جلسه سوم ذخیره و بازیابی اطلاعات:

بافرها:

بخشی از حافظه ی اصلی است که واسطه می شود بین بخش های داخلی کامپیوتر (CPU)، ورودی و خروجی ها.

علت استفاده از بافرها:

بالا بردن سرعت و هماهنگی کردن بین سرعت CPU و لوازم جانبی.

هر فایلی که باز می کنیم سیستم عامل یک بافر برای آن فایل در نظر می گیرد و بر ای خواندن و نوشتن دو بافر در نظر می گیریم و این بافرها را سیستم عامل در ناحیه جمع می کند و آن را ناحیه ی بافرها (Buffer Pool) می گویند.

روش های ایجاد بافر:

به سه روش ممکن است یک بافر ایجاد شود:

۱ـ به طور خودکار توسط سیستم عامل

۲ـ توسط برنامه نویس و با درخواست از سیستم عامل

۳ـ توسط برنامه نویس (سیستم فایل با استفاده از امکانات زبان)

روش های دستیابی به بافرها:

۱ـ روش انتقالی

۲ـ روش مکان یابی

در روش انتقالی سیستم فایل یا سیستم عامل اطلاعات را می خواند و داخل بافر می گذارد و یک حافظه ی کاری برای خودش دارد. اطلاعات مورد نظر از بافر منتقل می شود به حافظه ی کاری و بعداً در آن جا پردازش می شود و نتیجه بر می گردد.

در روش مکان یابی آدرس اطلاعات منتقل می شود به برنامه و برنامه می آید مستقیماً با اطلاعات داخل بافر کار می کند.

انواع بافر از نظر محل قرار گیری:

۱ـــ بافرهای سخت افزاری: آن بافری هستند که داخل سخت افزار قرار می گیرند و همچنین در دستگاه های جانبی.

۲ـ بافرهای نرم افزاری: آن بافری است که در حافظه ی اصلی قرار می گیرد و از طریق نرم افزار (برنامه ها) کنترل می گردد.

مدیریت بافرها:

در سیستم چند کاربره این واحد حتماً باید باشد که مدیریت روی اندازه ی بافرها ، تعدادشان و اندازه و زمان ایجاد و حذف و نوع بافر و ... کنترل این ها را مدیریت بافر گویند. در سیستم های تک کاربره هم اگر چند فایل بخواهیم داشته باشیم، مدیریت بافرها بسیار مهم است.

انواع بافرها از نظر ساختمانی (بحث در مورد بافرهای نرم افزاری است و بافرهای سخت افزاری قابل تغییر نیستند):

۱ـ بافرینگ ساده

۲ـ بافرینگ مضاعف

۳ـ بافرینگ چندگانه:

الف) صف معمولی ب) صف حلقوی (بافرینگ چرخشی)

۱ ـــ بافرینگ ساده تکی است و بافرینگ مضاعف دوتایی می باشد. بافرینگ ساده مدیریتش ساده است ولی سرعت عملیات را پایین می آورد و زمان انتظار برای CPU دارد.

۲ـ بافرینگ مضاعف مدیریتش پیچیده تر است ولی سرعت بالایی دارد.

۳ـــ بافرینگ چندگانه: چند بافر را در نظر می گیریم و در دو حالت یکی به صورت صف و دیگری به صورت حلقوی.

نرخ انتقال اطلاعات:

میزان اطلاعاتی که در یک ثانیه می شود منتقل گردد و واحدش بایت در ثانیه می باشد.

سرعت واقعی انتقال اطلاعات ← زمان درنگ دورانی (r) + زمان استوانه جویی (S) = زمان دسترسی تصادفی

زمان انتقال یک رکورد (Rtt):

$$Rtt = \frac{\text{طول رکورد R}}{\text{زمان انتقال بایت در یک ثانیه t}}$$

زمان انتقال یک بلاک (Btt):

$$Btt = \frac{\text{طول بلاک B}}{\text{نرخ انتقال اطلاعات t}}$$

$$\text{سرعت انتقال واقعی یک بلاک} = \frac{\text{اندازه بلاک B}}{\text{زمان انتقال بلاک Btt + زمان دستیابی str}}$$

هدف از طراحی سیستم های فایل دو مسئله است: ۱ـــ دست یابی سریع به اطلاعات ۲ـــ استفاده ی مفید از حافظه که سعی می کنیم این دو پارامتر را برای سرعت بیشتر تقویت کنیم.

ضابطه های اساسی در سیستم های فایل (طراحی فایل ها):

۱ـ حداقل بودن افزونگی:

در سیستم فایل تا آن جا که امکان دارد باید از اطلاعات تکراری کاست.

۲ـ دست یابی سریع به اطلاعات:

در فایل های خیلی بزرگ روش های معمولی جست و جو روش کندی می باشد و باید تکنیک هایی به کار برد که مستقیم به آن نقطه ی دلخواه برسیم.

۳ـ سهولت عملیات به هنگام سازی:

منظور این اســـت که فایل را از نظر ایجاد تغییرات در آن، وضـــعیت آخرین تغییرات را در آن ثبت کنیم یا تغییرات در فایل را به هنگام سازی نماییم.

۴ـ سهولت نگهداری سیستم:

سیستم در حین کار ممکن است یک سری تغییرات داشته باشد که بتواند به راحتی این تغییرات را اعمال نماید. مثلاً ورژن ها که تغییر می کنند، تغییرات به راحتی صورت گیرند. ۵ـــــ ضریب اطمینان بالا: یک سیستم مثل سیستم فایل که با اطلاعات سر و کار دارد باید ضریب اطمینان بالایی داشته باشد. مثلاً سیستم در موقع قطع برق حداقل از بین رفتن اطلاعات را داشته باشد. یا مثلاً در یک سیستم بیمارستان نباید اطلاعات غلط بدهیم؛ چرا که با جان انسان ها سر و کار داریم.

ملاک های ارزیابی فایل ها:

۱ـ متوسط اندازه ی رکورد که علاوه بر اطلاعات رکورد اطلاعات سیستمی در رکورد را نیز داریم.

۲ـ زمان لازم بر ای واکشی TF:

گرفتن اطلاعات از جایی

۳ـ زمان لازم بر ای به دست آوردن رکورد بعدی TN:

زمانی که لازم است تا رکورد بعدی را پیدا کنیم. چون به لحاظ فیزیکی رکوردها پشت سر هم نیستند.

۴ـ زمان لازم برای به هنگام سازی از طریق درج یک رکورد TI:

یعنی یک رکورد را ببریم سر جایش و به ترتیب قرار دهیم.

۵ـ زمان لازم بر ای به هنگام سازی از طریق تغییر یک رکورد TU:

مثلاً فیلدهای یکی از رکوردها را عوض کنیم. مثلاً دانشجویی که درسش را حذف کرده؛ و تغییراتی در رکورد بدهیم.

۶ـ زمان لازم برای خواندن همه ی فایل TX (خواندن سری):

مثلاً مسئول یک دانشگاه بخواهد لیستی از تمام دانشجویان داشته باشد، چقدر زمان لازم است؟

۷ـ زمان لازم برای سازمان دهی مجدد TY:

دوباره سازمان دهی کردن: مثلاً یک فایل را حذف کنیم و فایل دیگری را بنویسیم. لازم ا ست دوباره آن را مجدداً بازسازی و سازمان دهی کنیم که چقدر زمان می برد.

عملیات بخش منطقی :

عملیاتی که در یک فایل می توان انجام داد و سیستم فایل باید امکانات زیر را بدهد (عملیات بخش منطقی) و این عملیات از نظر User مطرح می باشد:

۱ـ دسترسی به رکورد مورد نظر (واکشی) TF

۲ـ به دست آوردن رکورد بعدی TN

۳ـ درج رکورد TI

۴ـ تغییر رکورد فعلی TU

۵ـ خواند فایل TX

۶ـ سازمان دهی مجدد TY

و این شش عمل را می توان در یک سیستم فایل انجام داد.

در بخش فیزیکی سه عمل کلّاً انجام می شود (این ها از نظر هد و دیسک مطرح اند):

۱ـ یافتن یا Seek

۲ـ خواندن یا Read

۳ـ نوشتن یا Write

شرح ضوابط مربوط به فایل ها:

۱ـ متوسط اندازه ی رکورد:

مثلاً رکوردی که برای اطلاعات یک دانشـــجو در نظر می گیریم غیر از این ســـاختارها یک ســـری اطلاعات سیستمی اضافه می شود و اطلاعات سیستمی که اضافه کنیم باید بدانیم که چقدر و چه مقدار می باشد و هرچه کمتر باشد رکورد کمتر می شود و کل فایل کمتر می شود.

در سطح رکوردها فایل ها را داریم که عبارتند از:

الف) متراکم: فایلی بدون فیلد خالی.

ب) فایل های پراکنده: فایلی با تعدادی فیلد خالی. مثلاً برای دانشـــجویان ده درس در نظر گرفته ایم. دانشجویی که سه درس را گرفته باشد بقیه ی فیلدها خالی می مانند و اندازه ی فایل بزرگ می شود چون که رکوردهایی وجود دارند که فیلدی از آن خالی است. ج) فایل هایی دارای افزونگی.

افزونگی :

فایلی که مقادیر بعضی از صفات خاصه اش چند بار تکرار شده باشند (در رکوردهای مختلف) دارای افزونگی می باشد. مثلاً که درس می گیرد صفات خاصه ی درس تکرار می شود و عنوان درس نیز چندین بار تکرار می گردد و یا نام استاد چند بار تکرار می شود که به این افزونگی می گویند.

افزونگی دو نوع است:

الف) طبیعی (به خاطر شرایط موجود) در محیط عملیاتی وجود دارد مثلاً شماره ی درس. و گاهی به خاطر مسائل تکنیکی که سیستم فایل می خواهد مجبوریم یک سری اطلاعات تکراری را نگه داریم.

ب) تکنیکی: به خاطر ایجاد یک استراتژی دستیابی خاص برای فایل تکرار بعضی یا تمام مقادیر یک یا چند صفت خاصه در محیط فیزیکی ذخیره سازی.

برای این که یک رکورد را پیدا کنیم چند روش Sort وجود دارد (روش های جست و جو در فایل یا استراتژی واکشی یک رکورد):

۱ـ دستیابی با جست و جوی خطی

۲ـ دستیابی با جست و جوی بلاکی

۳ـ دستیابی با جست و جوی باینری

۴ـ دستیابی با جست و جو به کمک شاخص در محیط ترتیبی

۵ـ دستیابی با جست و جو به کمک شاخص در محیط غیر ترتیبی

۶ـــ دستیابی با به دست آوردن آدرس از روی کلید (دستیابی مستقیم) یا از روی شماره ی نسبی رکورد در فایل

۷ـ دستیابی با استراتژی ترکیبی

هر چه پایین برویم سرعت زیاد می شود پیچیدگی نیز زیاد می شود و اتلاف حافظه نیز داریم. چون سرعت زیاد می شود به خاطر مکانیزم هایی که به کار می بریم.

۱ـ دستیابی با جست و جوی خطی:

از اول فایل شروع می کنیم و رکوردها را یکی یکی می خوانیم تا برسیم به رکورد مورد نظر که این کندترین روش و ساده ترین روش می باشد

۲ـ دستیابی با جست و جوی بلاکی:

فایل ها بلاک بندی شده باشند و بلاک ها نظمی نیز داشته باشند. اگر به بلاکی رسیدیم که در آن به نقطه ی مورد نظر رسیدیم خیلی خوب و سرعتش بهتر است چرا که مجموع رکوردها را با هم می خوانیم

۳ـ دستیابی با جست و جوی باینری:

ابتدا باید رکوردها مرتب شده باشند و با نصف کردن متوالی به رکورد مورد نظر می رسیم.

۴ـ دستیابی با جست و جو به کمک شاخص در محیط ترتیبی:

فایل ترتیبی و Sort شده است. علاوه بر آن یکسری اشاره گر جدا داریم که آن ها ما را هدایت می کنند به نقطه ی مورد نظر و شاخص کمک می کند که فایل را در یک محدوده ی کوچک بررسی نماییم.

۵ـ دستیابی با جست و جو به کمک شاخص در محیط غیر ترتیبی:

در محیط غیر ترتیبی چون نظمی ندارند در یک محدوده نمی شود بررسی شود.

۶ـ دستیابی با به دست آوردن آدرس از روی کلید:

با فایل هایی که به صورت تصادفی هستند سر و کار داریم و همان فایل های ترتیبی هستند با چگالی بالا. سرعت بالاست منتها اتلاف حافظه به همراه دارد.

۷ـ روشی از ترکیب شش روش بالا:

از دو یا چند استراتژی بالا استفاده می کنیم تا به رکورد مورد نظر برسیم.

دلایل کاهش کارایی فایل:

۱ـ از بین رفتن نظم ساختاری اولیه

۲ـ بروز حافظه های هرز در فایل

۳ـ بروز وضعیت نامطلوب در استراتژی دست یابی (شاخص دار)

عملیاتی که باید برای سازمان دهی مجدد انجام گیرد:

۱ـ خواندن تمام فایل

۲ـ خارج کردن رکوردهای حذفی

۳ـ فرمت بندی مجدد بلاک ها

۴ـ بازنویسی رکوردهای فعال

۵ـ بازسازی ساختار مربوط به استراتژی دست یابی (تصحیح فایلی که در آن آدرس ها نگهداری می شوند)

زمان بازنویسی بلاک ها str + btt

خواندن فایل ها به دو روش است:

۱ـ ترتیب منطقی رکوردها

۲ـ ترتیب فیزیکی رکوردها

به دست آوردن رکورد بعدی (Get Next):

این عمل را یک عمل ساختاری می گویند و از محتوای یک فیلد استفاده می کنیم و رکورد بعدی از رکورد فعلی به دست می آید. رکورد فعلی محتوایی و رکورد بعدی ساختاری است.

به هنگام سازی از طریق تغییر محتوای رکورد:

وقتی می خواهیم در فایلی در یک رکوردش تغییر ایجاد نماییم و نیازی به اضافه کردن رکورد جدید نداشته باشیم، دو حالت پیش می آید:

۱ـ یا با تغییر طول رکورد فایل تغییر نکند؛ این را به هنگام سازی درجا یا In Place می گویند. مثلا فقط اسم را عوض کنیم.

۲ـــ اگر مثلا شماره ی دانشجویی را عوض کنیم، ساختار فایل تغییر می کند چون Sort شماره ها به هم می خورد. اگر رکورد تغییر طول بدهد با تغییر طول یک فیلد؛ در این حالت به هنگام ســـازی را برون از جا می گویند که مجبوریم رکورد را از جا برداریم و جای دیگری قرار دهیم.

عمل Delete کردن در فایل یک عمل مستقل نیست و شکل خاصی از Update کردن می باشد.

موقعیت رکورد بعدی به رکورد فعلی می تواند یکی از سه حالت زیر باشد:

۱ـ هیچ ارتباطی بین آن ها وجود نداشته باشد.

۲ـ همجوار فیزیکی باشند.

۳ـ از رکورد فعلی به رکورد بعدی اشاره گری وجود داشته باشد.

زمان واکشی در یک فایل پایل برابر است با: برای بلاک $TF = \frac{1}{2} b \frac{B}{t'}$ و برای رکورد $TF = \frac{1}{2} n \frac{R}{t'}$

در ساختار فایل پایل $TN=TF$ ، چون ارتباط ساختاری بین رکورد فعلی و بعدی وجود ندارد.

عملیاتی که در بافر انجام می شود را در ارزیابی دخالت نمی دهیم چون زمان آن بسیار کم است.

حذف حالتی خاص از به هنگام سازی است؛ پس $TUD=TF+TRW$.

فایل پایل را نمی توان به صورت سریال خواند. چون بازیابی رکورد بعدی عملی نمی باشد ولی زمان خواندن پی در پی فایل برابر $2TF$ می باشد.

چه عواملی در ارزیابی متوسط اندازه ی رکورد دخیل اند:

۱ ـ بخش داده ای و غیر داده ای رکورد ۲ ـ متراکم یا غیر متراکم بودن فایل ۳ ـ وجود یا عدم وجود افزونگی در فایل

فایلی متراکم است که همه ی مقادیر صفات خاصه ی همه ی رکوردهایش مشخص باشد.

فایلی غیر متراکم است که بعضی از مقادیر بعضی از صفات خاصه ی بعضی از رکوردهایش موجود نباشد.

فایل را دارای افزونگی گوییم که مقادیر بعضی از صفات خاصه اش بیش از یک بار در محیط فیزیکی ذخیره شده باشند.

اگر N تعداد عنا صر مجموعه ای با شد که مقادیر صفات خا صه ی مورد نظر از آن گرفته شده ا ست، برای ذخیره سازی تمام این صفات به N بیت حافظه نیاز داریم.

واکشی یک رکورد دلخواه عملی است محتوایی و واکشی رکورد بعدی عملی است ساختاری.

عمل درج رکورد، مجموعه ای از عملیات لازم دارد که حجم آن بستگی به نوع ساختار فایل دارد.

عمل درج و به هنگام سازی، هر دو، محیط ذخیره سازی را تغییر می دهند.

حالاتی را که نیاز به خواندن تمام فایل می باشد، نام ببرید:

۱ـ سازمان دهی مجدد فایل ۲ـ ایجاد نسخه ای دیگر از فایل ۳ـ ایجاد یک استراتژی دستیابی

چه زمانی احتیاج به خواند تمام فایل نمی باشد؟ به هنگام سازی

موارد استفاده ی ساختار فایل پایل را بنویسید:

۱ـ در محیط هایی که در آن ها داده ها نظم پذیر نمی باشند ۲ـ بایگانی اطلاعات

به چه دلیلی فایل پایل سازمان دهی مجدد می شود؟ خارج کردن حافظه ی هرز

در ایجاد نسخه ی دیگری از فایل نیاز به سازمان دهی مجدد نمی باشد.

در یک فایل پایل عملیات لازم برای انجام عمل درج به ترتیب برابر است با:

۱ـــ خواندن آخرین بلاک فایل که سیستم آدرس آن را دارد ۲ـــ انتقال رکورد از ناحیه ی کاربری برنامه به بلاک ۳ـ بازنویسی بلاک

در فایل پایل بین رکورد بعدی و فعلی هیچ گونه ارتباط ساختاری وجود ندارد و واکشی رکورد بعدی به یک عمل واکشی نیاز دارد.