

(وقفه های کامپیوتری)

از نظر مجازی . تمام کامپیوتر ها راهکاری را فراهم می کنند تا قطعات دیگر (مثل I/O و حافظه) ، در کار عادی پردازنده وقفه ایجاد کنند . جدول زیر متداول ترین وقفه ها را نشان می دهد .

جدول : دسته بندی وقفه ها

برنامه وقفه هایی که در اثر بعضی از شرایط حاصل از اجرایی یک دستور العمل ، مثل سرریز ، تقسیم بر صفر ، سعی در اجرای دستور العمل نامعتبر ماشین ، و خروج از فضای حافظه مجازی رخ می دهند .

تایمر وقفه ای که توسط تایمر موجود در پردازنده ایجاد می شود . این وقفه موجب می شود سیستم عامل بعضی از اعمال را به طور منظم انجام دهد .

I/O وقفه ای که توسط کنترلر I/O ایجاد می شود تا کامل شدن یک عملیات یا وجود خطاهایی را در انجام عملیات اعلان کند .

خرابی سخت افزار این وقفه در اثر خرابی های مثل خرابی منبع تغذیه یا خطای توازن در حافظه رخ می دهد .

وقفه ها ابتدا برای بهبود کارایی پردازنده ایجاد شدند . به عنوان مثال ، اغلب دستگاه های I/O خیلی کندتر از پردازنده اند . فرض کنید پردازنده ای بر اساس الگوی چرخ داده هایی را به چاپگر ارسال می کند پس از هر عملیات نوشتن ، پردازنده باید متوقف شود و منتظر بماند تا چاپگر آمادگی خودش را اعلان کند .

مدت این انتظار ممکن است صدها یا هزاران برابر چرخه های دستور العملی باشد

که با حافظه کار ندارند . بدیهی است که زمان CPU به هدر می رود .

در این شکل ، برنامه کاربر در بین فراخوانی های دستورالعمل WRIT (جهت عمل

نوشتن) ، پردازش هایی را انجام می دهد . مجموعه دستورات ۱ و ۲ و ۳ فاقد I/O

هستند فراخوانی WRITE منجر به اجرای برنامه I/O می شود که یک برنامه

سودمند سیستم است و عمل I/O واقعی را انجام می دهد . این برنامه I/O شامل

سه بخش است :

➤ دنباله ای از دستور العمل ها که در شکل مشخص شد و عمل I/O واقعی را

انجام می دهند این عملیات ممکن است داده های خروجی را در میانگیری

نخیره کند و پارامترهایی را برای فرمان دادن به دستگاه آماده نماید .

➤ فرمان I/O واقعی ، بدون استفاده از وقفه ها ، وقتی این فرمان صادر می

شود ، برنامه ممکن است منتظر دستگاه I/O بماند تا این دستگاه وظیفه اش

را انجام دهد (یا به طور متناوب ، وضعیت دستگاه I/O را کنترل نماید تا

مشخص شود عمل I/O تمام شد یا خیر).

➤ دنباله ای از دستور العمل ها که در شکل با شماره ۵ مشخص شد و

عملیات را کامل می کند . این عملیات ممکن است برچسبی را مقدار دهد که

نشان دهنده موفقیت یا شکست آن باشد .

چون کامل شدن عمل I/O ممکن است مدت زیادی طول بکشد ، برنامه I/O منتظر می ماند تا عمل I/O کامل شود . لذا برنامه I/O در نقطه فراخوانی WRITE به مدت زیادی منتظر می ماند .

وقفه ها و چرخه دستور العمل

با استفاده از وقفه ها ، وقتی که عمل I/O در حال انجام است ، پردازنده می تواند دستور العمل های دیگری را اجرا کند . جریان کنترل را در شکل در نظر بگیرید . همانند قبل ، برنامه کار به نقطه ای می رسد که WRITE را فراخوانی می کند که یک فراخوان سبستم است . برنامه I/O که به این روش فراخوانی شد . فقط شامل کد آماده سازی و فرمان I/O واقعی است . پس از اجرای این دستور العمل ها ، کنترل به برنامه کاربر بر می گردد . در همین زمان ، دستگاه خارجی مشغول پذیرش داده ها از حافظه کامپیوتر و چاپ کردن آن است . این عملیات I/O به طور همزمان با دستور العمل های برنامه کار در حال اجرا است .

وقتی دستگاه خارجی آماده ارائه خدمات شد ، یعنی آماده شد تا داده های بیشتری را از پردازنده بپذیرد ، قطعه I/O مربوط به آن دستگاه خارجی ، سیگنال درخواست وقفه را به پردازنده می فرستد . پردازنده در پاسخ به این درخواست وقفه ، عمل برنامه فعلی را به تعویق می اندازد ، به برنامه ای می رود تا به آن دستگاه I/O خدمات دهد و پس از ارائه خدمات به آن دستگاه ، اجرای برنامه به

تعویق افتاده را از سر می گیرد . برنامه ای که در اثر وقوع وقفه اجرا می شود تا

به آن دستگاه I/O خدمات دهد ، اداره کننده وقفه نام دارد

از دیدگاه برنامه کاربر ، وقفه موجب می شود برنامه ای که در حال اجرا است به

تعویق افتد . وقتی پردازش وقفه کامل شد ، اجرای برنامه از سر گرفته می شود

لذا برنامه کاربر نباید کد خاصی برای پشتیبانی از وقفه داشته باشد ، بلکه

پردازنده و سیستم عامل ، مسئول به تعویق انداختن برنامه کاربر و از سرگیری آن

پس از پردازش وقفه اند .

برای پشتیبانی از وقفه ها ، چرخه وقفه به چرخه دستور العمل اضافه می شود

در چرخه وقفه ، پردازنده سعی می کند تا ببیند آیا وقفه ای رخ داد یا خیر . وقفه یا

سیگنال وقفه مشخص می شود . اگر هیچ وقفه ای رخ نداده باشد ، پردازنده به

چرخه واکنشی ادامه می دهد و دستور العمل بعدی برنامه فعلی را واکنشی می کند

. اگر وقفه ای رخ داده باشد ، پردازنده اجرای برنامه فعلی را به تعویق می اندازد و

روال اداره کننده وقفه را اجرا می کند و برنامه اداره کننده وقفه ، عموماً بخشی از

سیستم عامل است . این برنامه ، ماهیت و قفه را تعیین کرده عمل مورد نیاز را

انجام می دهد . در مثالی که مطرح کردیم ، اداره کننده وقفه تعیین می کند کدام

قطعه I/O وقفه را تولید کرد و ممکن است به برنامه ای برود که داده های بیشتری

را در قطعه I/O می نویسد . وقتی روال اداره کننده وقفه کامل شد ، پردازنده می

تواند اجرای برنامه کاربر را از نقطه ای که به تعویق افتاد از سر گیرد .

بدیهی است که در این فرآیند ، سربارهایی وجود دارد دستور العمل های اضافی
باید در روال اداره کننده وقفه اجرا شوند تا ماهیت وقفه را تعیین کنند و سپس عمل
مورد نظر را انجام دهند . با این وجود ، چون زمان زیادی انتظار تکمیل عمل I/O
صرف می شود ، با استفاده از وقفه ها ، پردازنده با کارایی بیشتری استفاده می
گردد.

برای این که قدر این کارایی را بدانید یک نمودار زمانی براساس جریان کنترل
است زمان لازم برای عمل I/O نسبتاً کوتاه است ، به طوری که این زمان ، از زمان
لازم برای کامل شدن اجرای دستور العمل های بین دو دستور العمل نوشتن در
برنامه کاربر ، کمتر است . در حالت عادی ، به ویژه برای دستگاه کندی مثل چاپگر
، زمان لازم برای عمل I/O خیلی بیشتر از زمان اجرای دنباله ای از دستور العمل
های کاربر است ، در این حالت ، برنامه کار بر قبل از کامل شدن عمل I/O جدیدی
آغاز گردد. شکل بعدی تنظیم زمانی این وضعیت را با استفاده وقفه ها و بدون
استفاده از وقفه ها نشان می دهد . می بینیم که کارایی تا حدی بهبود می یابد ،
زیرا در اثنای عمل I/O دستور العمل کاربر اجرا می شوند .

پردازش وقفه

وقوع وقفه منجر به رویدادهایی در نرم افزار و سخت افزار پردازنده می شود
دنباله ای از این رویدادها نشان می دهد وقتی دستگاه I/O ، یک عمل I/O را کامل
می کند .

دنباله ای از رویدادهای سخت افزاری زیر رخ می دهد :

۱- این دستگاه ، سیگنال وقفه را به پردازنده می فرستد.

۲- پردازنده قبل از پاسخگویی به وقفه ، اجرای دستور العمل فعلی را به پایان می رساند

۳- پردازنده وجود وقفه را بررسی می کند و می بیند که وقفه ای رخ داده است ووصول سیگنال وقفه را به دستگاه صادر کننده وقفه اعلام می کند ، این اعلام وصول به دستگاه اجازه می دهد تا سیگنال وقفه اش را حذف کند .

۴- اکنون پردازنده باید آماده انتقال کنترل به روال وقفه شود . ابتدا باید اطلاعات لازم را برای از سرگیری برنامه فعلی از نقطه وقوع وقفه ، ذخیره نماید . حداقل اطلاعات لازم ، کلمه وضعیت برنامه (PSW) و محل دستور العمل بعدی (PC) است . این اطلاعات را می توان در پشته کنترل سیستم ذخیره کرد.

۵- اکنون پردازنده ، آدرس محل ورود به روال وقفه ای را که به این وقفه پاسخ می دهد ، در شمارنده برنامه (PC) قرار می دهد . بسته به معماری کامپیوتر و طراحی سیستم عامل ، ممکن است برای هر نوع وقفه یک برنامه اداره کننده وقفه وجود داشته باشد یا برای هر دستگاه و هر وقفه چنین برنامه ای موجود باشد اگر بیش از یک روال اداره کننده وقفه وجود داشته باشد ، پردازنده باید تصمیم بگیرد کدام را فراخوانی کند . این اطلاعات ممکن است در سیگنال وقفه اصلی

وجودداشته باشد ، یا ممکن است پردازنده از دستگاه صادر کننده وقفه درخواست کند که این اطلاعات را در اختیارش قرار دهد .

وقتی شمارنده برنامه بار شد ، پردازنده به چرخه دستور العمل بعدی می رود که با چرخه واکنشی دستور العمل شروع می شود . چون واکنشی دستور العمل توسط محتویات شمارنده برنامه تعیین می شود ، نتیجه اش این است که کنترل به برنامه اداره کننده وقفه منتقل می شود . اجرای این برنامه منجر به عملیات زیر می گردد :

۶- در این نقطه ، شمارنده برنامه و PSW برنامه ای که دچار وقفه شد ، در پشته سیستم ذخیره شده است . اما ، اطلاعات دیگری وجود دارد که به عنوان بخشی از حالت برنامه در حال اجرا محسوب می شود . به خصوص محتویات ثبات های پردازنده باید ذخیره شوند ، زیرا این ثبات ها ممکن است توسط روال اداره کننده وقفه استفاده شوند .

بنابراین تمام این مقادیر و سایر اطلاعات مربوط به حالت برنامه باید ذخیره شوند . معمولاً روان اداره کننده وقفه با ذخیره محتویات تمام ثبات ها شروع می شود .

سایر اطلاعات مربوط به حالت برنامه در حال اجرا که باید ذخیره شوند ، در این مثال ، برنامه کاربر پس از اجرای دستور العمل موجود در محل N دچار وقفه می شود . محتویات تمام ثابت ها و آدرس دستور العمل بعدی (N+1) در پشته کنترل ذخیره می شود که در مجموع M کلمه است . مقدار اشاره گر پشته عوض می

شود تا به بالای پشته اشاره کند و مقدار شمارنده برنامه عوض می شود تا به روال اداره کننده وقفه اشاره کند .

۷- اکنون اداره کننده وقفه شروع به پردازش وقفه می کند . این عمل شامل بررسی اطلاعات و وضعیت مربوط به عملیات I/O یا رویداد دیگری است که منجر به وقفه شده است . علاوه بر این ، شامل ارسال فرمان های دیگر؛ اعلام وصول هایی به دستگاه I/O است .

۸- وقتی پردازش وقفه کامل شد ، مقادیر ثبات ها که در پشته ذخیره شدند از پشته بازیابی شده در ثبات ها بار می شوند .

۹- در مرحله آخر ، مقادیر PC,PSW از پشته بازیابی می شوند . در نتیجه ، دستور العمل بعدی که اجرا می شود ، مربوط به برنامه ای است که به تعویق افتاد بود .

ذخیره تمام اطلاعات مربوط به برنامه ای که دچار وقفه شد ، برای از سرگیری آن مهم است . علتش این است که وقفه روالی نیست که توسط برنامه فراخوانی شود ، بلکه وقفه در هر زمانی و در هر نقطه از اجرای برنامه می تواند رخ دهد و وقوع آن قابل پیش بینی نیست .

وقفه های چندگانه

تاکنون در مورد وقوع یک وقفه بحث کردیم . اما ، فرض کنید چند وقفه بتواند رخ دهد . به عنوان مثال ، ممکن است برنامه ای در حال دریافت داده ها از خط ارتباطی و چاپ نتایج باشد . چاپگر هر وقت عمل چاپ را کامل کرد وقفه ای را صادر می

کند . کنترلگر خط ارتباطی ، هر وقت به یک واحد از داده های می رسد وقفه ای را صادر می کند . بسته به نظام ارتباطی ، واحد داده ممکن است یک کاراکتر یا یک بایت باشد . در هر مورد ، ممکن است در اثنای پردازش وقفه چاپگر ، وقفه ارتباطی رخ دهد .

دو روش برای برخورد با وقفه های چندگانه وجود دارد یک روش این است که در اثنای پردازش یک وقفه سایر وقفه ها از کار انداخته شوند . معنای وقفه از کار افتاده این فاصله زمانی وقفه ای رخ دهد ، عموماً منتظر می ماند و پس از این که وقفه ها دوباره به کار انداخته شدند ، پردازنده وجود وقفه ها را بررسی می کند .

لذا وقتی برنامه کاربر در حال اجرا است و وقفه ای رخ دهد . وقفه ها فوراً از کار انداخته می شوند . پس از کامل شدن روال اداره کننده وقفه ، قبل از ادامه اجرای برنامه کاربر ، وقفه ها دوباره به کار می افتند و پردازنده بررسی می کند که آیا وقفه های دیگری رخ دادند یا خیر . چون در این روش ، وقفهها به ترتیب اجرا می شوند ، عالی و ساده است

عیب این روش این است که ، اولویت نسبی و محدودیت های زمانی را در نظر نمی گیرد . به عنوان مثال وقتی داده ها از خط ارتباطی می رسند . باید به سرعت دریافت شوند تا فضایی برای داده هایی بعدی ایجاد شود اگر دومین واحد داده ، قبل از پردازش واحد داده اولی برسد ، دومین واحد داده ممکن است به دلیل پر بودن سرریز کردن میانگیر دستگاه I/O ، از بین برود .

در روش دوم ، برای وقفه ها اولویت در نظر گرفته می شود و وقفه ای با اولویت بالاتر می تواند روال اداره کنند مربوط به وقفه ای با اولویت پایین تر را دچار وقفه کند به عنوان مثالی از روش دوم ، سیستمی با سه دستگاه I/O را در نظر بگیرید . این دستگاه ها عبارت اند از چاپگر ، دیسک ، و یک خط ارتباطی که اولویت آن ها به ترتیب ۲ ، ۴ و ۵ است . برنامه کاربر در زمان $t=0$ شروع می شود . در زمان $t=10$ وقفه چاپگر رخ می دهد . اطلاعات کاربر در پشته سیستم قرار می گیرد و اجرا در روال خدماتی چاپگر (ISR) ادامه می یابد . در حین اجرای اینروال ، در $t=15$ وقفه ارتباطی رخ می دهد . چون اولویت خط ارتباطی بیش از چاپگر است ، این وقفه پذیرفته می شود ISR چاپگر وقفه می شود ،حالت آن در پشته قرار می گیرد ،و اجرا در ISR ارتباطی ادامه می یابد .در اثنای اجرای این روال ، در زمان $t=20$ وقفه دیسک رخ می دهد . چون اولویت این وقفه کمتر از اولویت وقفه ارتباطی کامل است ، منتظر می ماند و اجرای ISR ادامه می یابد تا کامل شود وقتی در زمان $t=25$ روال خدماتی ارتباطی کامل شد ،حالت قبلی پردازنده بازیابی می شود که اجرای ISR چاپگر است . اما ،قبل از این که حتی یک دستوراز این روال اجرا شود ،پردازنده ،وقفه دیسک را که اولویت بالاتری دارد می پذیرد وکنترل اجرا به ISR دیسک منتقل می شود . وقتی این روال در $t=35$ خاتمه یافت ، ISR چاپگر از سرگرفته می شود. وقتی روال چاپگر در زمان $t=40$ خاتمه یافت ،کنترل به برنامه کاربر باز می گردد.

چند برنامه ای

حتی با استفاده از وقفه ها نیز ممکن است پردازنده با کارایی بالایی استفاده نشود . اگر زمان لازم برای کامل کردن عملیات I/O، خیلی بیشتر از زمان لازم برای اجرای دستورات برنامه در بین دو فراخوانی I/O باشد (وضعیت متداول) ، پردازنده مدت زیادی بی کار خواهد بود . یک راه حل این مسئله این است که چندین برنامه کاربر بتوانند همزمان فعال باشند .

به عنوان مثال ، فرض کنید پردازنده دو برنامه برای اجرا دارد . یکی از برنامه ها داده ها را از حافظه می خواند و در دستگاه خارجی می نویسد . برنامه دیگر ، یک برنامه کاربردی است که محاسباتی را انجام می دهد پردازنده می تواند برنامه خروجی را شروع کند ، فرمان نوشتن را به دستگاه خارجی صادر کند ، و سپس شروع به اجرای برنامه دیگر نماید . وقتی پردازنده با چندین برنامه سروکار دارد ، ترتیب اجرای برنامه ها به اولویت نسبی و انتظار یا عدم انتظار برای عمل I/O بستگی دارد . وقتی برنامه ای دچار وقفه می شود و کنترل به اداره کننده وقفه منتقل می گردد ، پس از کامل شدن اداره کننده وقفه ، ممکن است فوراً به برنامه کار بردی که در آن زمان در حال اجرا بود ، برگردد در عوض . ممکن است کنترل به برنامه منتظر دیگری منتقل گردد که اولویت بالاتری دارد . برنامه ای که دچار وقفه شد وقتی اجرا می شود که بالاترین اولویت را داشته باشد . این مفهوم

که در آن چندین برنامه برای اجرا شدن نوبت می گیرند ، چند برنامه ای نامیده می شود .

I/O بدون میانگیر

I/O بدون میانگیر که همان DMA بین دستگاه و فضای فرآیند است ، سریع ترین روش اجرای I/O توسط فرآیند است . فرآیندی که I/O بدون میانگیر را انجام می دهد ، در حافظه اصلی قفل می شود و نمی تواند به خارج مبادله شود . به این ترتیب ، با قفل شدن بخشی از حافظه اصلی ، فرصت های مبادله و در نتیجه کارایی کلی کاهش می یابد . دستگاه I/O نیز در حین انتقال به فرآیند اختصاص دارد و فرآیندهای دیگر نمی تواند از آن استفاده کنند.

دستگاه های یونیکس

یونیکس پنج نوع دستگاه را شناسایی می کند :

➤ گرداننده های دیسک

➤ گرداننده های نوار

➤ پایانه ها

➤ خطوط ارتباطی

➤ چاپگرها

گرداننده های دیسک که به وفور در یونیکس استفاده می شوند ، بلوکی اند و می توانند توان عملیاتی بالایی را ارائه کنند . لذا ، به نظر می رسد I/O این دستگاه ها

بدون میانگیر یا از طریق حافظه نهان میانگیر انجام شود. عملکرد گرداننده های

نوار مثل گرداننده دیسک است و از طرح های I/O یکسانی استفاده می کنند.

چون پایانه ها و کاراکترها نسبتاً کند مبادله می کنند، I/O پایانه از صف کاراکترها

استفاده می شود به طور مشابه، خطوط ارتباطی نیاز به پردازش ترتیبی داده ها

بر روی ورودی و خروجی دارند و با صف کاراکترها بهتر اداره می شوند. نوع

I/O که برای چاپگر استفاده می شود، به سرعت آن بستگی دارد. چاپگرهای کند

از صف کاراکترها و چاپگر سریع از I/O بدون میانگیر استفاده می کند. حافظه

نهان میانگیر نیز می تواند برای چاپگر سریع استفاده شود. اما، چون داده ها

ارسالی به چاپگر دوباره استفاده نمی شوند، سربار حافظه می توانند از آن

میانگیر زاید است.

I/O در ویندوز ۲۰۰۰

مدیر I/O مسئول تمام I/O های مربوط به سیستم عامل و واسط یکنواختی

رافراهم می کند که تمام انواع گرداننده ها می توانند از آن استفاده کنند.

دستگاه I/O در یونیکس			
صف کاراکترها	حافظه نهان میانگیر	بدون میانگیر	
	×	×	گرداننده دیسک
	×	×	گرداننده نوار
×			پایانه ها
×			خطوط ارتباطی
×		×	چاپگرها

مولفه های I/O اصلی

مدیر I/O از چهار مولفه تشکیل شده است :

➤ مدیر حافظه نهان مدیر حافظه نهان ، برای کل زیر سیستم I/O ، استفاده از

حافظه نهان را اداره می کند . مدیر حافظه نهان ، یک خدمت حافظه نهان را

در حافظه اصلی برای کل سیستم فایل و مولفه های شبکه فراهم می کند.

میزان حافظه نهان تخصیصی به فعالیت خاص را می تواند با توجه به

مقدار حافظه فیزیکی موجود به طور پویا تغییر دهد . مدیر حافظه نهان

برای بهبود کارایی کلی ، دو خدمت ارائه می کند :

○ نوشتن با تبدلی سیستم ، نوسازی ها را فقط در حافظه نهان ثبت می کند ،

نه دیسک ، بعداً وقتی تقاضا برای پردازنده کاهش می یابد ، مدیر حافظه نهان

تغییرات را روی دیسک می نویسد . اگر در این مدت یک بلوک حافظه نهان

تغییر کند ، صرفه جویی خاصی انجام می شود .

○ تحویل دادن با تنبلی این خدمت مثل نوشتن با تنبلی برای پردازش تراکنش

است . به جای این که تراکنشی را فوراً به عنوان تراکنش کامل شده و موفق

علامت گذاری کند ، سیستم اطلاعات پردازش شده را در حافظه نهان قرار

میدهد و سپس با استفاده از یک فرآیند پس زمینه آن را در سابقه سیستم

فایل می نویسد .

➤ گرداننده سیستم فایل مدیر I/O باگرداننده سیستم فایل مثل سایر

گرداننده های دستگاه رفتار می کند و پیام مربوط به گرداننده های

خاص را به نرم افزار گرداننده مناسب آن آداپتور دستگاه هدایت می

کند .

➤ گرداننده های شبکه W2K شامل قابلیت های شبکه بندی مجتمع

است و از برنامه های کاربردی توزیعی پشتیبانی می کند .

➤ گرداننده ای دستگاه سخت افزار این گرداننده ها از طریق نقاط

ورودی در کتابخانه پیوند پویای مجری W2K به ثابت های دستگاه

های جانبی دستیابی دارند . برای هر سیستم کامپیوتری که توسط

W2K پشتیبانی می شود ، مجموعه ای از این روال ها وجود دارد .

زیرا نام روال ها برای تمام سیستم های کامپیوتری یکسان است و

کد منبع گرداننده های دستگاه W2K بر روی انواع مختلف پردازنده

های قابل حمل است .

I/O همگام و ناهمگام

W2K دو حالت همگام و ناهمگام را برای عملیات I/O پیشنهاد می کند. حالت

ناهمگام وقتی به کار می رود که امکان بهینه سازی کارایی برنامه کاربردی وجود

داشته باشد . در I/O ناهمگام ، برنامه کاربردی عملیات I/O را آغاز می کند و

سپس طی انجام عملیات I/O به پردازش می پردازد . در I/O همگام ، برنامه

کاربردی تا کامل شدن عمل I/O مسدود می شود .

I/O ناهمگام از دیدگاه نخ فراخوان کارآمدتر است ، زیرا در حالی که عملیات I/O

توسط مدیر I/O در صف قرار می گیرد تا بعداً اجرا شود ، نخ اجاره دارد به

اجرایش ادامه دهد . اما ، برنامه کاربردی که عملیات I/O ناهمگام را فراخوانی کرد ،

باید به طریقی کامل شدن عملیات I/O را بداند . W2K چهار تکنیک مختلف را برای

اعلان کامل شدن I/O ارائه می کند :

➤ سیگنال دهی به شیء هسته دستگاه در این روش ، وقتی عملی روی یک

شیء کامل شد ، نشانگر وابسته به آن شیء یک می شود . نخ درگیر

عملیات I/O می تواند به اجرا ادامه دهد تا به نقطه ای برسد که باید برای

کامل شدن عمل I/O صبر کند . در آن نقطه ، نخ می تواند تا کامل شدن

عمل I/O منتظر بماند و سپس ادامه دهد . این تکنیک و کاربردش ساده است

ولی برای اداره کردن چند درخواست I/O مناسب نیست . به عنوان مثال ،

اگر نخی به اجرای همزمان چند فعالیت روی فایل دیسک نیاز داشته باشد

، مثل خواندن از یک بخش فایل و نوشتن در بخش دیگر فایل ، در این تکنیک

نخ نمی تواند بین کامل شدن عمل خواندن و عمل نوشتن تمایز قائل شود .

فقط می تواند بفهمد که یک عملیات I/O بر روی این فایل کامل شده است .

➤ سیگنال دهی رویداد شیء هسته این تکنیک اجازه می دهد که چند عمل

درخواست I/O همزمان بر روی یک دستگاه یا فایل انجام شود . نخ برای

هر درخواست یک رویداد ایجاد می کند . بعداً ، نخ می تواند بر روی یکی از

این درخواست ها یا کل مجموعه درخواست ها مسدود شود .

➤ I/O قابل هشدار این تکنیک برای هر نخ یک صف در نظر می گیرد که صف

فراخوانی رویه ناهنگام (APC) نام دارد . در این مورد نخ I/O را درخواست

می کند و مدیر I/O نتایج این درخواست ها را در صف APC نخ فراخوان

قرار می دهد .

➤ پورت های تکمیل I/O این تکنیک در کارگزار W2K برای بهینه سازی

استفاده از نخ به کار می رود . در اصل ، مخزنی از نخ ها برای استفاده

فراهم هستند ، به طوری که برای اداره کردن درخواست جدید نیاز به

ایجاد نخ جدید نیست .

RAID نرم افزاری

W2K از دو نوع پیکر بندی RAID استفاده می کند که در {M96} به صورت زیر

تعریف شده اند :

➤ RAID سخت افزاری چند دیسک فیزیکی مجزا با استفاده از گرداننده

دیسک یا سخت افزار جعبه دیسک در یک یا چند دیسک منطقی ترکیب می

شوند .

➤ RAID نرم افزاری فضای ناپیوسته دیسک با استفاده از گرداننده دیسک

نرم افزاری تحمل خطا (FDISK) در یک یا چند بخش منطقی ترکیب می

شوند .

در RAID سخت افزاری ، واسط کنترل کننده ، ایچ اد یا بازسازی اطلاعات

افزونگی را اداره می کند RAID نرم افزاری که درکارگزار W2K موجود است

، عملکرد RAID را به عنوان بخشی از سیستم عامل پیاده سازی می کند و با هر

مجموعه ای از دیسک ها قابل استفاده است . امکانات RAID نرم افزاری ، RAID

سطوح ۱ تا ۵ را پیاده سازی می کند . در مورد RAID سطح ۱ (آینه سازی

دیسک) ، دو دیسک حاوی بخش های اصلی و آینه ای ممکن است در یک گرداننده

یا گرداننده های مختلفی وجود داشته باشند. پیکر بندی آخری ، تکرار دیسک نام

دارد .

مدیریت حافظه در ویندوز ۲۰۰۰

مدیریت حافظه ویندوز ۲۰۰۰ (W2K) چگونگی تخصیص حافظه و انجام صفحه بندی را کنترل می کند. مدیر حافظه طوری طراحی شد که در سیستم های مختلفی به کار گرفته شود و از صفحاتی به اندازه ۴ تا ۶۴ کیلو بایت استفاده می کند. اندازه صفحات در سیستم های MIPS, PowerPC, Intel برابر با ۴۰۹۶ بایت و در سیستم DEC Alpha برابر با ۸۱۹۲ بایت است.

نقشه آدرس مجازی W2K

هر کاربر W2K از فضای آدرس ۳۲ بیتی جداگانه استفاده می کند که هر فرآیند می تواند ۴ گیگابایت حافظه داشته باشد. به طور پیش فرض، بخشی از این حافظه برای سیستم عامل رزرو می شود. بنابراین کاربر می تواند از فضای آدرس ۲ گیگابایتی استفاده کند و تمام فرآیندها در ۲ گیگابایت از فضای سیستم شریک هستند. گزینه ای وجود دارد که اجازه می دهد فضای کاربر به ۳ گیگا بایت افزایش یابد و ۱ گیگا بایت در اختیار سیستم قرار گیرد. مستندات W2K نشان می دهد که این ویژگی برای پشتیبانی از برنامه های کاربردی متقاضی حافظه زیاد، که بر روی کارگزارانی با چندین گیگا بایت حافظه RAM اجرا می شوند و استفاده از فضای آدرس بزرگ تر منجر به افزایش شدید کارایی می گردد، مفید است (مثل کاربردهای پشتیبانی تصمیم گیری و جست و جوی داده ها).

➤ 0x00000000 تا 0x0000FFFF برای کمک به برنامه نویسان ، جهت به دست

آوردن تخصیص های اشاره گر تهی کنار گذاشته شده است .

➤ 0x00010000 تا 0x7FFFFFFF فضای آدرس کاربر ، این فضا به صفحاتی

تقسیم می شود که ممکن است به حافظه اصلی بار شوند .

➤ 0x7FFF0000 تا 0x7FFFFFFF صفحه نگهبانی که توسط کاربر قابل

دستیابی نیست . این صفحه ، کار بررسی مراجعه اشاره گرها به خروج

مرزها را برای سیستم عامل آسان میکند .

➤ 0x80000000 تا 0xFFFFFFFF فضای آدرس سیستم این فرآیند ۲ گیگا

بایتی برای مجری W2K ، ریز هسته و کنترل کننده دستگاه به کار می رود

صفحه بندی W2K

وقتی فرآیندی ایجاد شد ، می تواند از فضای ۲ گیگابایتی (منهای ۱۲۸ بایت

) استفاده کند. این فضا به چند صفحه با اندازه ثابت تقسیم می شود که هر کدام می

توانند به حافظه اصلی آورده شوند . در عمل ، برای سهولت حسابداری ، هر

صفحه می تواند در یکی از سه حالت زیر باشد :

➤ رزرو شده مجموعه ای از صفحات پیوسته که مدیر حافظه مجازی برای

فرآیندی کنار میگذارد ، ولی تا زمانی که مورد استفاده قرار نگیرند ، از

سهمیه آن فرآیند کم نمی شوند . وقتی فرآیند بخواهد در حافظه بنویسد ،

بخشی از حافظه رزرو شده به فرآیند تحویل داده می شود .

➤ متعهد صفحاتی که مدیر حافظه مجازی، فضایی را در فایل صفحه بندی

(مثل فایل دیسکی که صفحات پس از حذف شدن از حافظه در آن نوشته

می شوند) در نظر می گیرد.

تمایز بین حافظه رزرو شده و متعهد مفید است، زیرا (۱) مقدار فضای دیسک کنار

گذاشته شده برای فرآیند خاص را به حداقل می رساند و فضای آن دیسک برای

فرآیندهای دیگر خالی می ماند و (۲) نخ یا فرآیند را قادر می سازد تا میزان حافظه

ای را تعیین کند که در صورت نیاز می تواند سریعاً تخصیص یابد.

طرح مدیریت مجموعه مقیم در W2K، تخصیص متغیر و حوزه محلی است وقتی

فرآیند فعال می شود، تعداد معینی از قاب های صفحه حافظه به عنوان مجموعه

کاری آن تخصیص می یابد. وقتی فرآیندی به صفحه ای مراجعه می کند که در

حافظه نیست، یکی از صفحات مقیم آن فرآیند به خارج مبادله و صفحه جدیدی به

حافظه آورده می شود مجموعه های کاری فرآیندهای فعال توسط قواعد کلی زیر

تنظیم می شوند:

➤ وقتی حافظه اصلی پر است، مدیر حافظه مجازی به مجموعه های مقیم

فرآیندهای فعال اجازه رشد می دهد. برای این کار، وقتی خطای صفحه

رخ میدهد، صفحه جدیدی به حافظه آورده می شود، ولی صفحه قدیمی تر

به خارج مبادله نمی شود. به این ترتیب، به مجموعه کاری آن فرآیند یک

صفحه اضافه می شود.

جهت خرید فایل word به سایت www.kandoo.cn.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

➤ وقتی حافظه خالی کم باشد ، مدیر حافظه مجازی ، صفحاتی از مجموعه

کاری فرآیند فعال را که اخیراً کمتر مورد استفاده قرار گرفته اند ، از آن

مجموعه کاری حذف می کند تا حافظه برای سیستم بازیابی شود. به این

ترتیب اندازه آن مجموعه کاری کاهش می یابد .