

فصل ششم

ساختارهای مبنایی فایل

ساختارهای مبنایی فایل

- اهداف اصلی هر سیستم فایل
 - سرعت عملیاتی در بازیابی و ذخیره‌سازی
 - صرفه‌جویی در حافظه
- برای رسیدن به دو هدف بالا در سیستم‌های ذخیره و بازیابی موارد زیر در نظر گرفته می‌شود
 - حداقل بودن میزان افزونگی (برای کاهش میزان حافظه مصرفی و کاهش هزینه بهنگام‌سازی)
 - دستیابی سریع (برای داشتن سرعت مطلوب در بازیابی و ذخیره‌سازی)
 - سهولت در بهنگام‌سازی (تا اطلاعات با کمترین هزینه بهنگام درآیند)

ساختارهای مبنایی فایل

- میزان رسیدن به اهداف نامبرده در سیستم فایل‌های مختلف متفاوت است
- هر سیستم فایل، فایلها بر اساس ساختار یا ساختارهایی ایجاد می‌کند
- ساختارهای مختلف در دو عمل اساسی، یعنی بازیابی و ذخیره‌سازی، رفتار متفاوت دارند
- رسیدن به اهداف نامبرده، در یک سیستم فایل مشخص، مستقیماً بستگی دارد به ساختارهای که در آن سیستم ایجاد و تامین می‌شود
- ما در این فصل ساختارهای مبنایی زیر را بررسی می‌کنیم
 - فایل با ساختار برهم (pile)
 - فایل با ساختار ترتیبی
- معرفی، بیان خصوصیات، ارزیابی کارایی

پارامترهای ارزیابی کارایی

۱. اندازه رکورد (متوسط مافظه لازم برای ذخیره‌سازی یک رکورد): R
۲. زمان لازم برای واکنشی یک رکورد دلفواه از فایل: T_F
۳. زمان لازم برای بازیابی رکورد بعدی: T_N
۴. زمان لازم برای بهنگام‌سازی از طریق درج یک رکورد: T_I
۵. زمان لازم برای بهنگام‌سازی از طریق ایجاد تغییر در یک رکورد: T_U
۶. زمان لازم برای خواندن کل فایل: T_X
۷. زمان لازم برای سازماندهی مجدد فایل: T_Y

پارامترهای ارزیابی کارایی

- اساساً شش نوع عمل روی فایلها، توسط سیستم فایل انجام می‌شود
 ۱. واکنشی رکورد
 ۲. بازیابی رکورد بعدی
 ۳. درج یک رکورد جدید (بهنگام‌سازی از طریق درج)
 ۴. تخریب در یک رکورد موجود (بهنگام‌سازی)
 ۵. خواندن تمام فایل
 ۶. سازماندهی مجدد

پارامترهای ارزیابی کارایی

- عملیات تخمیر دهنده محیط فیزیکی
 - درج
 - بهنگام‌سازی
 - مذف (مالت فاصی از بهنگام‌سازی)
 - سازماندهی مجدد
- انجام عملیات ششگانه باعث ۳ عمل اصلی در محیط فیزیکی می‌شود
 - مکان‌یابی
 - خواندن فیزیکی
 - نوشتن فیزیکی
- برنامه‌های کاربران مستقیماً به این سه عمل فیزیکی نمی‌پردازد و اینکه تا چه حد درگیر عملیات ششگانه هستند، بستگی دارد به سیستمی که بین برنامه‌ها و محیط فیزیکی ذخیره‌سازی واسط است

پارامترهای ارزیابی کارایی - متوسط اندازه رکورد

- در این ارزیابی تنها در نظر گرفتن بخش داده‌ای رکورد کافی نیست. بلکه عوامل دیگر را در محاسبه متوسط طول رکورد نیز در نظر گرفت.
- بنابراین عوامل موثر در این ارزیابی عبارتند از:
 - بخش داده‌ای رکورد
 - بخش غیرداده‌ای رکورد
 - متراکم یا غیر متراکم بودن فایل
 - پدیده افزونگی

پارامترهای ارزیابی کارایی - متوسط اندازه رکورد

- فایل متراکم:

- فایلی که تمام مقادیر همه صفات خاصه تمام رکوردهایش مشخص باشد

- فایل غیر متراکم:

- فایلی که برخی مقادیر صفات خاصه در برخی رکوردهایش مشخص نباشد

- افزونگی:

- فایلی دارای افزونگی است که مقادیر بعضی از صفات خاصه اش بیش از یکبار در محیط فیزیکی ، ذخیره سازی شده باشد

پارامترهای ارزیابی کارایی - واکنشی رکورد دلخواه

• واکنشی رکورد دلخواه

– یک عمل محتوایی: رکوردی که باید واکنشی شود، مقدار یک صفت خاصاش و گاه بیش از یک صفت خاصه به عنوان نشانوند جستجو داده می‌شود. لازمه این عمل:

• جستجو در فایل

• دستیابی به بلاک حاوی رکورد مورد نظر

• خواندن بلاک

– شیوه دستیابی عبارتند از: ترتیبی، تصادفی

– دستیابی از طریق: کلید اصلی، کلید ثانوی

پارامترهای ارزیابی کارایی - واکنشی رکورد دلخواه

- دستیابی ترتیبی: تعدادی بلاک به صورت پی در پی خوانده شود تا به بلاک مورد نظر برسد
- دستیابی تصادفی: با محاسبه آدرس، بلاک حاوی رکورد مورد نظر را به طور مستقیم می‌خواند

پارامترهای ارزیابی کارایی - بازیابی رکورد بعدی

• بازیابی رکورد بعدی

- رکورد بعدی مقداری: رکوردی است که مقدار نشانوند جستجوی آن، مقدار بعدی مقدار نشانوند جستجوی رکورد فعلی در نظم صعودی مقادیر نشانوند جستجو باشد (مفهوم رکورد بعدی در بحث لوکالیتی)
- مثال: اگر فایل دانشجو و درس را در نظر بگیریم، اگر فایل بر اساس نظم صعودی مقادیر شماره دانشجویی مرتب شده باشد و دانشجوی شماره x واکنشی شود، رکورد بعدی، رکورد دانشجویی است که مقدار شماره دانشجویی آن بلافاصله بزرگتر از مقدار x باشد
- رکورد بعدی مائز شرط: فایل دانشجو و درس را در نظر می‌گیریم، اگر در فایل دانشجو و درس مثلا اولین دانشجویی را بخواهیم که بیش از ۱۰ واحد گذرانده است را بازیابی کنیم، سپس درخواست رکورد بعدی بدهیم منظور رکورد دومین دانشجویی است که مائز همان شرط است

پارامترهای ارزیابی کارایی - بازیابی رکورد بعدی

- کاربر برای بازیابی رکورد بعدی نشانوند جستجو را نمی‌دهد، ساختار فایل باید به گونه‌ای باشد که امکان بازیابی رکورد بعدی را بدهد.
- یعنی با بازیابی رکورد فعلی بتواند رکورد بعدی را بدست آورد
- موقعیت رکورد بعدی به رکورد فعلی می‌تواند یکی از سه حالت زیر باشد:
 - همجوار فیزیکی باشند.
 - از رکورد فعلی به رکورد بعدی اشاره‌گری وجود داشته باشد.
 - هیچ ارتباطی بین آن‌ها وجود نداشته باشد.

پارامترهای ارزیابی کارایی - بهنگام سازی از طریق درج

- بهنگام سازی از طریق درج
 - درج یک رکورد جدید پس از لود اولیه آن در فایل است
 - رکورد باید در بلاکی وارد شود، بلاک باید یافته و خوانده شود
 - اینکه این بلاک، کدام بلاک باشد، دو حالت کلی زیر وجود دارد
 - رکورد باید در بلاک خاصی درج شود (اصطلاحاً نقطه منطقی درج)
 - بلاک خاصی مطرح نیست در هر جای فایل می تواند درج شود، معمولاً آخرین بلاک فایل است (با فرض جادار بودن، رکورد به انتهای فایل الماق شده و نشانگر انتهای فایل تخییر می کند)

پارامترهای ارزیابی کارایی-بهنگام سازی از طریق درج

• اصول عملیات درج

- یافتن و خواندن بلاکی که رکورد باید در آن درج شود
- جا دادن رکورد در بلاکی که در بافر است
- بازنویسی بلاکی
- در بعضی از ساختارها تنظیم نشانه‌روها

پارامترهای ارزیابی کارایی-بهنگام سازی از طریق تغییر در محتوی رکورد

- بهنگام سازی از طریق تغییر در محتوی رکورد

- تغییر مقدار یک یا چند صفت خاصه در رکورد

- رکورد متما باید واکنشی شود

- رکورد جدید(نسخه جدید) در بافر ساخته و در فایل نوشته شود

- انواع بهنگام سازی

- درجا (in place): رکورد بهنگام در آمده در جای قبلی اش نوشته می شود. همیشه امکان پذیر نیست، ممکن است طول رکورد در اثر بهنگام سازی تغییر کند

- برون از جا (out place): رکورد به هنگام در آمدنی در مکانی جدید نوشته می شود

پارامترهای ارزیابی کارایی-بهنگام سازی از طریق تغییر در محتوی رکورد

- اصول عملیات بهنگام سازی یک رکورد
 ۱. واکنشی رکورد بهنگام در آمدنی
 ۲. کار در بافر(ایجاد نسخه جدید)
 ۳. بازنویسی نسخه جدید(در جای قبلی) در بهنگام سازی در جا
 ۴. بازنویسی نسخه قدیم (با نشانگر حذف) در بهنگام سازی برون از جا و درج نسخه جدید در جای دیگر
 ۵. در بعضی ساختارها انجام عملیات پس از بهنگام سازی

پارامترهای ارزیابی کارایی - خواندن تمام فایل

- خواندن تمام فایل: در موارد زیر انجام می‌شود
 - درخواست کاربر مثلا برای لیست‌گیری (انجام پردازش خاص بر روی تمام رکوردها)
 - ایجاد نسخه دیگر فایل
 - سازماندهی مجدد

پارامترهای ارزیابی کارایی - خواندن تمام فایل

- خواندن فایل به دو صورت انجام می‌شود
 - پی‌درپی: یعنی بلاک به بلاک از آغاز فایل تا انتهای آن
 - سریالی: بر اساس نظم صعودی یک صفت خاصه معمولاً کلید
- اگر در ساختاری امکان ارزیابی رکورد بعدی وجود ندارد، خواندن سریال غیرممکن است
- خواندن پی‌در پی همان خواندن سریال است، اگر رکوردها در محیط فیزیکی به طور سریال ذخیره شده‌اند

پارامترهای ارزیابی کارایی - سازماندهی مجدد فایل

- سازماندهی مجدد فایل
- هر فایلی پس از لود اولیه دوره حیاتی دارد که در آن عملیات بازیابی و ذخیره‌سازی انجام می‌شود
- عملیات ذخیره‌سازی پس از لود اولیه محیط فیزیکی را تغییر می‌دهد
- در اثر تغییراتی که در فایل ایجاد می‌شود، ممکن است فایل کارایی اولیه‌اش را از دست بدهد
- دلایل کاهش کارایی فایل
 - از بین رفتن نظم ساختاری آغازین
 - بروز فضای هرز در هر فایل
- دلایل سازماندهی مجدد
 - امیا نظم آغازین
 - بازستانی فضای هرز

پارامترهای ارزیابی کارایی - سازماندهی مجدد فایل

- عملیات لازم برای سازماندهی مجدد فایل
 - خواندن تمام فایل (پی در پی یا سریال)
 - بلاک بندی مجدد رکوردها ضمن خارج کردن رکوردهای حذف شدنی
 - بازنویسی رکوردهای فعال
 - بازسازی ساختار مربوط به استراتژی دستیابی

زمان بازنویسی بلاک

- درج، بهنگام‌سازی ، سازماندهی مجدد
- زمان عمل نوشتن بلاک را که همان انتقال از بافر به دیسک است در یک ارزیابی کلی می‌توان مساوی زمان عمل خواندن بلاک در نظر گرفت

زمان نوشتن بلاک = زمان خواندن بلاک

زمان بازنویسی بلاک

- در اکثر مواقع رکورد در مکان قبلی‌اش نوشته می‌شود
- بازنویسی بلاک در مکان قبلی $= 2r$
- بلاکی که قرار است بازنویسی شود قبلاً خوانده شده و تخریب در محتوی آن داده شده است
- اگر زمان خواندن بلاک یعنی انتقال آن به بافر کوتاه و زمان انجام عملیات هم کوتاه باشد یعنی $C_B \ll 2r$ می‌توان بلاک در همان دور دیسک بازنویسی کرد

$$T_{RW} = \underbrace{2r - btt}_{\text{زمان لازم برای رسیدن مجدد آغاز بلاک به زیر نوک R/W}} + \underbrace{btt}_{\text{زمان انتقال از بافر به رسانه (نوشتن فیزیکی)}} = 2r$$

زمان لازم برای رسیدن مجدد آغاز بلاک به زیر نوک R/W

زمان انتقال از بافر به رسانه (نوشتن فیزیکی)

زمان بازنویسی بلاک

- اگر عملیات بافر به موقع انجام نگیرد یک دور دیگر باید منتظر بماند، پس T_{RW} یا یک دور یا دو دور دیسک باید منتظر بماند
 $T_{RW}=4r$
- اگر قبل از عمل بازنویسی استفاده دیگری از دیسک شود، مکان قبلی باید مجددا مورد دستیابی قرار بگیرد $C_B \gg 2r$

$$T_{RW}=T_F + C_B + b_{tt}$$

فایل با ساختار پایل

- رکوردها فاقد هر گونه نظم هستند
- بر اساس هیچ صفت خاصه‌ای مرتب نیستند
- رکوردها بر روی هم پیشته شده‌اند
- طول رکوردها ، تعداد فیلدها و مکان آنها متغیر است
- ساختار رکورد

$$A_1=V_1, A_2=V_2, A_3=V_3$$

$$A_1=V_1, A_3=V_3, A_4=V_4, A_5=V_5$$

$$A_2=V_2, A_4=V_4$$

اسم صفت خاص A_i

مقدار صفت خاص V_i

فایل با ساختار پایل

- چون رکورد قالب تعریف شده‌ای ندارد باید جفت اسم و مقدار ذخیره شود
 - باعث افزونگی داده می‌شود
 - از ذخیره اطلاعاتی که وجود ندارد جلوگیری می‌کند
- موارد استفاده
 - در محیط‌هایی که نظم‌پذیر نباشد، برای بایگانی
 - برای در نظر گرفتن ایمنی برای داده
 - مبنایی برای درک بهتر ساختارهای دیگر

فایل با ساختار پاییل - متوسط اندازه رکورد

- a تعداد کل صفت خاصه
- a' متوسط تعداد صفت خاصه
- A متوسط حافظه لازم برای اسم صفت خاصه
- V متوسط حافظه لازم برای مقدار صفت خاصه

$$R = a'(A + V + 2)$$

فایل با ساختار پایل - واکنشی رکورد

- نشانوند جستجو در درخواست به صورت $A_i = V$ داده می‌شود
- خواندن بلاک حاوی رکورد مورد نظر با جستجوی قطعی
- رکورد مورد نظر ممکن است اولین بلاک فایل باشد یا در آخرین بلاک فایل یا در هر بلاک دیگری
- به طور متوسط نصف بلاک‌های فایل خوانده می‌شود
- تعداد بلاکها b و اندازه بلاک B بایت

$$T_F = \frac{1}{2} * b * b'_{tt}$$
$$T_F = \frac{1}{2} * b * \frac{B}{t'}$$

$$T_F = \frac{1}{2} * n * \frac{R}{t'}$$

فایل با ساختار پایل – بازیابی رکورد بعدی

- رکورد بعدی، در این ساختار مفهومی ندارد
- زیرا ارتباط ساختاری بین رکورد فعلی و بعدی برقرار نیست

$$T_N = T_F$$

فایل با ساختار پایل-درج رکورد

- چون فایل فاقد هرگونه نظم است به انتهای فایل الحاق می‌شود. عملیات لازم عبارتند از
 - خواندن آخرین بلاک فایل که سیستم آدرس آن را دارد
 - کار در بافر
 - بازنویسی بلاک

$$T_I = s + r + b_{tt} + T_{RW}$$

$$T_I = s + r + b_{tt} + 2r$$

$$T_I = s + 3r + b_{tt}$$

فایل با ساختار پایل – عمل بهنگام سازی

- عمل بهنگام سازی به صورت بیرون از جا انجام می شود
- عملیات لازم
 - واکنشی رکورد بهنگام در آمدنی
 - ضبط نشانگر حذف شده در بخش پیشوندی نسخه قدیم
 - ایجاد نسخه جدید
 - بازنویسی نسخه قدیم (که نشانگر حذف دارد)
 - درج نسخه جدید در انتهای فایل

$$T_U = T_F + T_{RW} + T_I$$

$$T_{U_{delete}} = T_F + T_{RW}$$

فایل با ساختار پایل – خواندن تمام فایل

- باید تمام بلاکها خوانده شود (پی‌دی‌پی)

$$T_{Xseq} = 2T_F$$

$$T_{Xseq} = b * b'_{tt} = 2 * \frac{1}{2} b * b'_{tt} = 2T_F$$

- خواندن سریال فایل غیر ممکن زیرا بازیابی رکورد بعدی غیرممکن است، می‌توان فایل را روی یک صفت خاصه مرتب کرد و سپس خواند

$$T_{Xser} = T_{sort}(n) + T_{Xseq}$$

فایل با ساختار پایل – سازماندهی مجدد

- این فایل از لحاظ آمیای نظم آغازین نیاز به سازماندهی مجدد ندارد
- اما به خاطر نحوه عمل بهنگام سازی و همچنین در اثر عمل مذف، ممکن است فضای هرز به وجود آید
- o تعداد رکوردهای درج شده یک دوره میات فایل (از لود اولیه تا قبل از سازماندهی مجدد)
- d رکوردهای مذف شده (نشانگر مذف شده فورده باشند)

$$T_Y = (n + o) \frac{R}{t'} + (n + o - d) \frac{R}{t'}$$

زمان خواندن کل فایل

زمان بازنویسی کل فایل

فایل با ساختار پایل-معایب و مزایا

• مزایا

- اضافه کردن رکورد ساده و سریع
- خواندن پی‌درپی فایل ساده و سریع

• معایب

- واکنشی رکورد و بدست آوردن رکورد بعدی کند است
- ترتیبی خواندن بسیار کند
- چون برای حذف باید واکنشی انجام گیرد و واکنشی کند است پس برای عمل حذف نیز مناسب نیست

فایل با ساختار پایل - مثال

- فایلی با ساختار پایل شامل ۱۰۰۰۰۰ رکورد بوده و طول هر رکورد برابر ۴۰۰ اگر اندازه هر بلاک ۲۴۰۰ بایت و $b'_{tt} = 0.84ms$ باشد. زمان واکنشی یک رکورد را بدست آورید.

$$T_F = \frac{1}{2} * b * b'_{tt} = \frac{1}{2} * \frac{R}{t'} * n$$

$$Bf = \frac{B}{R} = \frac{2400}{400} = 6 \quad b = \frac{100000}{6} = 16667$$

$$T_F = \frac{1}{2} * b * b'_{tt} = \frac{1}{2} * 16667 * 0.84 = 7000ms = 7s$$

فایل با ساختار پایل – مثال

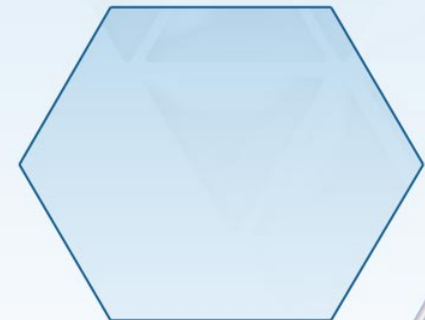
- فایل پایلی با ۱۰۰۰۰۰ رکورد ۴۰۰ بایتی داریم. مرتبا به ازای هر یک رکورد که حذف شده ۳ رکورد اضافه می‌شود تا هنگامی که تعداد رکوردهای فعال به ۱۵۰۰۰۰ برسد. اگر $B=2400$ بایت و $b'_{tt}=0.84$ باشد. زمان واکنشی یک رکورد پس از سازماندهی مجدد این فایل چند ثانیه خواهد بود؟

$$b = \frac{n * R}{B} = \frac{15000 * 400}{2400} = 25000$$

$$TF = \frac{1}{2} * b * b'_{tt} = \frac{1}{2} * 25000 * 0.84 = 10500ms = 10.5s$$

فایل با ساختار ترتیبی

- در لود اولیه تمام نمونه رکوردها بر اساس مقادیر یک صفت خاصه منظم هستند
- صفت خاصه معمولاً همان کلید اصلی است
- تمام رکوردها قالب از پیش تعریف شده دارند
- نیازی به ذخیره‌سازی اسم صفت خاصه نیست و تنها مقدار آن کافی است
- ساختار رکورد در فایل ترتیبی به صورت زیر می‌باشد



فایل با ساختار ترتیبی

- این ساختار نسبت به پایل مزایا و معایبی دارد
- مزایا
 - صرفه جویی در حافظه به خاطر عدم ذخیره‌سازی اسم صفت خاصه در رکوردها
 - ساده‌تر بودن قالب رکورد
 - نرفزار ساده‌تر برای ایجاد، مدیریت و پردازش فایل
 - وجود یک استراتژی دستیابی به علت داشتن نظم (الگوریتم جستجوی بهتری را می‌توان استفاده کرد)
 - پردازش سریال رکوردها تسریع و تسهیل می‌شود

فایل با ساختار ترتیبی

• معایب

– مصرف حافظه بیشتر به خاطر در نظر گرفتن اطلاعات نهست

– وجود پدیده عدم تقارن زیرا تنها یک صفت خاصه در عملیات روی فایل نقش دارد. استراتژی دستیابی ، متکی به همان صفت خاصه است

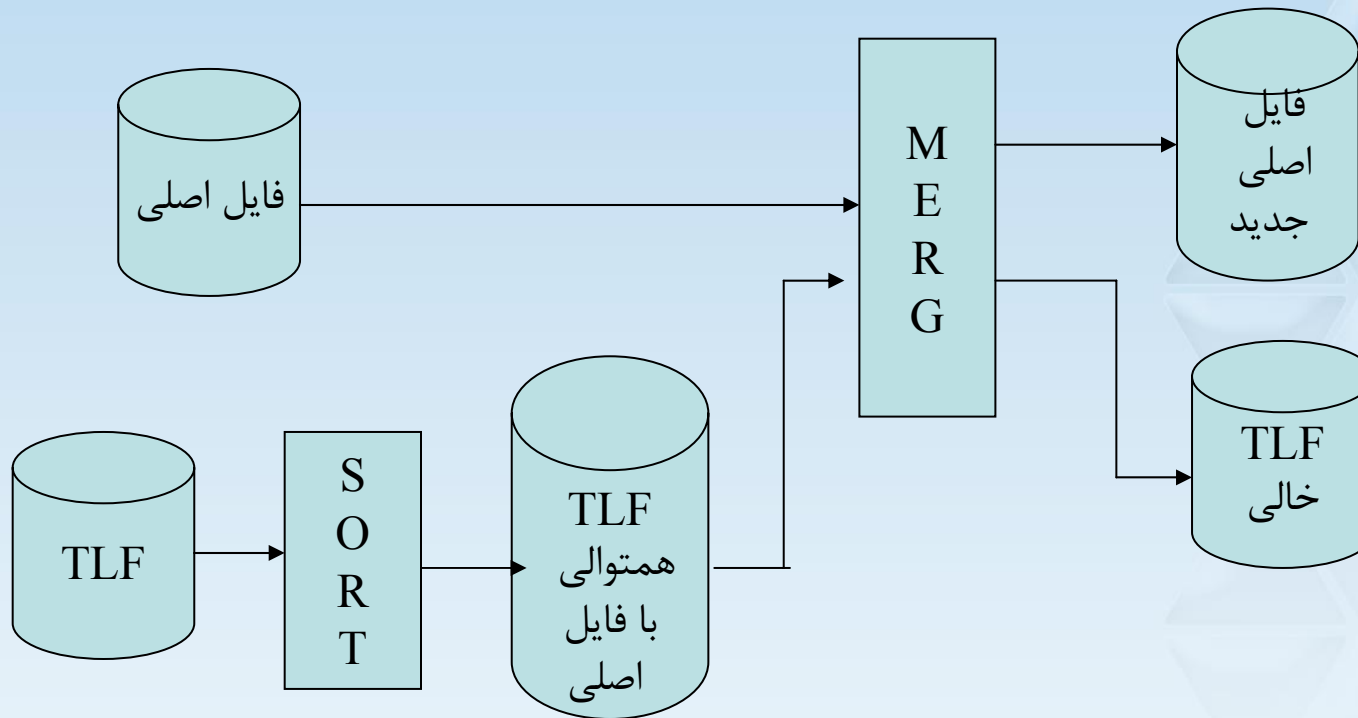
– کاهش انعطاف پذیری ساختار

- تغییر طول رکورد: در صورت تغییر طول رکورد فایل باید دوباره طراحی و ایجاد شود
- در عملیات درج، حذف، بهنگام سازی: به عنوان مثال درج تنها باید در نقطه منطقی درج انجام شود تا ساختار ترتیبی باقی بماند. انجام این کار در فایل های بزرگ بسیار زمانبر است

فایل با ساختار ترتیبی

- به دلیل کاهش انعطاف پذیری ساختار، فایل ترتیبی به صورت زیر در نظر گرفته می‌شود
 - فایل ترتیبی را عملاً یک فایل فواندنی ایجاد می‌کنند که نوشتن در آن انجام نمی‌گیرد
 - فایل جداگانه‌ای به نام فایل ثبت تراکنشها که تمام عملیات تخییر دهنده در آن انجام می‌گیرد
- بنابراین دو فایل داریم
 - فایل اصلی که ترتیبی ایجاد شده است
 - فایل ثبت تراکنشها (Transaction Logic File) TLF
 - یک فایل پایل است
 - حجم آن با توجه به حجم عملیات تخییر دهنده (میزان فعال بودن فایل) در نظر گرفته می‌شود
 - فایل TLF پس از مدتی در فایل اصلی ادغام می‌شود
 - رکورد جدید به انتهای فایل TLF اضافه می‌شود

فایل با ساختار ترتیبی



فایل با ساختار ترتیبی - موارد استفاده

- تغییر طول رکورد مطرح نباشد
- واکنشی سریع تک رکورد مد نظر نباشد
- وقتی رکوردها را به صورت یکجا پردازش کنیم
- پردازش سریالی به صورت پریودیک لازم باشد

فایل با ساختار ترتیبی-متوسط اندازه رکورد

- با توجه به ساختار رکورد داریم

$$R = a * v$$

$$S_{file} = n * a * v$$

- پس از لود اولیه باید ظرفیت فایل تراکنشها را نیز در نظر گرفت (فرض کرده‌ایم فایل تراکنشها ظرفیت 0 رکورد دارد)

$$S_{file} = (n + o) * a * v$$

فایل با ساختار ترتیبی-واکشی رکورد

- جستجو بر اساس صفت غیر کلید باشد،

- فایل تبدیل به پایل می‌شود. جستجو به صورت قطعی است

- فایل TLF، o رکورد در لحظه واکشی o' رکورد

$$T_F = \frac{1}{2} * (n + o') * \frac{R}{t'}$$

- جستجو بر اساس صفت کلید باشد

- جستجوی دودویی می‌توان انجام داد

- جستجو در دو سطح انجام می‌شود

- جستجو در فایل تا بلاک مورد نظر پیدا شود (واحد جستجو بلاک)

- هر بلاک که به بافر می‌آید، جستجوی دودویی درون بلاک

$$T_F = \log_2^b(s + r + b_{tt} + C_B) + T_{F_o} \quad T_{F_o} = \frac{1}{2} * o' * \frac{R}{t'}$$

فایل با ساختار ترتیبی-واکشی رکورد

- ابتدا فایل اصلی جستجو می‌شود در صورت عدم وجود رکورد سراغ فایل تراکنش‌ها رفته و به صورت خطی جستجو می‌کنیم
- تعداد رکوردهای فایل اصلی بیشتر باشد فاکتور اول غالب خواهد بود
- اگر تعداد رکوردهای فایل تراکنش بیشتر باشد فاکتور دوم غالب خواهد بود

فایل با ساختار ترتیبی-بازیابی رکورد بعدی

- با خواندن هر بلاک، B_f رکورد بدست می‌آید که هر یک بعدی رکورد قبلی است

$$T_N = \frac{B/t'}{B_f} = \frac{B/t'}{B/R} = \frac{R}{t'}$$

فایل با ساختار ترتیبی-درج

- رکورد درج شدنی در فایل اصلی درج نمی‌شود زیرا باید در نقطه خاصی درج شود که در فایل بزرگ زمانبر است
- رکورد در فایل TLF درج می‌شود تا در سازماندهی مجدد نظم مورد نظر بدست آید
- برای ارزیابی زمان درج دو حالت زیر را در نظر می‌گیریم
 - درج در فایل کوچک
 - درج در حالت کلی

فایل با ساختار ترتیبی-درج

• درج در فایل کوچک

– می‌توان در فایل کوچک نقطه درج را یافت و رکورد را در آن نقطه درج کرد. در این صورت بقیه رکوردها به سمت پایین شیفت داده می‌شود

– چون نقطه منطقی درج مشخص نیست به طور متوسط نصف رکوردها به سمت پایین شیفت داده می‌شود

– پس عملیات زیر لازم می‌باشد

• یافتن نقطه منطقی درج T_F

• درج رکورد در بلاک مورد نظر

• شیفت دادن رکوردها از نقطه منطقی درج به سمت انتهای فایل $\frac{B}{t'} + T_{RW}$

$$T_I = T_F + \frac{1}{2} * b * \left(\frac{B}{t'} + T_{RW} \right)$$

فایل با ساختار ترتیبی-درج

• درج در حالت کلی

– رکورد درج شدنی در آخرین بلاک فایل تراکنشها مانند درج در فایل پایل انجام می‌شود

$$T_I = s + 3r + B_{tt} + \frac{T_Y}{O}$$

– هر چند سازماندهی مجدد همزمان با درج انجام نمی‌شود اما پس از سازماندهی مجدد رکوردها واقعا در سر جای خود قرار می‌گیرد.

– سازماندهی مجدد این O رکورد به زمانی نیاز دارد که بین این O رکورد سرشکن می‌شود

فایل با ساختار ترتیبی - عمل بهنگام سازی

- اگر مقدار کلید عوض نشود می توان رکورد را به صورت درجا به هنگام کرد.
- ولی معمولا بهنگام سازی در فایل تراکنشها انجام می شود به این صورت که
 - رکورد بهنگام در آمدنی واکنشی می شود
 - عمل بهنگام سازی در بافر انجام می شود
 - رکورد بهنگام در آمدنی همراه یک رکورد کوچک دیگر در فایل تراکنشها درج می شود
 - در رکورد کوچک، تاریخ بهنگام سازی و نشانگر حذف شده وجود دارد تا مشخص کند نسخه قدیم رکورد در فایل اصلی حذف شدنی است

$$T_U = T_F + T_I$$

فایل با ساختار ترتیبی - خواندن تمام فایل

- زمان خواندن پی‌در پی برابر است با

$$T_{Xseq} = (n + o') * \frac{R}{t'}$$

- برای خواندن سریال باید فایل تراکنش‌ها مرتب شود

$$T_{Xser} = T_{sort}(o') + (n + o') * \frac{R}{t'}$$

- البته در عمل، اگر تعداد دفعات خواندن سریال طولانی باشد فایل اصلی را با TLF ادغام می‌کنند، یعنی سازماندهی مجدد انجام می‌شود و سپس فایل اصلی جدید که کاملاً ترتیبی است، خوانده می‌شود

فایل با ساختار ترتیبی - سازماندهی مجدد

- مرتب کردن فایل تراکنشها تا با فایل اصلی هم‌توالی شود
- خواندن فایل اصلی
- خواندن فایل تراکنش
- بلاک‌بندی مجدد رکوردها (ادغام طبق نظم) ضمن فارچ
- کردن رکوردهای حذف‌شده
- بازنویسی کل فایل

$$T_Y = T_{\text{sort}}(o) + n * \frac{R}{t'} + o * \frac{R}{t'} + (n + o - d) * \frac{R}{t'}$$