

پرسش ۲

سه گونه انتقال خروجی و ورودی داریم : ۱. برنامه سازی شده / ۲. بر اساس وقفه / ۳. به کمک DMA

برنامه سازی شده : در این روش وقتی فرآیندی که در دست CPU است، به دستورات ورودی و خروجی میرسد؛ سیستم عامل واحدهای I/O را فراخوانی میکند تا به بررسی ورودی و خروجی پردازند. و در این هنگام CPU میبایست به صورت مداوم و پیاپی به سراغ واحدهای ورودی و خروجی برود؛ تا بفهمد که آیا کارشان با آن فرآیند تمام شده است؛ یا خیر. از آن جایی که کسی به CPU خبر نمیدهد و خود CPU میبایست این موضوع را بررسی کند که آیا کار I/O تمام شده است؛ این روش کارایی سیستم را کاهش میدهد. (در صورتی که در سیستم های مبتنی بر وقفه این موضوع مستقیماً به CPU اطلاع داده میشود)

بر اساس وقفه : این روش سرکلاس هم توضیح داده شده است. در این متد، فرآیند موجود در CPU هنگامی که به I/O میرسد. به واحدهای I/O سپرده میشود و CPU به کار مفید خود ادامه میدهد. و دیگر منتظر نمیماند که فرآیند ورودی و خروجی به پایان برسد. و واحدهای I/O با ایجاد وقفه به CPU پایان فرآیند ورودی و خروجی را اطلاع میدهند. همانطور که سرکلاس هم گفته شد وقفه مانند ناقوس کلیسا میماند چنانچه نواخته شد، CPU مجدداً به بررسی همان فرآیند میپردازد. نکته اینجاست که CPU هر بار قبل از انجام کار میبایست سیگنال وقفه را بررسی کند که آیا فعال شده است یا خیر. که این روش هم اگرچه از روش برنامه سازی شده کارآمد تر است اما باز هم CPU را درگیر خود میسازد.

به کمک DMA : در این روش بین واحدهای I/O و RAM مسیری در نظر میگیریم. یعنی واحد DMA مستقیماً به RAM دسترسی دارد. وقت هایی که فرآیند به I/O میرسد. CPU اطلاعات I/O این فرآیند را برای DMA ارسال میکند که شامل ورودی یا خروجی بودن، آدرس خانه این فرآیند در RAM، مقدار ورودی و خروجی، و دستگاه I/O مربوطه ... است. پس از آن پردازنده به کار مفید میپردازد (روی فرآیندی دیگر کار میکند) و هنگامی که کار تمام شد DMA پایان کار را به CPU اعلام میکند. نکتهی این روش در اینجاست که چون CPU و DMA از یک BUS استفاده میکنند. گاهی اوقات ممکن در هنگامی که تجمع دستورات I/O وجود دارد سیستم کند شود ولی این روش از دو روش قبلی کارا تر است.

پرسش ۶

کارهایی که سیستم ها میتوانند انجام دهند به دو دسته تقسیم میشود: ۱. کاربر / ۲. کرنل . از آنجایی که منابع مورد استفاده در کامپیوتر محدود است و هم در اختیار کاربر است و هم سیستم عامل؛ سیگنال کرنل تعریف میشود که اگر یک بود یعنی در مود کرنل هستیم و اگر صفر بود در مود کاربر. هنگامی که در مود کاربر قرار بگیریم نمیتوانیم بعضی از دسترسی های که ممکن است موجب اختلال در عملکرد سیستم عامل شود را انجام دهیم. و هنگامی که دسترسی های مهم تری را بخواهیم (نظیر کنترل I/O ، کنترل وقفه و...) میبایستی به مود کرنل برویم privileged instruction دستوراتی هستند که در مود کرنل انجام میپذیرند.

پرسش ۸

از آنجایی که یکی از استفاده‌کننده‌های CPU خود سیستم عامل است. به همین دلیل میبایست حواسش به این نکته باشد که CPU برای مدت زیادی در اختیار یک فرآیند قرار نگیرد. برای مثال اگر یک فرآیند در حلقه همیشه برقرار گیرکند (به اصطلاح لوپ بینهایت!) یا مقدار زیادی پردازش داشته باشد که باعث شود مدت زمان زیادی CPU در اختیار سیستم عامل نباشد. در این حالت چنانچه سیستم منتظر پایان فرآیند بنشیند، هیچ گاه مجدداً CPU به سیستم عامل بازگردانده نخواهد شد. به همین دلیل برای حل این مشکل از **وقفه تایمر** استفاده میشود. که تایمر پس از مدت زمانی معین وقفه ایجاد میکنند و سیستم عامل CPU را در اختیار میگیرد و میتواند در مورد فرآیند تصمیم بگیرد (دیگر به آن CPU ندهد یا مجدداً به آن فرصت پردازش بدهد)

پرسش ۱۰

فراخوانی های سیستمی یا همان system call ها درواقع اینترفیس هست که میتوان به واسطه آن به سرویس های سیستم عامل دسترسی داشت. بسته به سطح فراخوانی ها این سیستم کال ها یا با زبان C یا با زبان های سطح پایین تر مانند اسمبلی قابل برنامه نویسی است.

پرسش ۱۱

فرض کنیم پردازنده در حال انجام فرآیند هست و در این میان وقفه توسط یک دستگاه I/O برانگیخته میشود. پردازنده در پایان هر دستورالعمل و قبل از شروع دستورالعمل بعدی سیگنال وقفه (که به یکی از پایه‌های CPU ارسال میشود) را بررسی میکند. در گذشته که تعداد دستگاه‌های I/O کم بودند؛ چندین سیگنال وقفه و چندین پایه برای آن وجود داشت. اما در CPU های جدیدتر از این روش استفاده میکنند که سیگنال درخواست وقفه ارسال میشود. CPU سیگنال تصدیق وقفه را ارسال میکند، ولی هنوز نمیداند که فرستنده این سیگنال کیست. سپس واحد کنترل کننده شماره دستگاه صادرکننده وقفه را روی BUS میگذارد. و CPU متوجه میشود که وقفه از چه ناحیه‌ی چه دستگاهی صادر شده است.

پرسش ۱۲

قسمت اول : CPU به عنوان ورودی سیگنال خط وقفه (یا Interrupt – Line) دارد که این سیگنال وقت هایی که وقفه رخ داده است یک میشود. CPU هر عملیاتی را که میخواهد انجام دهد. قبل از اجرای عملیات این سیگنال را چک میکند که چنانچه وقفه‌ای رخ نداده بود (سیگنال صفر بود) ادامه عملیات خود را انجام میدهد. و چنانچه وقفه اتفاق افتاده بود. اطلاعات همان لحظه Process را ذخیر میکند(به اصطلاح سفره اش را جمع میکند) و به بررسی وقفه میپردازد. به این شکل که پردازنده وقفه را به Interrupt handler میفرستد و دستورات مربوط به هندل کردن وقفه را انجام میدهد. پس از اجرای وقفه اطلاعات آن را پاک میکند و به Process قبلی را ادامه میدهد(به اصطلاح سفره را پهن میکند!)

قسمت دوم: دو نوع وقفه در پردازنده ها وجود دارد : یک.Maskable: این وقفه ها در مواقع نیاز(حالت هایی که Process در حال اجرا ضروری باشد) میتواند توسط CPU نادیده گرفته شود. دو.Nonmaskable: این وقفه ها هیچگاه نمیتواند نادیده گرفته شود و همواره CPU هنگام دریافت چنین وقفه هایی به بررسی وقفه میپردازد.

پرسش ۱۷

PSW یک رجیستر پردازنده است که اطلاعاتی مهم نظیر مود کرنل یا کاربر، نتیجه مقایسه های انجام شده، اولویت گذاری هایی که پردازنده و سیستم عامل دارند و... را در خود نگه میدارد. این رجیستر برای برنامه های کاربر قابل خواندن است. اما نوشتن در آن محدودیت دارد.