جلسه پنجم ذخیره و بازیابی اطلاعات:

فایل با ساختار پایل یا برهم:

این فایل ساده ترین ساختار را دا شته و رکوردهای آن بر ا ساس هیچ فیلدی مرتب نی ستند. طول رکوردها متغیر بوده و تعداد فیلدها و مکان آن ها در رکورد، در نمونه های مختلف ممکن ا ست متفاوت با شد بنابراین در کنار مقدار هر فیلد نام آن نیز نوشته می شود.

در بهترین حالت، نظم بین رکوردها، نظمی اســـت زمانی (ترتیبی Entry sequential) انگار رکوردها بر یکدیگر پشته شده اند.

شماره رکورد	
۱	۴۲=سن ، برق = رشته ، لیسانس = مدرک ، حسینی = نام
۲	فوق لیسانس = مدرک ، احمدی = نام ، ۳۸= سن
...	

فرم کلی ساختار هر رکورد به شکل زیر ا ست: A3=V3,…،A1=V1,A2=V2 که A1 نام فیلد (ا سم صفت خاصه)، و V1 مقدار آن فیلد (مقدار صفت خاصه) می باشد.

متوسط اندازه ی رکورد: فایل در Load اولیه n رکورد دارد.

کل تعداد فیلدها را a در نظر می گیریم و متوسط تعداد فیلدها را در یک رکورد با a' نشان می دهیم.

متوسط حافظه ی لازم برای هر فیلد را A بایت در نظر می گیریم و R=a'(A+V+2)

متوســـط حافظه ی لازم برای هر مقدار را V بایت در نظر می گیریم؛ پس داریم: برای علامت مســـاوی یک بایت، و برای جداســـازی نیز یک بایت در نظر می گیریم. پس برای کل فایل میزان حافظه n برابر می شـــود. عامل منفی لزوم تکرار اسم فیلدها در نمونه های مختلف رکوردها می باشد.

زمان واکشی در رکورد TF:

از کل ســـاختمان فایلی داریم، بهترین حالت حداقل یک رکورد را بخواند و بدترین حالت آن اســـت که کل رکوردها را بخواند. پس زمان واکشی برابر با زمان خواندن نصف فایل ها است به طور متوسط .

برای حالت بلاک بندی: $TF = \frac{1}{2} n \frac{R}{t'}$ برای حالت بدون بلاک بندی: $TF = \frac{1}{2} b \frac{B}{t'}$

$t' b/s$ =سرعت انتقال خواندن انبوه و b=تعداد بلاک ها و B=طول بلاک ها و n=تعداد رکوردها

$\frac{B}{t'}$ این قدر بایت بر ثانیه خوانده می شود $\leftarrow$ $nR = bB$ = طول فایل برحسب بایت

زمان به دست آوردن رکورد بعدی TN:

چون در این سازمان هیچ نظمی نداریم معلوم نیست که رکورد بعدی کجا ست و یک بار باید همه را بخواند یعنی TN=TF

زمان درج رکورد TI:

از آن جا که فایل پایل هیچ گونه نظمی ندارد، رکورد جدید همواره در انتهای فایل اضـــافه (Append) می شود.

۱ـ خواندن آخرین بلاک فایل از دیسک به بافر $s + r + btt$

۲ـــ اضافه کردن رکورد به بلاک خوانده شده. از آن جا که این عمل در حافظه صورت گرفته و بسیار سریع است، زمان آن را در محاسبات خود وارد نمی کنیم.

۳ـ بازنویسی بلاک مذکور

پس زمان درج در فایل پایل برابر اســـت با: $T_I = s + r + btt + T_{RW}$ و از آن جا که زمان عمل درج رکورد در بلاک موجود در حافظه اغلب کم تر از زمان یک دور زدن دیسک است پس: $T_I = (s + r + btt) + 2r$

زمان به هنگام سازی از طریق تغییر مقادیر فیلدها (Update):

به هنگام سازی درجا و برون از جا داشتیم و در این سیستم های فایل (پایل) مجبوریم از برون از جا استفاده کنیم.

۱ـ رکورد مورد نظر را واکشی می کنیم

۲ـ این رکورد را برای حذف علامت گذاری کنیم

۳ـ رکورد حذف شدنی را در جای خود بنویسیم

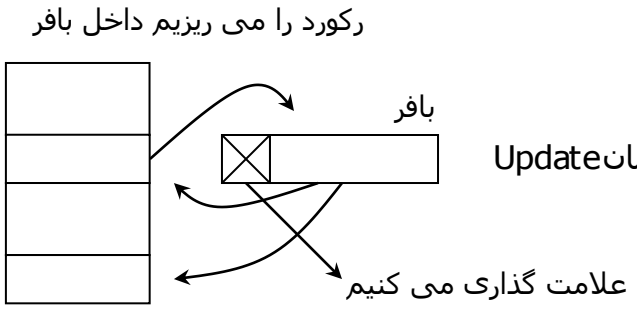
۴ـ رکورد را اصلاح می کنیم

۵ـ رکورد جدید را به فایل اضافه می کنیم.

(موارد ۱و ۲ و ۳، Delete یا نتیجه ی Update شده را می بریم ته فایل می نویسیم به خاطر سادگی کار یا سر جایش آن را قرار می دهیم با علامت گذاری)

موارد ۲و ۴ در بافر صورت می گیرد

۴ـ ایجاد دسته جدید است.



زمان Update $Tu = TF + T_{WR} + T_I$

محاسبه ی TD: از روی همین، عمل Delete کردن را می نویسیم و آن یک Update کردن درجا است.

حذف یک رکورد اینقدر زمان می گیرد: $T_{uD} = TF + T_{WR}$

زمان خواندن کل فایل Tx:

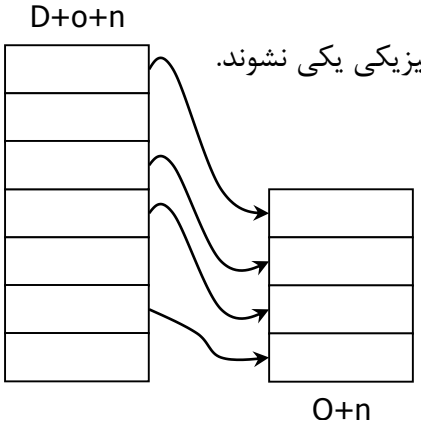
خواندن کل فایل به علت های زیادی خوانده می شود و خواندن فایل دو روش دارد.

۱ـ یا بر اساس ترتیب منطقی یا سریال Serial $T_{x\,seq} = 2TF$

۲ـ یا بر اساس ترتیب فیزیکی یا ترتیبی Sequential $T_{x\,ser} = nTF$ که زمان کمتری می برد

ممکن است موقعی که سریال می خوانیم مثلا اگر بر اساس ترتیب دانشجویی بخوانیم ممکن است در یک فیلد نداشته باشیم چرا که رکوردها نظمی ندارند و آن هایی را که نداریم ته فایل قرار می گیرند.

اگر بر اساس Sort کردن حساب کنیم و فایل Sort شده باشد، قبل از خواندن.



تذکر: Sort یعنی طوری رکوردها را عوض کنیم که ترتیب منطقی و فیزیکی یکی نشوند.

زمان سازمان دهی مجدد TY: $T_{x\,ser} = T_{Sort} + T_{x\,seq}$

برای این که یک نظم به فایل بدهیم و حافظه های هرز را از بین ببریم و فایل را کوچک کنیم.

حال فرض کنیم D رکورد حذف شدنی داشته باشیم و ۵ رکورد

منتقله و n تعداد رکوردهای اولیه در فایل:

در فایل جدید o+n رکورد داریم. n تعداد رکوردهای اولیه در فایل و o تعداد رکوردهای درج شده.

$$T_y = \underbrace{(n+o)\frac{R}{t'}}_{\text{زمان خواندن رکوردها}} + \underbrace{(n+o-D)\frac{R}{t'}}_{\text{زمان نوشتن رکوردها}}$$

ویژگی ها و کاربرد ها:

در هنگامی که رکوردهای کمی حذف می شوند استفاده از این فایل از نظر مصرف حافظه مؤثر است. اضافه کردن رکوردها در این ساختار ساده و سریع است.خواندن پی در پی این فایل ساده و سریع است. ولی این فایل در عملیات واکشی یک رکورد و به دست آوردن رکورد بعدی کند است. ترتیبی خواندن رکوردهای این فایل بسیار کند می باشد. فقط هنگامی که میخواهیم فایل را از ابتدا تا انتها بخوانیم (بدون هیچ ترتیب خاص) ساختار پایل از بقیه ی ساختارها مناسب تر است.

حذف رکوردهای تکراری در فایل پایل زمان بر است.هنگامی که اندازه ی فایل کوچک باشد، چگونگی ساختار فایل اثر زیادی در سرعت اجرای عملیات ندارد.