرستد و داده‌ی مربوط به کلید فشرده شده را درخواست می‌کند. کنترلر کیبورد این داده را که معمولاً کد اسکی مربوط به کلید فشرده شده است، به CPU ارسال می‌کند.
- (ه) ذخیره‌سازی داده در حافظه:  
CPU داده دریافت‌شده را به حافظه RAM منتقل می‌کند تا در صورت نیاز توسط برنامه‌های در حال اجرا (مثلاً یک نرم‌افزار پردازش متن) استفاده شود.
- (و) ادامه اجرای برنامه‌ها:  
پس از پردازش وقفه، CPU به برنامه قبلی بازگشته و اجرای آن را از سر می‌گیرد.

### ۲. اسکن عکس با اسکنر

- (آ) ارسال سیگنال از اسکنر به کنترلر دستگاه:  
وقتی کاربر درخواست اسکن یک عکس را می‌دهد، اسکنر شروع به جمع‌آوری داده‌های تصویری می‌کند. این داده‌ها به صورت پیکسل به پیکسل به کنترلر اسکنر ارسال می‌شوند. کنترلر دستگاه این داده‌ها را در بخش‌های کوچک بسته‌بندی می‌کند تا برای انتقال آماده باشند.
- (ب) ارسال وقفه به CPU:  
همانند کیبورد، کنترلر اسکنر نیز پس از جمع‌آوری بخشی از داده‌ها، یک وقفه به CPU ارسال می‌کند تا به آن اطلاع دهد که داده‌های جدید برای پردازش آماده هستند.

## پاسخ

- (ج) سرویس‌دهی به وقفه:  
CPU به وقفه اسکنر پاسخ داده و دستورالعمل‌های مربوط به مدیریت وقفه را اجرا می‌کند. CPU سپس از کنترلر اسکنر درخواست داده می‌کند.
- (د) انتقال داده به حافظه:  
کنترلر اسکنر داده‌های تصویر را به CPU ارسال می‌کند و CPU این داده‌ها را به حافظه اصلی منتقل می‌کند. در اینجا ممکن است از DMA استفاده شود تا حجم بالای داده‌های تصویر بدون نیاز به پردازنده مستقیماً به حافظه منتقل شود.
- (ه) ادامه اسکن و پردازش:  
اسکنر همچنان به جمع‌آوری داده‌های جدید ادامه می‌دهد و هر بار که یک بخش از داده‌ها آماده شد، یک وقفه دیگر به CPU ارسال می‌شود تا داده‌های جدید به حافظه منتقل شوند. این فرآیند تا زمانی که اسکن کامل شود، تکرار می‌شود.

## سوال سوم

میدانیم گاهی اوقات CPU در وضعیت HALT قرار می‌گیرد. این وضعیت را توضیح دهید و شرح دهید در چه مواردی CPU در آن قرار می‌گیرد.

وقتی CPU در حالت HALT قرار می‌گیرد، تمام فرآیندهای عادی متوقف می‌شوند و پردازنده به صورت نیمه‌فعال باقی می‌ماند. (پردازنده کلاک می‌خورد اما الگوریتم فن‌یومن اجرا نمی‌شود) در این حالت، CPU هیچ دستوری از برنامه‌های در حال اجرا را پردازش نمی‌کند و به جای آن، در یک حالت انتظار کم‌مصرف قرار می‌گیرد. این وضعیت به معنی خاموش شدن کامل CPU نیست، بلکه پردازنده به نوعی در حالت انتظار قرار دارد تا زمانی که یک رویداد خاص، مانند یک Interrupt یا reset رخ دهد و آن را از حالت HALT خارج کند.

- یکی از رایج‌ترین مواردی که CPU در وضعیت HALT قرار می‌گیرد، زمانی است که سیستم منتظر یک Interrupt است. به عنوان مثال، وقتی پردازنده منتظر دریافت داده از دستگاه‌های I/O است (مانند انتظار برای فشردن کلید در کیبورد)، می‌تواند به جای اجرای دستورات بیهوده، وارد حالت HALT شود. سپس با وقوع یک وقفه (مثلاً فشار دادن کلید روی کیبورد)، دازنده از حالت HALT خارج شده و پردازش را ادامه می‌دهد.
- معمولاً در زمان‌هایی که سیستم نیاز به پردازش ندارد یا بیکار است، پردازنده می‌تواند به حالت HALT برود تا مصرف انرژی را کاهش دهد. این روش به ویژه در دستگاه‌های قابل حمل مانند لپ‌تاپ‌ها و گوشی‌های هوشمند که مصرف انرژی اهمیت زیادی دارد، کاربرد دارد.
- در سیستم‌های چند هسته‌ای، ممکن است یک یا چند هسته‌ی پردازنده در حالتی قرار بگیرند که نیاز به پردازش نداشته باشند. در چنین مواقعی، این هسته‌ها به حالت HALT می‌روند تا مصرف انرژی بهینه‌تر شود و تنها در صورتی که نیاز به پردازش مجدد داشته باشند، فعال می‌شوند.

## سوال چهارم

در خصوص Bootstrap و تفاوت آن با Bootloader تحقیق کنید و همچنین توضیح دهید چرا برنامه Bootstrap در حافظه داخلی ذخیره نمی‌شود.

Bootstrap و Bootloader بسیار مرتبط با هم هستند اما تفاوت‌های جزئی‌ای هم با هم دارند که در ادامه به آن‌ها می‌پردازیم.

### ۱. Bootloader:

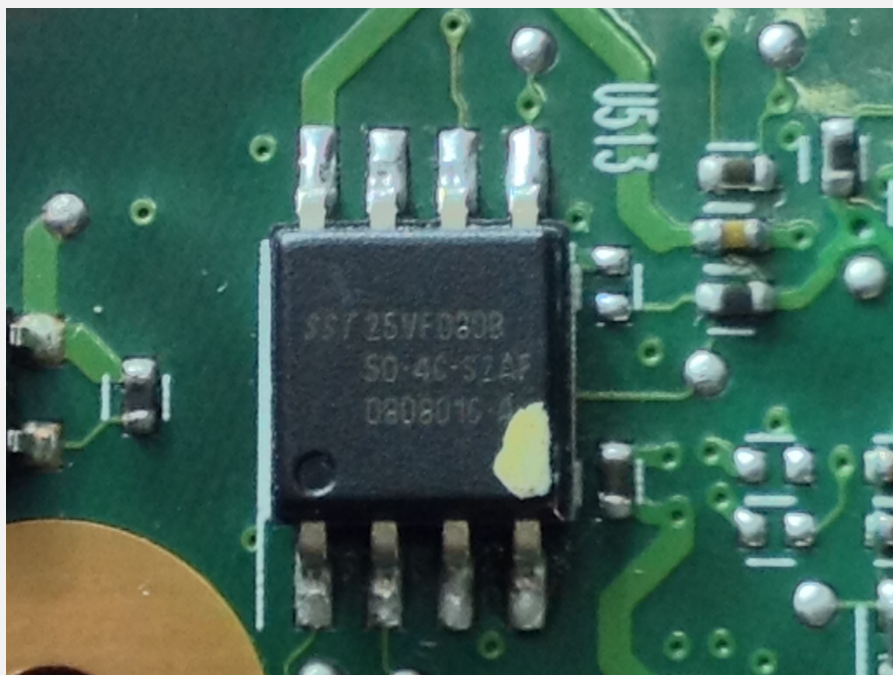
وظیفه‌ی اصلی bootloader بارگذاری سیستم‌عامل در حافظه‌ی اصلی است. وقتی کامپیوتر روشن می‌شود، سخت‌افزار اولیه مثل BIOS یا UEFI فعال می‌شود و bootloader را از حافظه غیر فرار مثل هارد دیسک بارگذاری می‌کند. این برنامه به طور معمول در اولین بخش دیسک به نام MBR ذخیره می‌شود.

Bootloader در واقع قطعه کدی است که قبل از اجرای هر سیستم عاملی اجرا می‌شود و سیستم‌عامل را برای اجرا آماده می‌کند.

### ۲. Bootstrap:

از اصطلاح قدیمی bootstrapping به معنای راه‌اندازی خودکار سیستم نشأت می‌گیرد. این برنامه که گاهی به آن bootstrap loader نیز می‌گویند، در حافظه غیر فرار (مانند ROM یا EPROM) ذخیره می‌شود و به‌طور مجزا بر روی مادربرد قرار می‌گیرد. Bootstrap معمولاً در سطح ابتدایی‌تری نسبت به Bootloader عمل می‌کند و صحت و سلامت سخت‌افزار را برای بارگذاری bootloader تست می‌کند.

در شکل زیر نمونه‌ای از آیسی حافظه غیر فراری که Bootstrap در آن قرار دارد آورده شده است:



شکل ۳: آیسی ذخیره‌کننده برنامه Bootstrap