

بسم الله تعالى

ذخیره و بازیابی اطلاعات

Information Storage
&
Retrieval

سیستم و ساختار فایل
File Structures

تهیه و تنظیم : مهندس رحیم اکبری

تعیین حافظه :

به هر وسیله ای که توانایی حفظ اطلاعات (ذخیره سازی) را داشته باشد به گونه ای که کاربر (کاربر) بتواند در هر زمان به داده های مورد نیاز (دستیابی) از آن دستیابی (دسترسی) داشته باشد را حافظه می گویند.

محیط کامپیوتر : شامل دو محیط است : ۱- محیط درون ماشین ۲- محیط برون ماشین

۱- محیط درون ماشین:

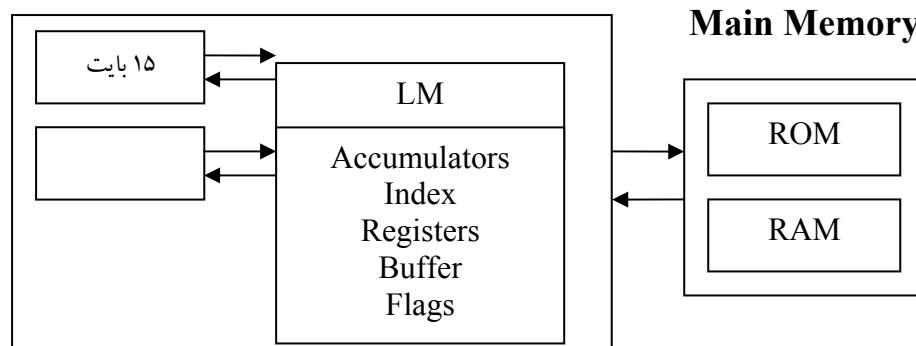
شامل خود ماشین با اجزا، و عناصر داخلی اش مانند:

شامل **Main Memory** -
(Random Access Memory) RAM

(Read Only Memory) ROM

شامل : (Central Processing Unit) **CPU** -
(Control Unit) CU
(Arithmetic Logical Unit) ALU
(Local Memory) LM
Flags, Buffer, Register, Index, Accumulator

CPU



۲- محیط برون ماشین:

شامل دستگاه های جانبی می باشند که برخی از این دستگاه ها برای ذخیره سازی اطلاعات نیز بکار می روند که شامل:

حافظه های جانبی (EXTENDED MEMORY)

- ۱ Magnetic Reel to Reel Drives
- ۲ Magnetic Cassette Drives
- ۳ Floppy Disk Drives
- ۴ Hard Disk Drives
- ۵ DVD Drives
- ۶ CD Drives

Input	Outrut
1.Card & Tape Readers	1.Card & Tepepunches
2. Audio Amplifiers	2.Printers
3.Laser Disk Drivers	3.Plotters
4. Transducers	4.Audio Speakers
5.Video	5.Machine Control Devices
6.Modem	6.Modem
7.Network	7.Network

حافظه ها به دو دسته تقسیم می شوند:

- حافظه اولیه (اصلی - درون ماشین): که پردازنده جهت اجرای برنامه ها مستقیماً با آن سروکار دارد.
- حافظه ثانویه (جانبی - برون ماشین): جهت ضبط اطلاعات و فایل ها به کار می رود.

*توجه:

ما در درس ذخیره و بازیابی با حافظه های جانبی سروکار داریم و در درس ساختمان داده هم با حافظه های اصلی سورکار داریم.

ویژگیها یا ((خصوصیات)) کلی حافظه ها :

۱- فرار و غیر فرار: حافظه هایی که با رفتن برق اطلاعات آنها از بین می رود ، حافظه های فرار (Volatile) و آنهایی که با رفتن برق داده های خود را حفظ می کند ، غیر فرار (Non Volatile) می باشند. حافظه اصلی اغلب فرار و حافظه جانبی غیر فرار می باشد.

۲- خواندنی یا خواندنی نوشتندی(R,R/W): بعضی حافظه ها مثل (RAM) یا هارد دیسک خواندنی نوشتند (R/W)، بعضی دیگر مثل (cd-rom) فقط خواندنی هستند.

به خواندن اطلاعات از حافظه اصلی به اصطلاحاً واکشی یا Fetch گفته می شود .

۳- آدرس دهی یا همان نشانه پذیری (Addressability): هر حافظه ای دارای یک شیوه آدرس دهی میباشد که به کمک آن خانه هایش مورد دستیابی قرار می گیرد. مثلاً حافظه (RAM) آرایه ای از بايتها است که هر کدام یک آدرس (عدد یکتا) دارند. یا هارد دیسک به صورت جفت عدد (شماره هد-شماره سیلندر-شماره سکتور) آدرس دهی می شود.

۴- ظرفیت (Capacity): هر حافظه دارای ظرفیتی است که به بیت و بایت و ... بیان می شود.

۵- زمان دستیابی (Access time): از لحظه ای که دستور خواندن و نوشتمن داده می شود تا لحظه ای که حافظه مورد نظر مورد دستیابی قرار می گیرد را زمان دستیابی میگویند. مثلاً زمان دستیابی حافظه (RAM) حدود ۱۲۰ نانو ثانیه و زمان دستیابی دیسک حدود ۳۰ میلی ثانیه است. یعنی از این نظر (RAM) در حدود ۲۵۰ برابر سریعتر از دیسک می باشد.

۶- نرخ یا سرعت انتقال (Transfer Rate): مقدار اطلاعاتی است که در واحد زمان از حافظه قابل انتقال است و یکی از واحدهای آن بایت در ثانیه است . مثلاً در یک ثانیه ۱۲۰ بایت داده از این حافظه منتقل می شود.

سلسله مراتب حافظه ها : (Storage Hierarchy)

بدلیل محدودیتها و مزایای حافظه ها که در زیر به بعضی از آنها اشاره می گردد سلسله مراتب حافظه ها پدید آمد:

۱- حافظه های اصلی (RAM) در حال حاضر همچنان محدودیت ظرفیتی دارند (مثلا حدود 128 MB یا 1 GB)

۲- فقط بخشی از اطلاعات برنامه های در حال اجرا لازم است در حافظه اصلی قرار گیرد و لازم نیست تمام بخش های همه برنامه ها به حافظه (RAM) آورده شود.

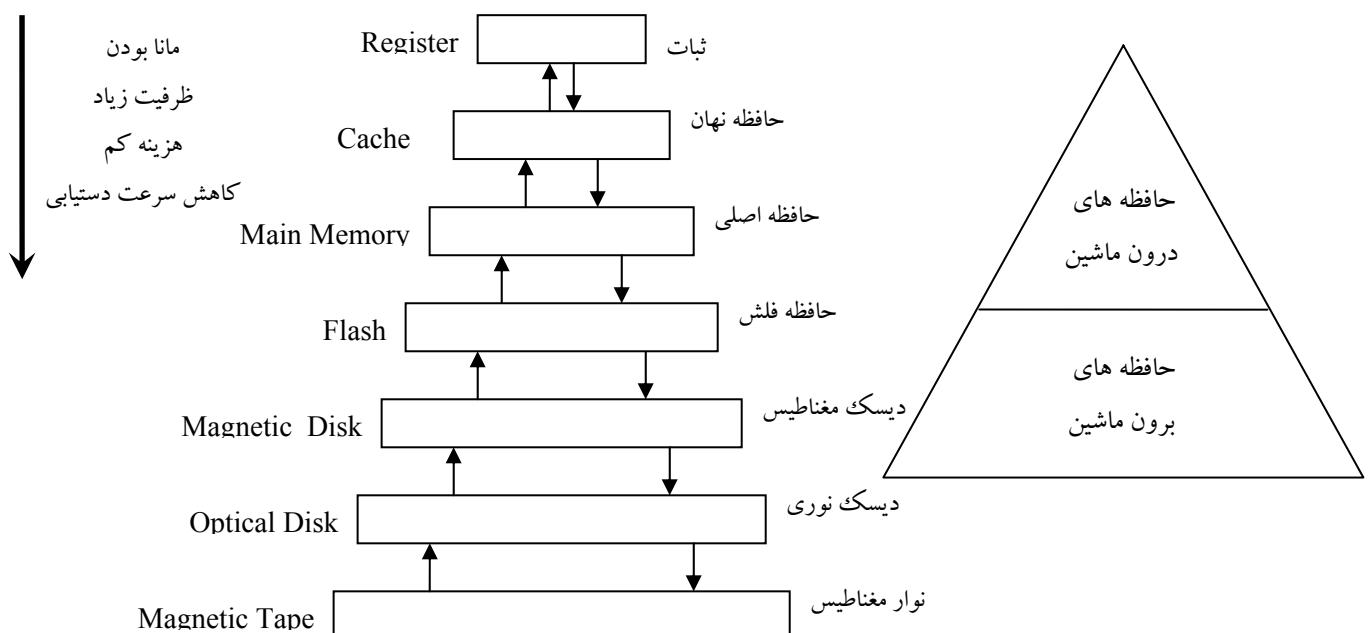
۳- حافظه های سریع ، گران هستند.

۴- حافظه های درون ماشینی اغلب فرار می باشند و داده های آنها با رفتن برق از بین می روند.

۵- بسیاری از برنامه ها به حافظه ای بیشتر از حافظه (RAM) نیاز دارند در این حال مثلا از بخشی از دیسک بعنوان حافظه مجازی (Virtual Memory) استفاده می شود.

۶- حافظه های درون ماشینی دارای سرعت زیاد ، حجم کم ، هزینه بالا و فرار نسبت به حافظه های برون ماشینی اند.

۷- ممکن است نیاز باشد چندین فرایند (پردازش Process) بصورت هم روند (Concurrent) به اطلاعات دسترسی داشته باشند برای این منظور کافی است آن اطلاعات را مثلا بر روی دیسک به اشتراک (share) گذاشت.



*توجه:

ما به چگونگی ذخیره سازی اطلاعات بصورت فایل ها در محیط برون ماشینی می پردازیم ، برای این منظور باید از رایج ترین رسانه های محیط برون ماشینی یعنی نوار و دیسک شناخت نسبی داشته باشیم.

منظور از ذخیره و بازیابی چیست؟

این درس با عنوان ذخیره و بازیابی اطلاعات (Information Storage and Retrieval) و نیز ساختار فایل ها (File Structures)

معروف است که منظور از ساختار فایل ، سازماندهی داده ها روی دستگاه های ذخیره سازی ثانویه است . به عبارت دیگر این درس به نحوه ذخیره داده ها در فایلها و عملیات لازم برای دستیابی به داده ها بر روی حافظه های جانبی می پردازد. اغلب برای

ذخیره و بازیابی از حافظه دیسک استفاده میگردد که مشکل اصلی در آن ، زمان نسبتاً زیادی است که برای بازیابی اطلاعات از دیسک لازم میباشد.

انواع حافظه های جانبی از نظر تکنولوژی ساخت :

- ۱- تکنولوژی الکترومکانیکی : کارت و نوار منگنه شدنی
- ۲- تکنولوژی الکترومغناطیسی : نوار مغناطیسی ، دیسک مغناطیسی و طبله (Drum)
- ۳- تکنولوژی الکتروپاتیک : دیسک نوری
- ۴- تکنولوژی الکترومغناپتیک(نوری): دیسک (مغناطیسی - نوری) (MO)(Magnetic - Optic)

نوار مغناطیسی:

نوار مغناطیسی اصولاً برای پردازش ترتیبی یا پی در پی (Sequential) اطلاعات استفاده می گردد از نظر تکنولوژی ساخت نوار های مغناطیسی به ۴ دسته تقسیم می شوند:

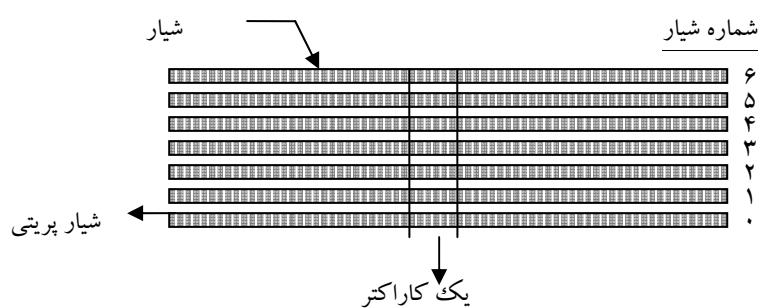
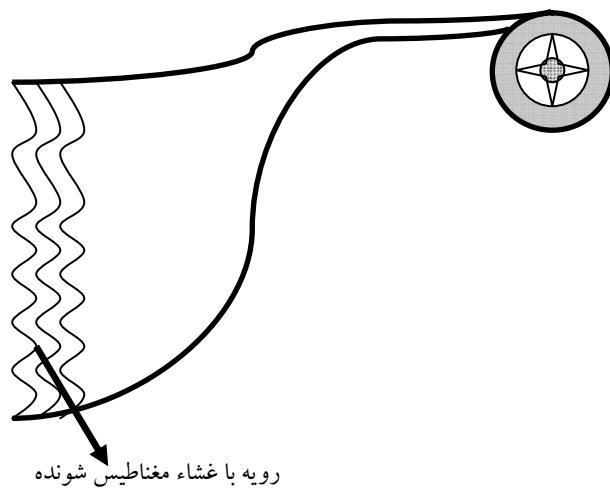
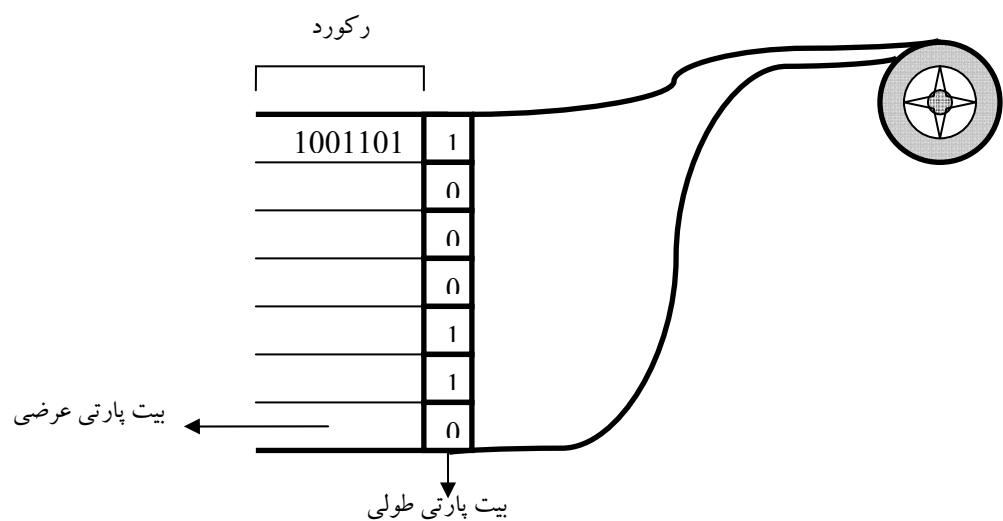
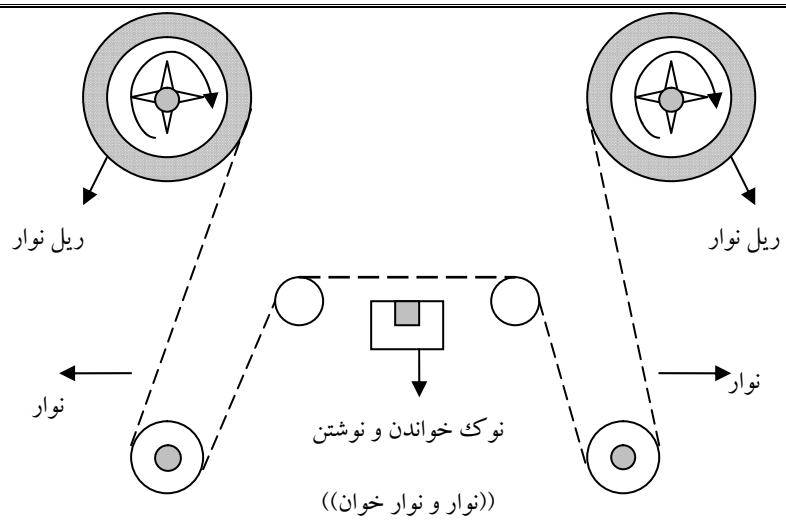
- ۱- ریل به ریل
- ۲- نوار کارت ریج
- ۳- نوار کاست
- ۴- نوار صوتی

فرق نوار کارت ریج با نوار ریل به ریل:

نوار کارت ریج در یک محفظه پلاستیکی قرار داده شده تا در مقابل گرد و خاک محافظت گردد.

نوار و نوار خوان:

نوار دارای غشاء مغناطیس شونده با شیار (Track) هایی که از نظر تعداد دو نوع میباشد ، نوار ۷ شیاره و نوار ۹ شیاره که یکی از شیار های نوار جهت کنترل خطأ (بیت پریتی) استفاده می شود ، دو نوع بیت پریتی وجود دارد ، یکی بیت پریتی طولی ، یکی بیت پریتی عرضی .

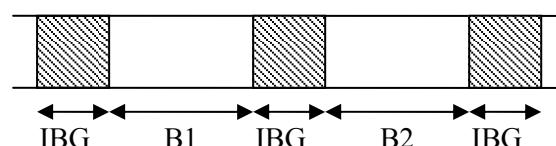


چگالی (Density)

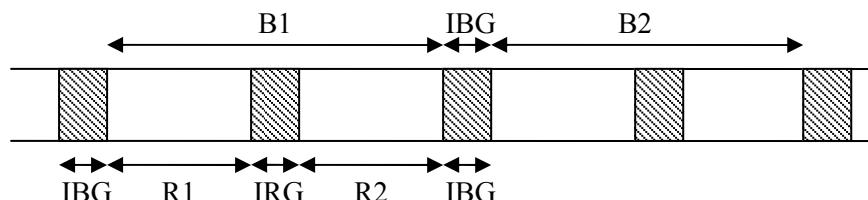
به تعداد بیت های قابل ضبط در هر اینچ نوار ، چگالی گفته می شود. چگالی در نوار با واحد بیت در اینچ (bpi) بیان می گردد که با توجه به نحوه ذخیره عرضی کاراکتر ها در شیار ها ، معادل همان بایت در اینچ یا کاراکتر در اینچ می باشد.

گپ (Gap)

به فضای بلاستفاده یا هرز (Waste) بین دو بلک یا دو رکورد ، گپ بین بلکها (IBG) یا گپ بین رکوردها (IRG) گفته می شود.



((گپ بین بلک ها))

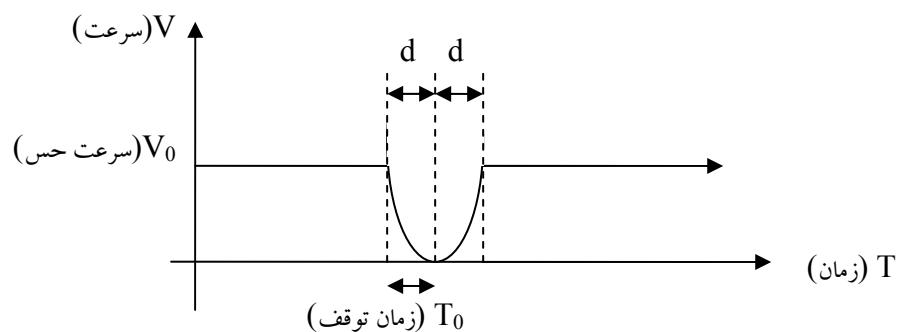
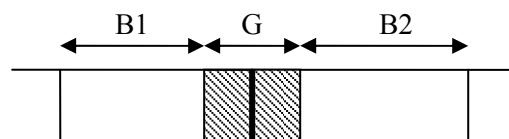


((گپ بین رکوردها))

-فضای گپ جهت توقف هد یا حرکت دوباره آن استفاده می شود.

سرعت حس:

برای آنکه هد بتواند داده های موجود بر روی نوار را بخواند لازم است به سرعت مناسب و یکنواختی به نام سرعت حس برسد ، از طرفی هنگام توقف ، فضایی جهت رسیدن سرعت حس به سرعت صفر و فضایی هم برای رسیدن به سرعت یکنواخت مورد نیاز است.



$$\text{سرعت متوسط} = \frac{v_0}{2} = \frac{d}{t_0} \Rightarrow t_0 = \frac{2d}{v_0} \Rightarrow \left(t_0 = \frac{G}{V_0} \right)$$

زمان توقف :

$\frac{\text{Inch}}{\text{Sec}}$ واحد سرعت حس :

مثال:

اگر $G = 0.9$ inch باشد زمان t_0 برابر خواهد بود با:

$$t_0 = \frac{G}{V_0} = \frac{0.9}{450} = \frac{9}{4500} = \frac{1}{500} = \frac{2}{500} = \frac{2}{1000} = 2ms$$

میلی ثانیه

مثال:

اگر سرعت حس نوک یک نوار برابر $V_0 = 150 \frac{\text{inch}}{\text{Sec}}$ و زمان رسیدن به این سرعت 50 ثانیه باشد، IBG چقدر است؟

نحوه ذخیره سازی فایل روی نوار:

فایل ها به صورت بلاکهایی به شکل پشت سر هم روی نوار ذخیره می شود که هر بلاک مجموعه ای از رکوردها می باشد. چندین فایل را می توان روی یک نوار ذخیره کرد که در این حالت هر فایل یک علامت ابتدایی فایل (BOF) و یک علامت پایان فایل (EOF) دارد.

پارامتر های نوار:

الف: پارامترهای ظرفیتی که عبارتند از:

۱. چگالی (با واحد bpi)
۲. طول نوار (غالباً با واحد فوت)
۳. اندازه (IBG) (معمولاً بین 0.3 تا 0.75 اینچ است)

ب: پارامتر های زمانی که عبارتند از:

۱. سرعت لغزش نوار (با واحد اینچ در ثانیه)
۲. نرخ انتقال (با واحد بایت در ثانیه)
۳. زمان حرکت - توقف (بر حسب میلی ثانیه)

*توجه:

نرخ انتقال به دو صورت می باشد:

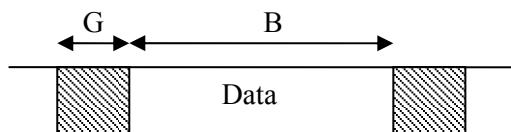
۱. اسمی (Nominal Rate): توسط کارخانه سازنده بیان می شود

طبق فرمول می توان محاسبه کرد: $S_n = L \times D$

D: چگالی

L: طول به اینچ

S_n : ظرفیت اسمی



قابل محاسبه است : با توجه به این که به ازای (B) بایت داده باید (B+G) بایت از ظرفیت نوار مصرف شود لذا در صد

$$\text{استفاده واقعی از نوار} : \frac{B}{B+G} \times 100$$

$$S_E = \frac{B}{B+G} \times S_N$$

تذکر ۱: می بینیم که طول گپ و طول بلاک در میزان واقعی استفاده از نوار تاثیر دارند.

$$S_E = \frac{B - W'_B}{B - W'_B + G} \times S_N$$

به جای (B) معادل $(B - W'_B)$ را می گذاریم که (W'_B) میزان حافظه هر ز به ازء یک بلاک می باشد (غیر از گپ)

تذکر ۲: امروزه از نوار ها برای بایگانی و آرشیو استفاده می شود.

نوار کاست:

این دستگاه کاملاً شبیه نوارهای صوتی متداول است . نوعی از آن دارای عرض 0.15 اینچ ، طول 100 تا 150 فوت بوده و ارزانترین حافظه هایی است که عمدها در مینی و کامپیوتر های کوچک کاربرد دارد.

ضبط کننده های صوتی تطبیق داده شده با کامپیوتر :

در اکثر کامپیوتر های شخصی و تجاری از ضبط کننده های صوتی ارزان برای ضبط داده ها استفاده می شود . سرعت گونه ای از این نوار ها 1.875 اینچ در ثانیه بوده ، کند حرکت می کند . طول اینگونه نوار 562 فوت و ظرفیت آن 500,000 بایت در هر طرف است . این نوع نوار ها در ابعاد دیگری هم وجود دارد .

نوار کارتریج:

تفاوت آن با نوارهای ریل به ریل این است که این نوارها در یک محفظه پلاستیکی جای دارند تا از تماس محیط خارجی و گرد و خاک محفوظ بمانند. این گونه نوار های نیز دو ریل دارند که نوار بر آنها می لغزد. انواع استانداردی از این گونه نوار ها موجود است از جمله:

- نوار نوع 300: طول 300 و 450 فوت ، عرض 0.25 اینچ ، ابعاد محفظه $4*0.7*0.25$ اینچ
- نوار نوع 100: طول 140 فوت ، عرض 0.15 اینچ ، ابعاد محفظه $3.2*0.5*0.15$ اینچ

به نوع دوم گاهی مینی کارتریج می گویند . نوع اول با چهار شیار ، 4.3 MB و نوع دوم با دو شیار ، 6.4 KB ذخیره می کند . در نوع دیگری از کارتریج به نام QIC ، عرض 6.27 mm و ظرفیت آن 40 MB تا 10 GB است.

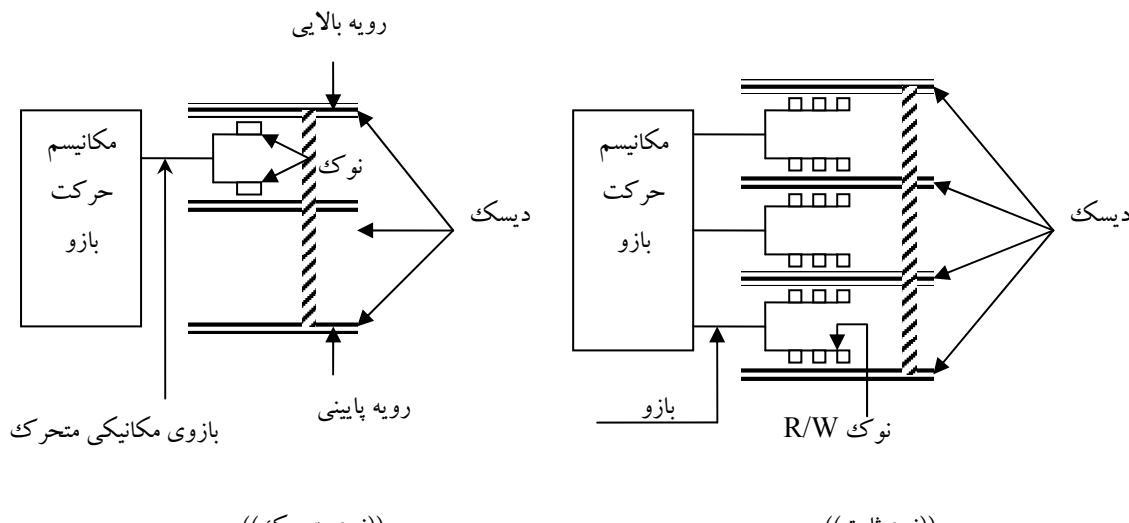
دیسک مغناطیسی: ۱- هارد دیسک ۲- فلاپی دیسک

نحوه دستیابی به داده های ذخیره شده در آن بصورت تصادفی می باشد که اصطلاحاً به آن (Direct Access Device) DASD گفته می شود . از جنبه های مختلفی می توان آنرا بررسی کرد :

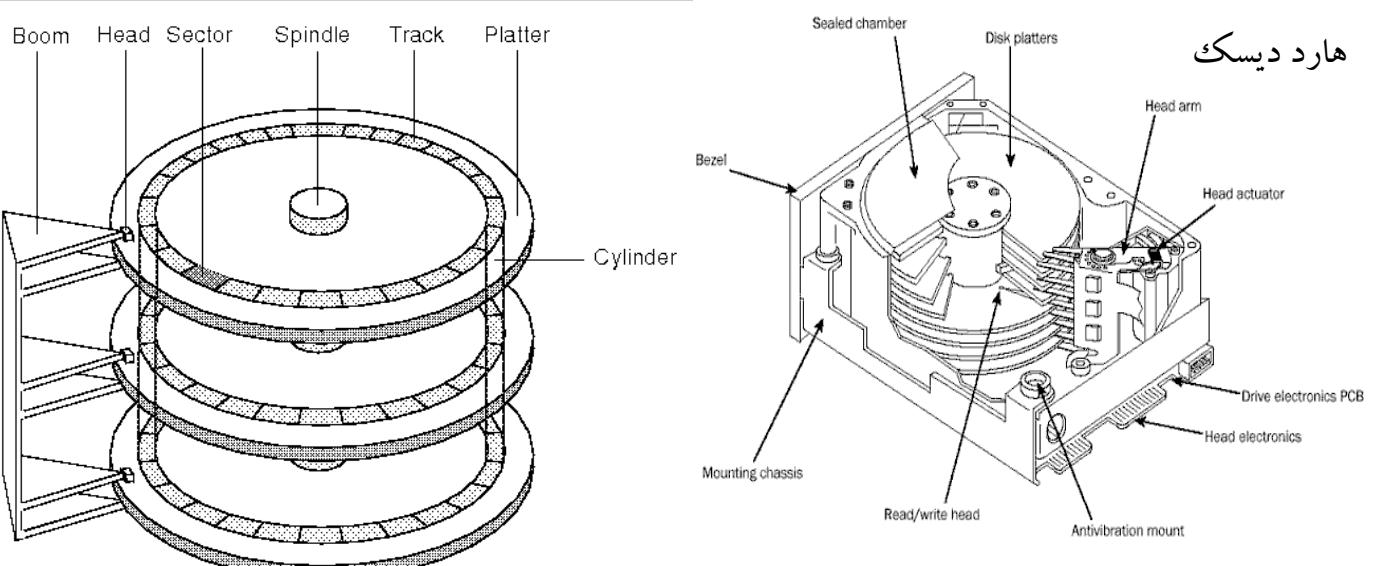
۱- از نظر توانایی جابجا شدن دیسکها به دو دسته ثابت (Fixed) و قابل جا به جایی تقسیم می شود.

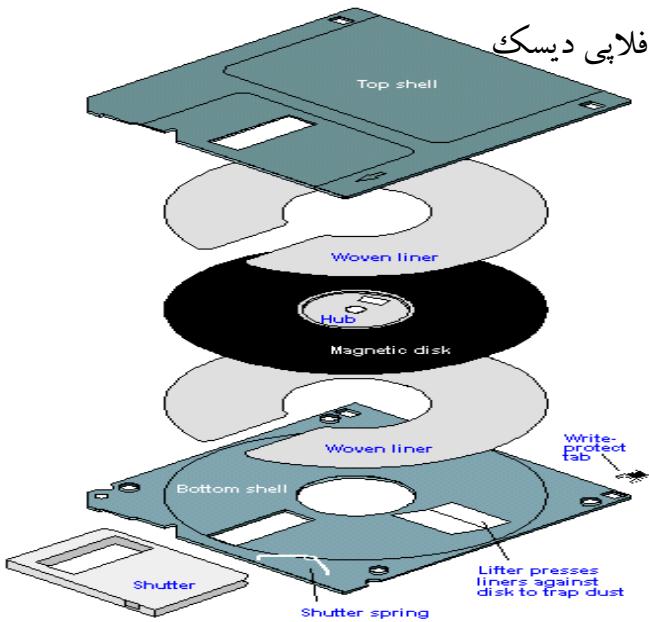
دیسک مغناطیسی ثابت مانند: هارد دیسک ، دیسک مغناطیسی جا بجا پذیر مانند فلاپی

۲- از نظر هد به دو دسته هد ثابت (Fixed Head) و هد متحرک تقسیم می شوند . در اغلب دیسک های امروزی هد می تواند در راستای شعاع حرکت کرده و از شیاری به شیار دیگر برود . در دیسک هایی با هد ثابت هر شیار برای خود هدی دارد و بدین ترتیب بازوی هد حرکت نمی کند . دیسکهای با هد ثابت سریع تر و گرانتر می باشند.



۳- از نظر جنس صفحه به دو دسته دیسک سخت و دیسک فلاپی تقسیم می شوند . در دیسک سخت (Hard Disk) صفحات از جنس آلمینیوم بوده و در فلاپی ها (Floppy Disk) از نوع پلاستیک می باشد





۴- از نظر تعداد صفحات به دو دسته تک صفحه ای (Single Platter) مانند: فلاپی دیسک و چند صفحه ای (Multiple Platters) مانند هاری دیسک تقسیم می شوند. دیسک های چند صفحه ای را گاهی اوقات پک (Pack) می نامند. دیسکی با (n) صفحه تعداد ($2n$) رویه دارد که گاهی رویه های بالایی و پائینی جهت حفاظت استفاده شده و ($2n-2$) رویه دیگر جهت ذخیره سازی بکار می روند در بعضی از دیسک ها تمام ($2n$) رویه برای ضبط اطلاعات استفاده میگردد.

تقسیمات دیسک مغناطیسی :

شیار (Track): به دوایر هم مرکز بر روی رویه ها ، شیار گفته می شود.

استوانه (Cylinder): به شیار های هم شعاع بر روی رویه های مختلف ، استوانه گویند.

قطاع یا سکتور (Sector): هر شیار از تعدادی سکتور تشکیل یافته است.

*توجه:

شیار ها معمولاً از بیرون به سمت داخل و از صفر شماره گذاری می شوند. بیرونی ترین شیار ، شماره صفر دارد.

در شیارهای داخلی بیتها به همدیگر نزدیکترند. در هر سکتور اغلب نیم کیلو بایت یا (512) بایت اطلاعات ذخیره می شود.

طول سکتورها در شیارهای خارجی بیشتر از داخلی است و در نتیجه چگالی اطلاعات در شیارهای داخلی بیشتر است بعارت

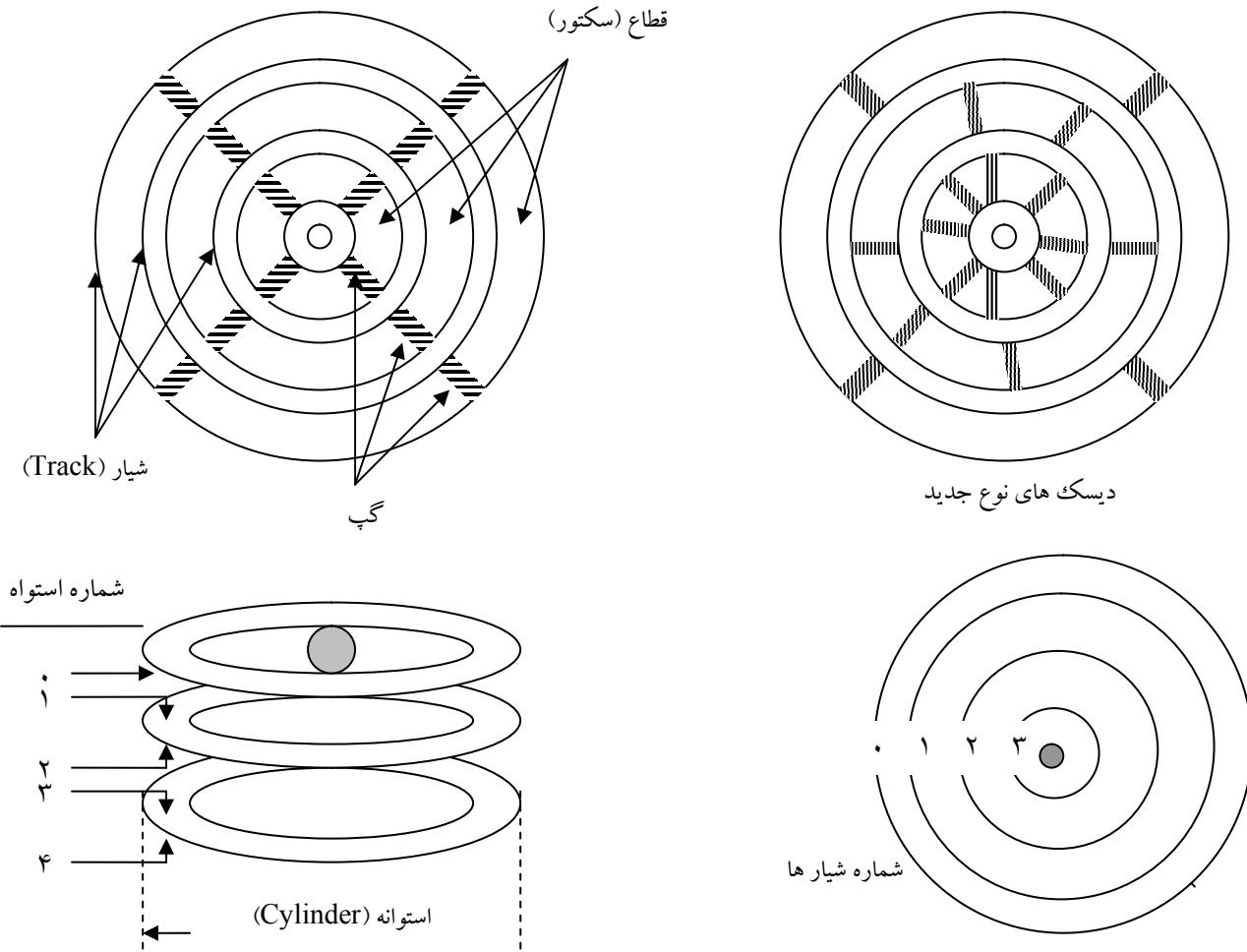
دیگر در شیارهای خارجی تر مقداری از فضای اطلاعاتی به هدر رفته است. برای از بین بردن این حالت در دیسکهای امروزی

که IDE (Integrated Drive Electronics) نام دارد سطح دیسک را به منطقه هایی (Zone) تقسیم می کنند که

هر (Zone) شامل تعدادی شیار است تعداد سکتورهای موجود در هر (Zone) نسبت به (Zone) دیگر متفاوت است معمولاً تعداد

سکتورهای (Zone) های بیرونی بیشتر از تعداد سکتورهای (Zone) های درونی است. این تقسیم بندی توسط مدار کنترل کننده

روی هارد دیسک بگونه ای شبیه سازی می گردد که سیستم عامل و برنامه ها این تغییرات را نمی بینند و دیسک را مشابه یک دیسک ساده فرض می کنند.

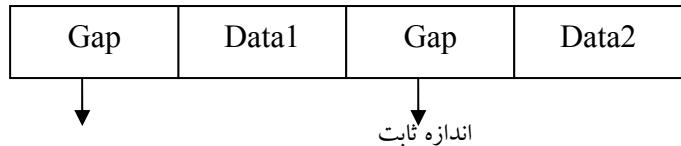


تذکر ۱: اهمیت سیلندر در اینست که تمام داده های روی یک سیلندر می توانند بدون حرکت دادن بازوی هد دستیابی شود لذا هنگام ذخیره یک فایل سعی می کنند حتی الامکان بر روی شیارهای یک سیلندر ذخیره گردد.

تذکر ۲: برای دستیابی به یک از سکتور سکتور خاص، بایت حاوی آن سکتور (معمولاً 512 بایت) به ناحیه ای به نام بافر خوانده شده و سپس بایت مورد نظر در آن بافر بازیابی می شود. نوع جدیدی از هارد دیسکها (هوشمند) خودشان بافر داخلی دارند تا سرعت (R یا W) را بالا ببرند.

تذکر ۳: دو نوع سکتور وجود دارد: ۱- سکتور سخت افزاری ۲- سکتور نرم افزاری

۱- سکتور سخت افزاری (Hard Sector): که توسط کارخانه سازنده ایجاد میشود و ثابت باقی می ماند (فرمت کردن سطح پایین)



۲- سکتور نرم افزاری (Soft Sector): (گاه موسوم به بلاک) که از طریق نرم افزار (سیستم عامل) قابل ایجاد است و به این کار فرمت کردن نرم افزاری می گویند که در ابتدای هر سکتور مشخصات کنترلی نظیر طول هر سکتور ذخیره می شود.

تذکر۴: برای آدرس دهی در دیسکهای مغناطیسی شماره هایی که موجود است:

- شماره د (B) یا A
- شماره استوانه
- شماره شیار در استوانه
- شماره سکتور

تذکر۵: چگالی دیسکهای مغناطیسی برای شیارهای خارجی کمتر از شیارهای داخلی است ولی در دیسکهایی که از تکنیک استفاده میکنند چگالی در تمام شیارها تقریباً یکسان است. (Zone)

تذکر۶: اندازه چگالی دیسکها را تا سال (2000) از قانون تجربی هو گلند (Hoagland) بصورت زیر بدست می آورند:

$$= 10^{\frac{year-1970}{10}} \text{ mb/inch}^2$$

(سال) چگالی

مثال:

با استفاده از قانون هو گلند، چگالی دیسکها در حدود سال (1980) تقریباً چند مگابایت در اینچ مربع بوده است؟

$$= 10^{\frac{(1980-1970)}{10}} = 10^{\frac{10}{10}} = 10^1 = 10 \text{ mb/inch}^2$$

چگالی

پارامتر های دیسک (دیسک های مغناطیسی):

این پارامترها به دو دسته زمانی و پارامترهای ظرفیتی تقسیم میشوند

پارامتر های ظرفیتی:

۱. تعداد رویه (یا هد)
۲. تعداد سیلندر ها (یا تعداد شیار در رویه)
۳. تعداد سکتور در هر شیار
۴. اندازه هر سکتور (اغلب 512 بایت)

* با داشتن این چهار پارامتر می توان ظرفیت اسمی دیسک را بدست آوریم. (قبل از محاسبه شده است).

پارامترهای زمانی:

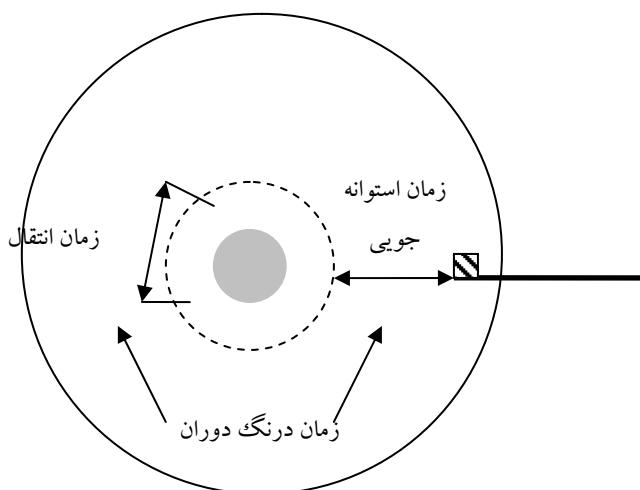
۱. زمان پیگرد یا استوانه جویی (Seek Time):

زمان لازم جهت انتقال هد به سیندلر مورد نظر را گویند این زمان برای دیسکهای با هد ثابت برابر صفر است ، متوسط این زمان را با حرف (S) نمایش می دهیم که این (S) برای محیط های چند کاربره (Multiuser) بیشتر از محیط های تک کاربره (Single User) می باشد.

۲. زمان درنگ دوران یا زمان انتظار دوران (Rotational Latency):

پس از آنکه هد به شیار مورد نظر رسید ، زمانی برای چرخش دیسک لازم است تا سکتور مورد نظر در زیر هد قرار بگیرد که به آن زمان درنگ دورانی می گویند . متوسط درنگ دوران را با ($\frac{1}{2}r$) نشان میدهند که نصف زمان لازم جهت یک دور (2r) چرخیدن دیسک می باشد . سرعت چرخش دیسکها اغلب با واحد تعداد دور در دقیقه یا (RPM) (Minute) سنجیده می شود .

$$\text{زمان یک دور دیسک} \leq \text{زمان درنگ دوران} \leq 0$$



نوار مغناطیسی :

نوار مغناطیسی اصولاً برای پردازش ترتیبی و پی در پی (Sequential) اطلاعات استفاده می شود . از نظر تکنولوژی ساخت نوارها به چهار دسته تقسیم می شوند (منطبق شده با کامپیوتر) :

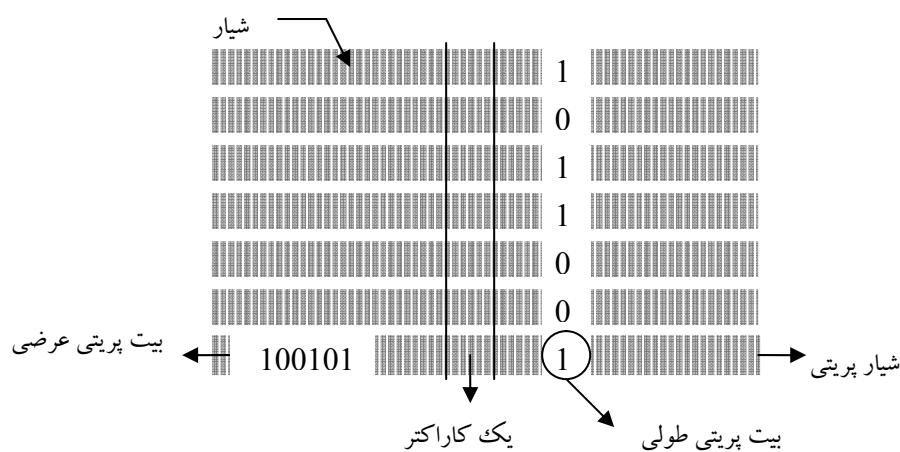
- ۱ - ریل به ریل نوار معمولی
- ۲ - نوار کارتريج ۳
- نوار کاست
- ۴ - نوار صوتی

فرق نوار کارتريج با نوار ریل به ریل :

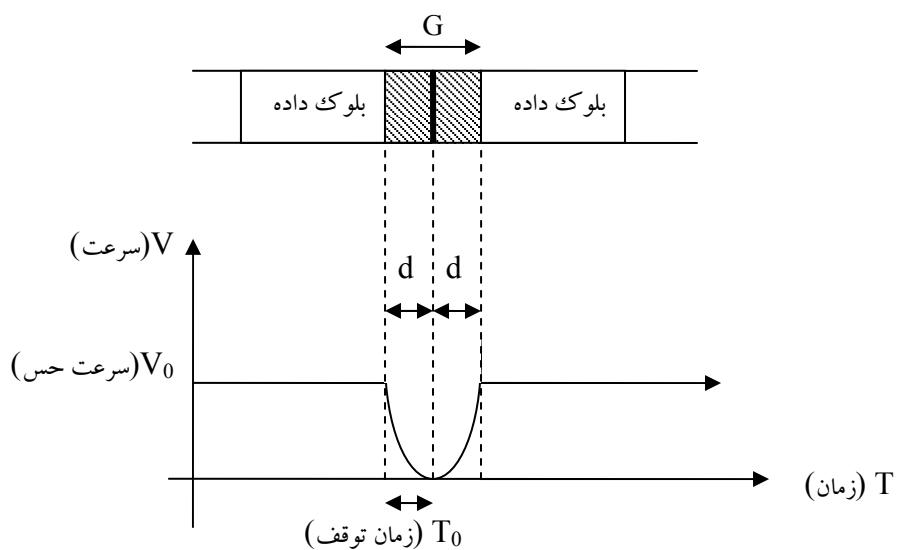
آن است که کارتريج ها در یک محفظه پلاستیکی قرار داده شده اند تا از گرد و خاک محافظت گردد .

نحوه ذخیره داده ها بر روی نوار : به صورت رشته های ترتیبی بر روی شیارهایی (track) است که در سطح نوار وجود دارد. یکی از شیارهای نوار جهت کنترل خطای (بیت پریتی) استفاده می شود.

انواع بیت های پریتی در نوار : ۱- پریتی عرضی ۲- بیت پریتی طولی



برای آنکه هد بخواهد داده های موجود بر روی نوار را بخواند لازم است به سرعت مناسب و یکنواختی بنام سرعت حس برسد از طرف دیگر هنگام توقف، فضایی جهت رسیدن سرعت حس به سرعت صفر مورد نیاز است. در واقع Gap جهت توقف هد یا حرکت دوباره آن استفاده می شود.



$$G = \frac{V_0}{2} = \frac{d}{t_0} \Rightarrow t_0 = \frac{2d}{V_0} \Rightarrow t_0 = \frac{G}{V_0}$$

زمان : t_0 سرعت حس : V_0 طول گپ : G

توقف

واحد سرعت حس : $\frac{\text{Inch}}{\text{Sec}}$

مثال:

اگر $\frac{\text{inch}}{\text{s}} = 450$ باشد زمان t_0 برابر خواهد بود با:

$$t_0 = \frac{G}{V_0} = \frac{0.9}{450} = \frac{9}{4500} = \frac{1}{500} = \frac{2}{500} = \frac{2}{1000} = 2\text{ms}$$

میلی ثانیه

فایل ها به صورت بلاک هایی به شکل پشت سرهم ورینوار ذخیره می شوند . هر بلاک مجموعه ای از رکوردها می باشد بر روی یک نوار می توان چندین فایل را ذخیره کرد که در این حالت هر فایل یک علامت ابتدای فایل (BOF) و یک علامت پایان (EOF) دارد .

پارامترهای نوار :

الف - پارامترهای ظرفیتی :

۱. چگالی (با واحد bpi بیت در اینچ)
۲. طول نوار (غالبا با واحد فوت)
۳. اندازه گپ (G) یا IBG (G) ها که معمولاً بین $0/3$ تا $0/75$ اینچ هستند .

ب - پارامترهای زمانی :

۱. سرعت لغزش نوار (اینچ در ثانیه)
۲. نرخ انتقال (بایت در ثانیه)
۳. زمان حرکت - توقف (بر حسب میلی ثانیه)

تذکر ۱ : نرخ انتقال به دو صورت اسمی (Naminal rate) و واقعی (Effective rate) بیان می شود .

نرخ اسمی توسط شرکت سازنده بیان می شود و نرخ واقعی قابل محاسبه است .

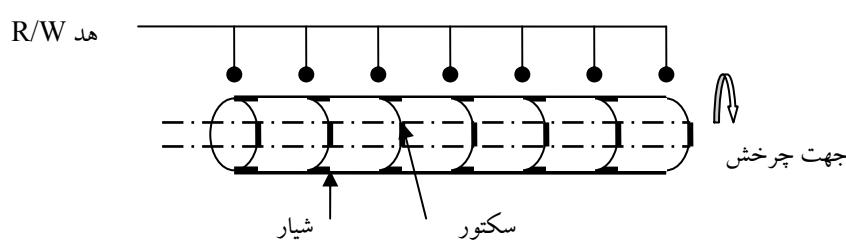
تذکر ۲ : دستیابی ترتیبی در نوارها سریع بوده و حمل و نقل و نگهداری آن نسبت به دیسک ها ساده بوده و ارزانتر بوده و شرایط محیطی مختلفی را به خوبی تحمل می کند . امروزه نوارها به همراه CD ها برای آرشیو و بایگانی استفاده می شود .

دیسک نرم (فلایپی) :

از رسانه های خارجی با دستیابی مستقیم بود که از نظر شکل ظاهری مشابه دیسک های متداول البته با ابعادی کوچکتر است که جنس آن از نوعی پلاستیک است و سختی دیسک های سخت را ندارد . این دیسک ها در محفظه ای جهت حفاظت آن از تماس های خارجی و هم به منظور ثابت نگه داشتن آن قرار گرفته اند . دیسک ها در اندازه های $3/5$ ، $5/25$ و 8 اینچی ساخته شده اند که نوع $3/5$ در دنیای رواج دارند . از فلایپی دیسک ها مانند نوار برای تهیه پشتیبان و یا نقل و انتقال داده ها استفاده می شود برخلاف دیسک های سخت که در آن نوک خواندن / نوشتن با دیسک تماس دارد . این دیسک ها می توانند یک رویه یا دو رویه باشند و داده ها به صورت سریال روی شیارها ذخیره می شوند . این دیسک ها مجهز به مکانیزمی ساده در مقابل عمل نوشتن هستند .

طبله (Drum) :

طبله استوانهای است که در سطح خارجی آشیاهایی وجود دارد و اغلب برای یک شیار یک R/W تعییه شده طبله می تواند با هد ثابت (تعداد شیار با نوک R/W برابر باشد) و یا هد متحرک باشد (تعداد شیار با تعداد نوک هد R/W متفاوت باشد) طبله از دیسک ها و نوار سریعتر است ولی ظرفیت آن کمتر است و امروزه به ندرت از طبله استفاده می شود .



: دیسک های نوری (optical disk)

در این دیسک ها به جای مغناطیسی از نور لیزری جهت ذخیره اطلاعات استفاده می شود . در این تکنولوژی فضای لازم جهت ذخیره سازی یک بیت خیلی کمتر بوده و بدین دلیل استفاده از دیسک های نوری باعث کاهش فضای ذخیره سازی می گردد این دیسک ها انواع متفاوتی دارند

۱ - : (compact disk read only memory)

دارای ظرفیت ۵۰۰ مگا تا چند گیگابایت است . این دیسک فقط خواندنی بوده و از این جهت به آن WORM (write once memory) نیز می گویند اغلب از این نوع دیسک ها جهت ذخیره سازی فایل هایی که تغییر نمی کنند (مثل برنامه ها) استفاده می شود . مزیت CD-ROM ها ، ظرفیت بالا ، بهای کم و دوام آنهاست و نقطه ضعف آنها زمان استوانه جویی و نرخ انتقال پایین است . به دلیل سرعت کم CD ها ، طراحی ساختار فایل در آنها نسبت به دیسک مشکل تر بوده و امروزه اغلب به عنوان حافظه نوع سوم (یا حافظه خارج از خط) جهت آرشیو پشتیبان گیری و تکثیر و ارائه برنامه ها استفاده می شود . به دلیل اینکه ابتدا CD ها جهت اجرای موسیقی طراحی شده بودند و دارای ظرفیت بالا و سرعت متوسطی هستند .

۲ - نوع دیگری از دیسک های نوری DVD (digital video disk) هستند :

دارای ظرفیت بیش از ۱۰ گیگابایت هستند این دیسک ها برای ذخیره سازی فایل ها ابداع شده است که برای ذخیره سازی فایل ها هم انجام می شود .

CD های اولیه فقط یکطرفه بودند ولی بعضی DVD ها به صورت دو طرفه نیز ساخته می شوند .

تذکر ۱ : در دیسک های MO (magneto optic) از دو تکنولوژی نور و مغناطیسی استفاده می گردد و قابلیت R/W را دارند .

تذکر ۲ : اطلاعات بر روی بعضی CD ها به نام CD_EOD (CD_Read write) یا CD_RW (erasable optical disk) قابل پاک شدن و نوشتمن مجدد می باشد .

نحوه ذخیره کردن داده ها در CD ها :

همچنانچه قبل از دیسک ها (هارد دیسک) با سرعتی ثابت می چرخند و سرعت زاویه های ثابتی را دارند ، شیارها مجزا و متعددالمرکز بوده و چگالی داده ها در شیارهای خارجی کمتر از شیارهای داخلی است و دیسک برای همه موقعیت های هد با سرعت یکسانی می چرخد و آدرس دهی به کمک سه عدد (شماره هد ، شماره سیلندر ، شماره سکتور) انجام می پذیرد ، سرعت خطی در شیارهای بیرونی بیشتر از شیارهای داخلی است . ولی ساختار ذخیره داده ها در CD-ROM ها به شکل حلزونی می باشد .

و داده ها بر روی سکتورهایی که اندازه آنها با هم برابرند ذخیره می شود در صورتی که در دیسک های سخت اندازه سکتورهای مختلف یک دیسک باهم تفاوت داشته و معمولا در آنها فایل ها با شروع از یک استوانه ، استوانه به استوانه در فضایی پیوسته روی دیسک ذخیره می شوند ممکن است فایل در فضایی ناپیوسته هم ذخیره شود و یا پس از مدتی ناپیوستگی و بندبند شدگی

(fragmentation) در محیط ذخیره سازی فایل ها (درون یک فایل و یا بین فایل ها) پدید آید در این صورت از سیستم می توان درخواست کرد تا فضای اشغال شده را دوباره پیوسته و یکپارچه کند (Defragmentation) در صورتیکه در CD_ROM برای خواندن داده ها باید شیار با سرعت ثابتی از زیر پیکاپ (pick up) نوری رد شود به عبارت دیگر سرعت خطی باید ثابت باشد برای ثابت ماندن سرعت خطی هم ، سرعت چرخش دیسک هنگام خواندن لبه های بیرونی می بايستی کندر از هنگام خواندن لبه های داخلی باشد . یعنی سرعت زاویه ای ثابت نیست . در این ساختار مارپیچی ، چگالی داده ها در کل شیار ثابت است .

نحوه آدرس دهی در CD_ROM ها :

با سه عدد (سکتور : ثانیه : دقیقه) است مثلا ۱۸:۴۳:۲۷ یعنی سکتور ۲۷ ، در ثانیه ۴۳ از دقیقه ۱۸ چرا که برای CD_ROM ها جهت جمع آدرس دهی از تکنیک اجرای موسیقی در هر ثانیه به هفتاد و پنج سکتور تقسیم بندی و هر سکتور هم ۲ کیلو بایت اطلاعات را ذخیره می کند و طبق قرارداد اولیه هر CD حداقل باید بتواند یک ساعت موسیقی را ذخیره سازد لذا هر CD حداقل باید بتواند ۵۴۰۰۰ کیلوبایت داده را در خود نگهداری کند

$$1 \text{ ساعت} = 3600 \text{ ثانیه} \longrightarrow 3600 * 75 * 2 = 540000 KB = 540 MB$$

تکنیک های ضبط مغناطیسی اطلاعات :

سطح نوار یا دیسک مغناطیسی مثبت ، منفی یا منفی داشته باشد . از نظر نحوه مغناطیسی شدن تکنیک های ضبط اطلاعات متعددی ابداع شده از جمله :

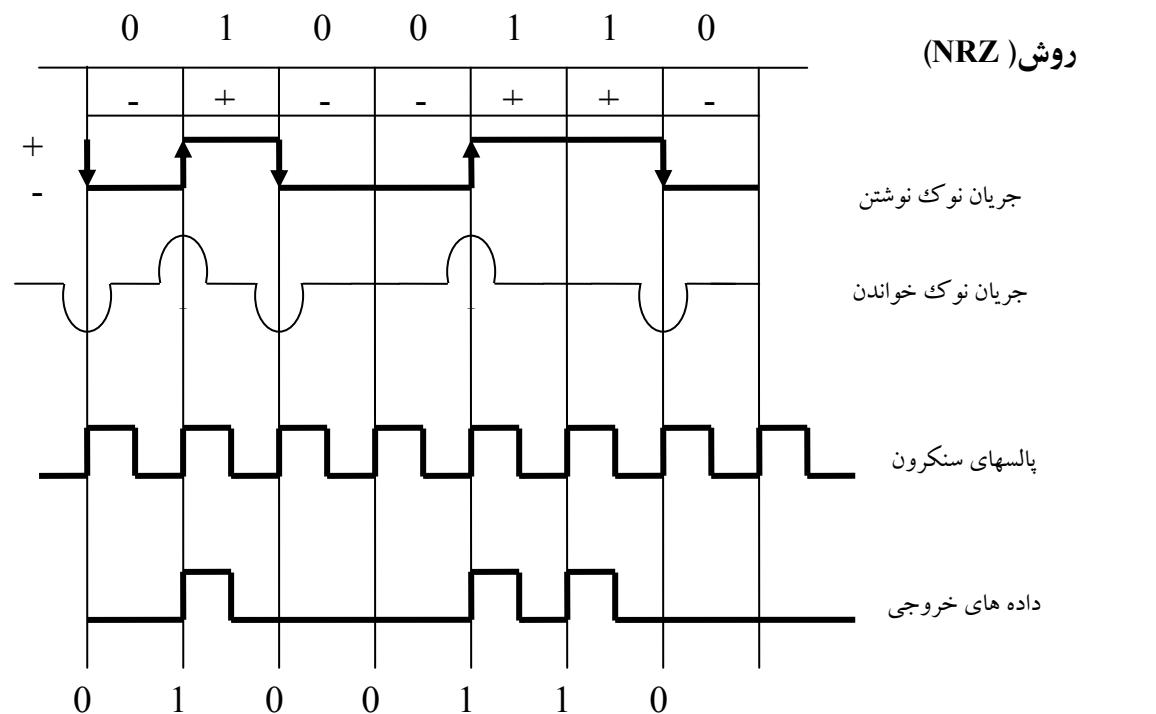
۱. بازگشت به صفر (Return to zero = RZ)
۲. بی بازگشت به صفر (Non return to zero = NRZ)
۳. بی بازگشت به صفر معکوس (Non return to zero Inverted = NRZI)
۴. کد کردن (Run length limit = RLL)
۵. PE
۶. FM
۷. MFM

۱ - تکنیک بازگشت به صفر (RZ) :

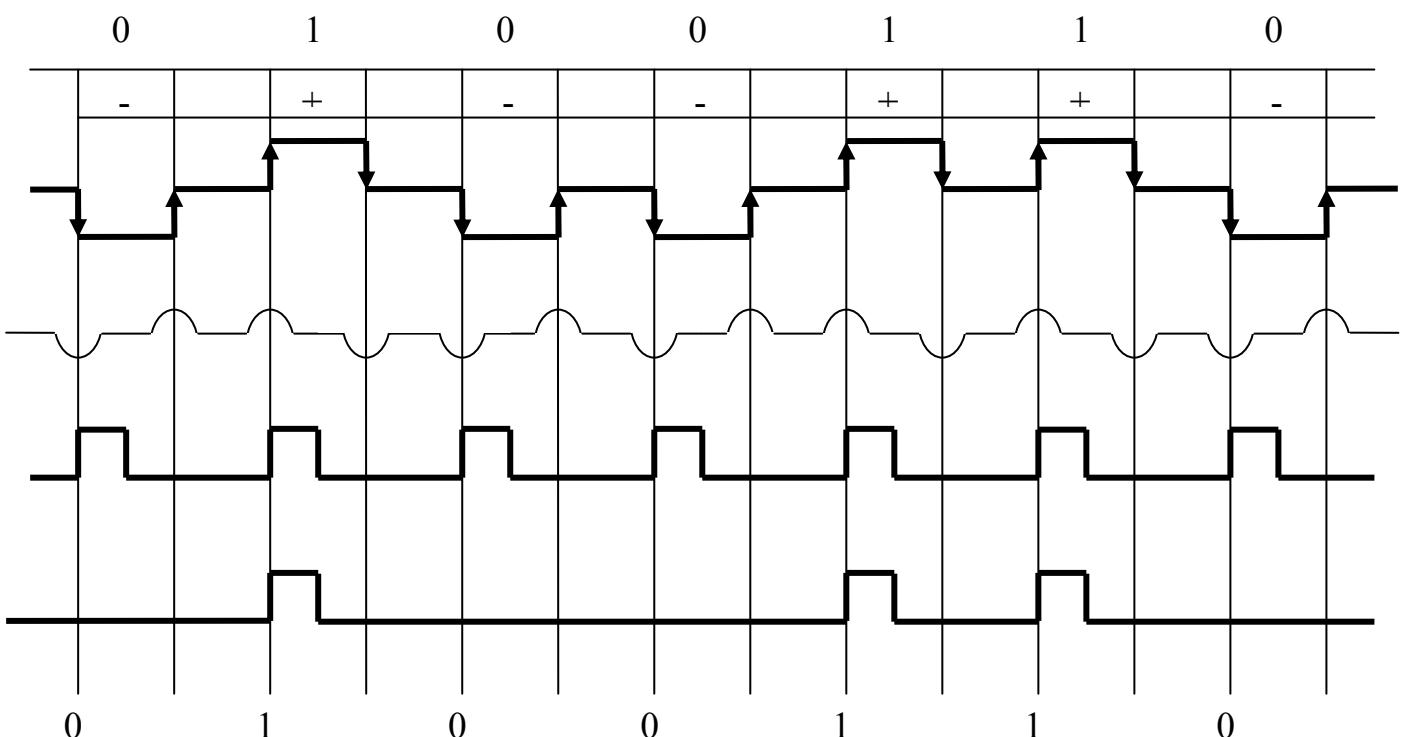
عبور جریان در یک سمت سبب ایجاد نقطه مغناطیسی روی سطح می شود . عبور جریان در سمت دیگر نقطه منفی و اگر جریانی وجود نداشته باشد نقطه حالت خنثی پدید می آید . در این روش در نقاطی که در آنها جریان نوشتن صفر است عمل ضبط صورت نمی گیرد با این دلیل به این روش «بازگشت به صفر» می گویند . هنگام خواندن وجود تغییر در حالت مغناطیسی نقاط باعث پدید آمدن جریان در هد (نوک) می شود مثلا اگر سطح نوار تماما به صورت مثبت مغناطیس شده باشد جریانی از هد خواندن عبور نمی کند . هنگام خواندن اگر جهت جریان نوک خواندن مثبت باشد ب "۱" و اگر منفی باشد به صفر تغییر می شود .

۲- تکنیک بازگشت به صفر (NRZ)

در این تکنیک جریان هد نوشن هیچگاه صفر نیست به عبارتی دیگر بر روی نوار نواحی خشی هیچگاه وجود ندارد. هنگام خواندن جریان مثبت به ۱ و جریان منفی به صفر تعبیر می شود. باید توجه شود که در این روش یک های پی در پی و صفرهای پی در پی تغییری در شار ایجاد نمی کنند در هنگام نوشن برای بیت "۱" جریان مثبت و برای بیت ". ". جریان منفی است و جریان صفر نداریم در هنگام تعیین داده خروجی اگر جریان منفی باشد به صفر و اگر جریان مثبت باشد به "۱" تعبیر می شود و اگر جریان وجود نداشته باشد داده قبلی تکرار می شود. تکانه ۴ (پالسها) ی سنکرون شروع بیت هار نشان می دهد



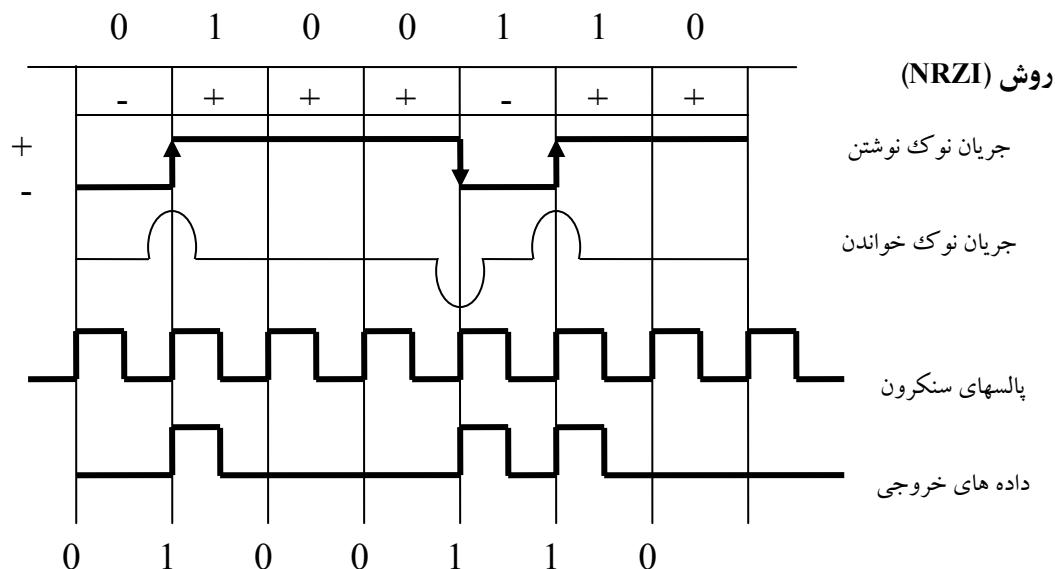
روش (RZ)



شکل جریان در هد خواندن از روی شکل جریان و هد نوشتن بدست می آید و هر گاه جریان هد نوشتن بالارونده بود جریان هد خواندن مثبت است و هر گاه جریان هد نوشتن پایین رونده بود جریان هد خواندن منفی است.

۳- تکنیک بی بازگشت به صفر معکوس (NRZI):

در این تکنیک هر تغییر شار (بدون توجه به جهت آن) نمایانگر "۱" و عدم تغییر شار نمایانگر "۰" می باشد. در هنگام نوشتن ناهمگام رسیدن به "۱" جهت جریان در هد نوشتن را عوض می کنیم و در هنگام خواندن اگر جریان در هد خواندن باشد (هر جهتی) این جریان به "۱" تعبیر می شود و اگر جریان وجود نداشته باشد به صفر تعبیر می شود.



*توجه:

چگالی ذخیره سازی اطلاعات یعنی تعداد بیت هایی که در واحد طول می تواند ذخیره شود بستگی به تعداد تغییر شار در واحد طول دارد. هر چقدر تعداد تغییر شارها در واحد طول کمتر باشد امکان ذخیره سازی بیت های بیشتری در سطح وجود دارد همانطور که در سه شکل قبلی مشاهده می کنیم برای ذخیره و بازیابی رشته (۰۱۰۱۱۰) در روش RZ تعداد ۱۴ تغییر شار، در روش NRZ تعداد ۵ تغییر شار و در روش NRZI تعداد سه تغییر شار داریم پس چگالی NRZI از دو روش دیگر بیشتر است.

$$8 \text{ bit} = 1 \text{ byte}$$

$$1024 \text{ byte} = 1 \text{ K byte}$$

$$1024 \text{ Kbyte} = 1 \text{ M byte}$$

$$1024 \text{ M byte} = 1 \text{ G byte}$$

«فصل دوم»

موجودیت (entity type)

به فرد ، شیء ، پدیده یا مفهومی که می خواهیم در رابطه با آن اطلاع داشته باشیم موجودیت گفته می شود .

محیط عملیاتی :

به محیطی که در رابطه با آن می خواهیم یک سری داده ها را ذخیره ، بازیابی و پردازش کنیم محیط عملیاتی یا «فرد جهان واقع» گفته می شود .

مثال

محیط عملیاتی دانشگاه از موجودیت های دانشجو ، استاد ، درس ، کارمند ، کلاس و غیره تشکیل یافته است . تذکر : انواع موجودیت ها توسط صفات خاصه (Attributes) مربوط به هر یک ، از موجودیت ها متمایز می گردند . هر موجودیت صفات خاصه مربوط به خود را دارد مثلا استاد دانشگاه می تواند صفات خاصه نام ، مدرک ، سن ، آدرس ، سابقه تدریس و غیره را داشته باشد .

فیلد :

مکان ذخیره شدن یک واحد معنادار یا یک فقره اطلاع را فیلد گویند . فیلد کوچکترین واحد اطلاعات در فایل است که محتوای آن از نظر کاربرد دارای معنای منطقی می باشد .

طول این واحد مکان حداقل یک کاراکتر (گاه یک بیت) است و حداکثر بستگی به سیستم فایل و به برنامه سازی دارد . محتوای این فیلد ممکن است واحد معنایی داده (فیلد ساده) باشد که در این صورت حالت تجزیه ناپذیر (Atomicity) دارد و یا این مکان از چند مکان کوچکتر تشکیل شده باشد که می گوئیم فیلد مرکب است لذا داده ذخیره شده در آن از نظر معنایی تجزیه پذیر است .

اطلاع :

هر صفت خاصه از دو مولفه «اطلاع» گفته می شود که اطلاع توسط انسان یا ماشین ، تولید ، ذخیره ، بازیابی و پردازش می شود .

مثال :

«نام خانوادگی» صفت خاصه و «عظمی» مقدار صفت خاصه است حال عبارت زیر یک اطلاع است :

عظمی = نام خانوادگی

تعریف رکورد :

مجموعه ای از فیلد های مربوط به یک موجودیت تشکیل رکورد را می دهند . مثلا در موجودیت استاد مجموعه فیلد های «نام استاد - رشته - مدرک - سابقه تدریس - دانشکده» را رکورد گویند .

تعریف فایل :

مجموعه ای از رکوردها نیز فایل را تشکیل می دهند .

-بررسی

۱ - ساختارهای فیلد

۲ - ساختارهای رکورد

ساختارهای فیلد :

برای مشخص کردن فیلد رکوردها را محل های مختلف زیر وجود دارند :

۱ - قرار دادن فیلدها در طول های از قبل تعیین شده و ثابت مثل :

نام	فامیلی	مدرک	رشته
علی	احمدی	دکترا	فیزیک
کریم	حسینی	فوق لیسانس	ادبیات
فراز	فاطمی	کامپیوتر	فوق لیسانس

ایراد این روش :

الف - برای رساندن فیلدها به طول معین باید از فاصله خالی استفاده شود و این فضاهای خالی باعث بزرگ شدن فایل و اتلاف حافظه دیسک می گردد .

ب - ممکن است داده هایی بزرگتر از حد آن فیلد در آینده باشند . مثل نام ، آدرس که تنوع طولی دارند لذا برای اینها این روش خوب نیست ولی برای برخی از داده ها (نظیر معدل) مناسب می باشد .

۲ - قرار دادن طول فیلد در ابتدای هر فیلد ، مثل :

03ali06javadi06doctor07physics

۳ - استفاده از کاراکتر ویژه به عنوان فاصل (delimiter) در انتهای هر فیلد مثل:

ali,vadi,octor07physics

۴ - بکار بردن نام هر فیلد در مقابل هر فیلد مثل :

name = Ali , family = Javadi , city = Tehran

ایراد این روش :

اتلاف حافظه در اثر ذخیره نام فیلدها است

مزیت این روش :

هر فیلد خود توصیف بوده و فیلدها می توانند جا به جا شوند . همچنین مقادیر بعضی از فیلدها در صورت عدم وجود ذخیره نمی گردد .

ساختارهای رکورد:

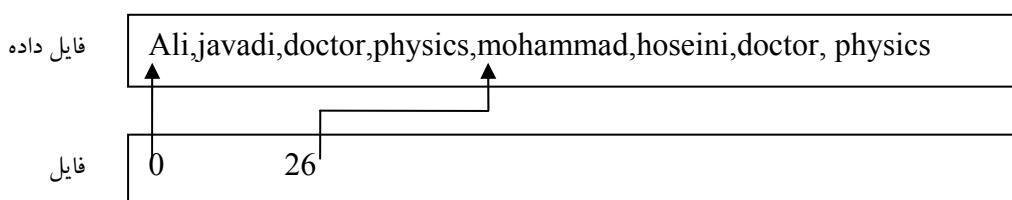
بعضی از روش‌های سازماندهی به رکوردها عبارتند از:

۱- رکوردهای با طول ثابت: در این روش تمام فایل با هم برابر می‌باشد و این متداولترین روش سازماندهی رکوردهاست. ثابت بودن طول رکورد الزاماً به معنای ثابت بودن طول رکوردهای تشکیل دهنده آن نیست.

۲- تعیین طول رکوردها بر حسب تعداد فیلهای آن: در این روش هر رکورد از n فیلد تشکیل یافته است و n برای کل فایل ثابت است. مثلاً اگر $n = 4$ باشد آنگاه فایل می‌تواند به صورت ali,Ahmadi,physics,doctor,mohammad,hoseini,computer,doctor,....

۳- ذخیره طول رکورد در اول هر رکورد: در این روش در ابتدای رکورد طول آن ذخیره می‌شود. این روش اغلب برای کار با طول متغیر به کار می‌رود.

۴- استفاده از اندیس برای نگهداری آدرس‌های آفست هر رکورد نسبه به اول هر فایل: این اندیس‌ها می‌توانند در فایل جداگانه‌ای ذخیره شوند مثل:



۵- ذخیره یک علامت ویژه فاصل در انتهای هر رکورد

مفهوم رکورد:

از همه دیدگاه (سطح) بررسی می‌شود:

۱- رکورد در سطح انتزاعی (presentation)

۲- رکورد در سطح برنامه کاربر (سطح منطقی)

۳- رکورد در محیط ذخیره سازی

رکورد در سطح انتزاعی:

مفهوم رکورد مستقل از جنبه‌های نمایش آن در محیط منطقی برنامه کاربر و همچنین پیاده سازی آن در محیط ذخیره سازی مطرح می‌کنیم و مجموعه اطلاعاتی است که در مرود هر یک از نمونه‌های متمایز یک یا بیش از یک نوع موجودیت (Entity) از یک خرد جهان واقع می‌خواهیم در اختیار داشته باشیم.

رکورد در سطح منطقی:

مفهوم رکورد از دید کاربر برنامه ساز مطرح می‌شود (جنبه نمایشی آن)

رکورد در محیط ذخیره سازی: رکورد آن گونه که در محیط فیزیکی پیاده سازی می‌شود بررسی می‌گردد
خرد جهان واقع (Micro read world):

بخشی از جهان بیرونی و محسوس که فعالیت‌های نگهداری داده و پردازش داده در آن انجام می‌گیرد تا مجموعه‌ای از نیازهای اطلاعاتی رفع شود. مانند حسابداری یک سازمان، بایگانی یک اداره و ...

موجودیت (Entity)

پدیده ، فرد ، شیء یا مفهوم که که در مورد آن می خواهیم «اطلاع» داشته باشیم که هر محیط عملیاتی موجودیت های خاص خود را خواهد داشت مثلا: محیط عملیاتی دانشگاه ، موجودیت های دانشجو ، استاد ، درس و ... را دارد .

صفت خاصه (Attributes): هر موجودیتی جهت تمایز از موجودیت دیگر صفات خاص خود را دارد که از یکدیگر تمایز می

شوند

مثال :

نوع موجودیت : دانشجو

صفات خاصه :

نام و نام خانوادگی ، رشته ، سال ورود ، سال تولد ، قد ، وزن و ...

اطلاع :

هر صفت خاصه دارای دو مولفه است : اسم صفت خاصه و مقدار صفت خاصه . وقتی که این جفت مولفه موجود باشد می گوییم «اطلاع» حاصل است

مقدار صفت خاصه	اسم صفت خاصه
علی احمدی	= نام و نام خانوادگی
کامپیوتر	= رشته

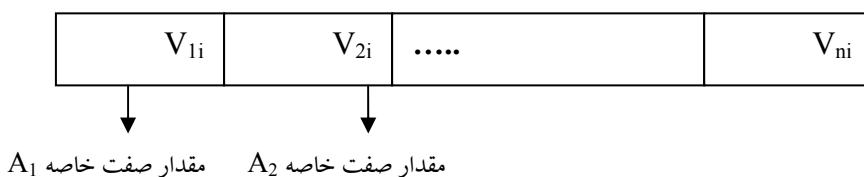
دو طرح برای ساختار رکوردها :

• طرح با قالب ثابت مکان (Fixed positional)

• طرح با قالب غیر ثابت مکان (Non Fixed _ Non positional)

طرح با قالب ثابت مکان :

در این طرح ، در هر فیلد ، فقط مقدار صفت خاصه ذخیره می شود و برای اینکه مشخص شود محتوای هر فیل مربوط به کدام صفت خاصه است در تعریف ساختار رکورد با کمک مکان فیلد ، در قالب از پیش ثابت مشخص می شود .



طرح با قالب غیر ثابت مکان :

در این طرح ، در هر فیلد از نمونه رکورد ، هم اسم صفت خاصه و هم مقدار آن ذخیره می شود و مکان یک فقره اطلاع در نمونه های مختلف یک نوع رکورد از قبل مشخص و ثابت نیست . در این طرح تعداد فیلدها ، طول آنها و مکان آنها در نمونه های مختلف یک نوع رکورد یکسان نیست .

در این طرح معمولاً طول رکوردهای نمونه متغیر است

رکورد

R1 $A_1 = V_{1_1}, A_2 = V_{2_2}, A_3 = V_{3_3}, \dots$

R2 $A_3 = V_{3_2}, A_1 = V_{1_2}, A_2 = V_{2_2}$

پس می توان گفت رکورد در سطح منطقی از نظر طول دو نوع است :

- رکورد با طول ثابت (Fixed length read)
- رکورد با طول متغیر (Variable length read)

دلایل متغیر بودن طول یک رکورد :

به سه دلیل این طول می تواند متغیر باشد :

۱. متفاوت بودن تعداد صفات خاصه مورد نیاز برای نمونه ای مختلف یک موجودیت
۲. متغیر بودن طول فیلدها .

مثال :

در مورد دانشجو - درس ، صفت خاصه شماره درس متغیر است مثلا یک دانشجو ۳ درس و دانشجوی دیگر ۲ درس را ثبت نام کرده است

۳. وجود صفت خاصه چند مقداری

شماره کارمند	نام	نام فرد تحت تکلف	تاریخ تولد فرد تحت تکلف
--------------	-----	------------------	-------------------------

شماره صفت مرکب خاصه چند مقداری

رکورد ذخیره شده در محیط ذخیره سازی :

در این سطح ، رکورد علاوه بر داده هایی که دارد ممکن است ، بخش دیگری به نام غیر داده ای هم داشته باشد نام دیگر غیر داده ای عبارت است از : (prefix) ، کنترل سیستمی ، متابخش (meta section) (یا متاپار)

بخش غیر داده ای	بخش داده ای
-----------------	-------------

اطلاعاتی که در بخش غیر داده ای رکورد ذخیره می شوند :

۱. رکورد

۲. نوع رکورد

۳. اشاره گرها (نشان نما)

۴. فلاگ های عملیاتی بمشل (operating Flag)

۵. فلاگ های حفاظتی (Protection Flag)

۶. اطلاع هایی خاص در بعضی از ساختار ها

• فیلد حاوی طول رکورد :

وقتی که رکوردها طول متغیر دارند از این فیلد برای درج طول رکورد استفاده می شود که اگر رکورد با طول ثابت باشد نیازی به این فیلد نیست .

• فیلد حاوی نوع رکورد :

فایل حاوی مجموعه ای از نمونه های مختلف از یک نوع رکورد (فایل تک نوعی Mono type) یا چندین نوع رکورد (فایل چند نوعی Multi type) است که دارای نام و ساختار مشخص (وجود ساختار برای فایل الزامی نیست) است . فیلد حاوی نوع رکورد در فایل های چند نوعی برای مشخص نمودن نوع رکورد الزامی است .

• فیلدهای حاوی نشانه روهای اشاره گرها :

برای پیاده سازی ساختار منطقی فایل (ارتباطات ساختاری بین رکوردها) و ایجاد ساختار فیزیکی از اشاره گرها استفاده می شود . حجم اشاره گرها در ساختارهای مختلف متفاوت است در واقع اشاره گر آدرسی است که در فیلدی جای داده می شود تا مکان «داده ای» را زا نقطه ای از فایل به نقطه ای دیگر نشان دهد . در واقع هر اشاره گر دارای یک مبدأ و یک مقصد است . از اشاره گرها برای موارد زیر می توان استفاده کرد :

۱. رکورد به رکورد

۲. رکورد به بلاک

۳. بلاک به بلاک

۴. بلاک به رکورد

۵. گروهی از بلاک ها به گروهی دیگر

۶. فایل به فایل

در ک بهتر نقش اشاره گرها : ساختار منطقی فایل ، ساختاری است که ارتباط بین رکوردها را از دید برنامه ، فایل پرداز نشان می دهد و بر اساس آن رکوردهای فایل تحت پردازش قرار می گیرند در صورتی که از نظر محیط فیزیکی ذخیره سازی لزوما همچوار نیستند .

• فیلد(های) حاوی فلاگ(های) عملیات :

فلاگ تعدادی بیت یا بایت است حاوی اطلاعات وضعیتی و راهنمایی و به منظورهای مختلفی استفاده می شود . در سیستم فایل ، فلاگ عملیاتی به دو منظور به کار می رود :

۱. برای نشان دادن عملی که قرار است بر روی رکورد انجام شود . *

۲. برای نشان دادن عملی که بر روی رکورد انجام شده است . **

* مانند فلاگ حذف با تاخیر که گاه به حذف منطقی هم موسوم است بدین ترتیب که در بعضی از ساختارهای فایل وقتی که کاربر درخواست حذف رکورد می کند سیستم این عمل را با تاخیر انجام می دهد دلیل آن این است که برای این منظور در بخش پیش بندی رکورد، «فلاگ حذف» درج می کند تا بعداً رکورد را به طور فیزیکی حذف کند.

** مانند فلاگ به هنگام سازی. در بعضی از ساختارها وقتی که رکورد به هنگام در می آید در فیلدی از بخش پیشوندی فلاگی، مشخص می شود که نسخه جدید توسط کدام کاربر و در چه زمانی و تاریخی تولید شده است.

این نوع فلاگ گاه به صورت یک رکورد شامل چندین فیلد در فایلی جداگانه ای (فایل log در سیستم مدیریت پایگاه داده ها) ذخیره می شوند.

• فیلد حاوی فلاگ حفاظتی (قفل حفاظت):

در این فیلد فلاگی درج می شود تا رکورد را از دستیابیهای غیر مجاز مصون بدارد. به این فلاگ گاه قفل در سطح رکورد (Record Level Lock) نیز می گویند به گونه ای که تنها کاربر مجاز می تواند به آن رکورد دستیابی داشته باشد (این فلاگ در فایل های اشتراکی، کاربرد دارد).

حق دستیابی کاربر به فایل یا بخشی از آن:

- حق دستیابی برای خواندن
 - حق دستیابی برای نوشتمن
۱. حق نوشتمن برای درج رکورد جدید
 ۲. حق نوشتمن برای حذف رکورد
 ۳. حق نوشتمن برای بهنگام سازی رکورد

: (Data Protection) داده ها

۱. سطح فیلد
۲. سطح رکورد
۳. سطح گروهی از بلاک ها
۴. سطح فایل
۵. سطح گروهی از فایل ها

خصوصیات کلید رکورد (کلید اصلی، کلید ثانویه، کلید خارجی):

کلید، صفت خاصه ساده یا مرکبی است که دو خاصیت زیر را دارا باشد:

در نمونه های مختلف، رکورد در دوره حیات فایل، مقدار یکتاپی را داشته باشد
طول آن حدالامکان کوتاه باشد.

کلید رکورد در واقع شناسه (صفت خاصه) یک نوع موجودیت است و به کمک هر مقدار یک نمونه از موجودیت از هر نمونه دیگر متمایز می گردد. طول کلید می تواند ثابت (مدیریت ساده) و یا متغیر (مدیریت پیچیده) باشد

کلید اصلی و کلید ثانویه :

یک نوع موجودیت ممکن است بیش از یک شناسه و در نتیجه رکورد نمایشگر آن بیش از یک کلید داشته باشد که در یکی از کلیدها که کوتاهتر و مهمتر از بقیه باشد به عنوان کلید اصلی و کلیدهای دیگر به عنوان کلید ثانویه در نظر گرفته می شوند.

کلید خارجی :

گاهی یافتن صفت خاصه ای که دو شرط فوق را داشته باشد دشوار است لذا یک صفت خاصه ای را که جزء صفات خاصه طبیعی موجودیت نیست به مجموعه صفات خاصه طبیعی آن اضافه می کنند که به چنین صفت خاصه ای کلید خارجی می گویند. مانند شماره کارمندی ، شماره قطعه ، شماره دانشجویی

معمولًا مقادیر این کلید با «کد گذاری» خاصی انتخاب می شود و به طور سنتی اجزاء تشکیل دهنده این کلید را معنادار در نظر می گیرند.

dd	d	dd	d	d	ddd
↓	↓	↓	↓	↓	↓
سال	کد جنسیت	کد دانشکده	کد رشته	کد گرایش	کد شناسه دانشجو

فایل :

مجموعه ای است نامدار با ساختار دورنی مشخص از نمونه های مختلف یک نوع رکورد (فایل تک نوعی) و گاهی بیش از یک نوع رکورد (فایل چند نوعی). در این صورت می گوییم که فایل یک ساختار داده ای خارجی است یعنی در حافظه خارجی (برون ماشینی) ذخیره شده است.

گاهی ممکن است که دنباله ای از رکوردها نباشد ، بلکه دنباله ای بی ساختار از نویسه ها (کاراکترها) باشد که از نظر سیستم فایل معنای خاصی هم ندارند.

انواع فایل :

۱. با ساختار : یعنی دنباله ای از رکوردهاست.
۲. بی ساختار : یعنی دنباله ای از نویسه ها (کاراکترها) است.

منظور از ساختار فایل :

۱. ساختار منطقی : نشان دهنده سازمانی است که بر اساس آن رکوردهای منطقی گرد هم آمده اند . به عبارت دیگر ساختار منطقی از دید کاربر ، چگونگی ارتباطات و پیوندهای بین رکوردهای منطقی را نشان می دهد.

۲. ساختار فیزیکی : نشان دهنده چگونگی ذخیره سازی بلاک های فایل در رسانه (مثلاً دیسک) است در واقع ساختار فیزیکی ، نشان دهنده دید برنامه ساز سیستم نسبت به فایل است.

ساختار فیزیکی :

نشان دهنده چگونگی ذخیره سازی بلاک های فایل در رسانه (مثالاً دیسک) است در واقع ساختار فیزیکی ، نشان دهنده دید برنامه ساز سیستم نسبت به فایل است .

سیستم فایل :

در سیستم های جدید چندین لایه سخت افزاری و نرم افزاری موجود است که سیستم بتواند با کارایی و معمولاً نابنامه ساز است ، پاسخ دهد . End User انعطاف پذیری بیشتری به درخواست کاربر انتهايی هر لایه که وظایفش از لایه پایین تر استفاده می کند و به نوعه خود به لایه بالاتر خدماتی را ارائه می دهد.

رویه های کتابخانه ای :

این رویه ها برای موارد ذیل استفاده می گردند :

۱. برای اجرای درخواست کاربر پایانی ، خود رویه های را در سیستم فایل فرا می خواند مثل رویه های OPEN, CLOSE, READ, WRITE,...

فرمان کاربر انتهايی
برنامه های کاربردی
رویه های کتابخانه ای
سیستم فایل
سیستم عامل
مجموعه دستورات ماشین
سخت افزار

سیستم فایل :

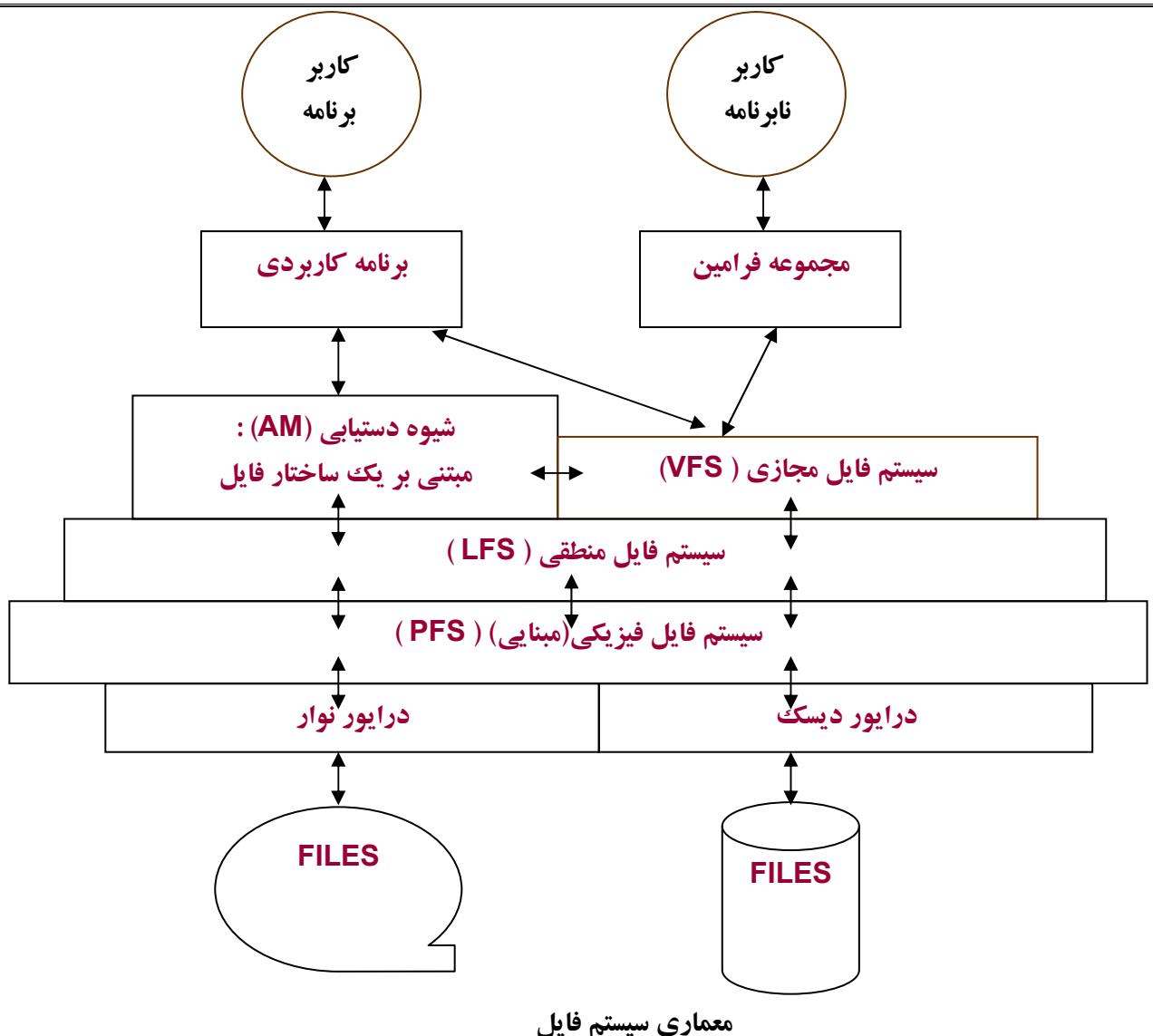
سیستم فایل درخواست های لایه بالاتر را برای عملیات ورودی / خروجی فیزیکی به صورت فراخوانی های تابعی در سیستم عامل تبدیل می کند .

سیستم عامل :

برای اینکه عملیات فیزیکی در محیط دیسک انجام شود سیستم عامل یک «برنامه کانال» را به اجرا درمی آورد.

- معماری سیستم فایل و لایه های آن :

- اجزای اصلی این معماری عبارتند از :



معماری سیستم فایل

درایور :

درایور در پایین ترین سطح معماری قرار دارد و مستقیماً با کنترلر (کانال) در تماس است.

سیستم فایل فیزیکی (MBeany) :

این لایه مسئول ذخیره سازی بلاک ها روی رسانه خارجی و انتقال آنها از رسانه به بافر و برعکس است. به طوری که با محتوای بلاک ها و یا ساختار فایل کاری ندارد و در بعضی از سیستم ها، این لایه بخشی از خود سیستم عامل است.

سیستم فایل منطقی :

این لایه امکان درخواست های کاربر را انجام می دهد بطوری که بدون در نظر گرفتن مفهوم بلاک در محیط فیزیکی به کاربر این امکان را می دهد تا به رکوردها دستیابی داشته باشد در واقع به صورت واسطه بین سیستم فایل فیزیکی و لایه شیوه دستیابی عمل کرده و لایه محیط فیزیکی ذخیره سازی را به صورت یک محیط منطقی می بیند.

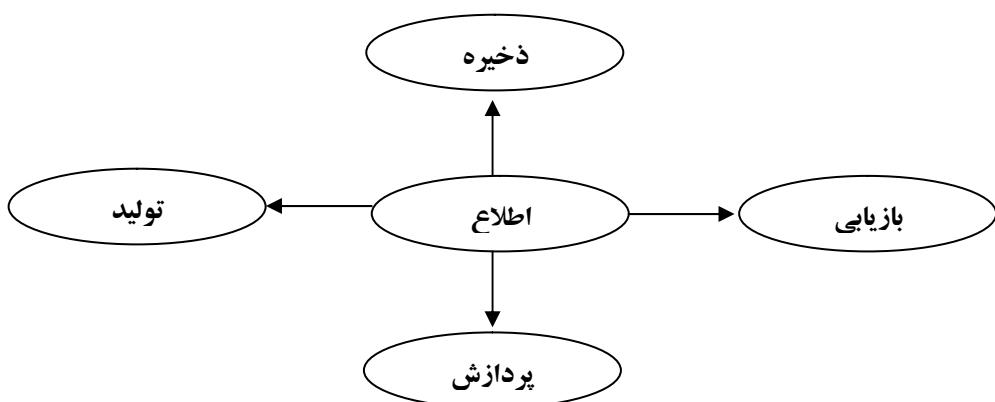
شیوه دستیابی :

این لایه واسطه بین برنامه کاربردی و سیستم فایل منطقی بوده که معمولاً برنامه کاربردی هم با یک زبان سطح بالا نوشته شده است. در واقع یک روش مشخص برای دستیابی به رکوردها در اختیار برنامه کاربردی قرار می‌دهد.

سیستم فایل مجازی :

این لایه همسطح با لایه شیوه دستیابی قرار دارد و گاهی ممکن است لایه شیوه دستیابی تحت این لایه مجازی کند بدین معنا که کاربر برنامه ساز شیوه دستیابی موردنظر خود را طراحی و تولید می‌کند و آن را به واحد سیستم فایل پیوند می‌دهد.

چهار عمل اصلی که انسان با اطلاع انجام می‌دهد :



نامگذاری فایل :

فایل از دید کاربر (بویژه غیر برنامه ساز) یک مکانیسم انتزاعی دارد که امکان ذخیره و بازیابی اطلاعات مورد نیاز خود را به آن می‌دهد لذا برای دستیابی به این فایل هر برنامه‌ای که آن را ایجاد می‌کند باید یک نام هم به آن اختصاص دهد.

قواعد نامگذاری فایل‌ها :

قواعد نامگذاری فایل‌ها در سیستم‌های مختلف یکسان نیست معمولاً نام فایل یک رشته الفبای عددی تا هشت کاراکتر و در برخی سیستم‌ها طول این رشته تا 255 کاراکتر هم می‌باشد. برخی سیستم‌ها به حروف کوچک و بزرگ حساس نیستند (مانند سیستم عامل DOS) ولی در برخی تفاوت دارد (مانند سیستم Unix). در اکثر سیستم‌ها نام فایل از دو بخش تشکیل شده که با «.» از هم جدا می‌شوند.

پسوند فایل . نام فایل

پسوند فایل معمولاً گویای محتوای فایل یا نقش فایل است. MS-DOS دارای پسوند سه حرفی و UNIX تعدادی پسوند دارد

بلاک واحد رد و بدل اطلاعات بین حافظه ای جانبی و حافظه اصلی بین سیستم فایل است که ممکن است در یک عمل I/O چندین بلوک خوانده یا نوشته شود و هر بلوک دارای چندین رکورد می باشد که بین بلوک ها فضاهای بلا استفاده بنام گپ وجود دارد.

تعداد رکوردهای یک بلاک : Bf

R : اندازه یک رکورد

مثال :

می خواهیم فایلی با یک میلیون رکورد 6250 bpi ذخیره کنیم اگر شکاف بین بلاکی (گپ) 0.3 اینچی $Bf = 50$ طول نوار جهت ذخیره سازی فایل را محاسبه کنید.

$Bf = 50$

گپ = 0.3 اینچ

چگالی = 6250 bpi

رکورد = 100

$$\frac{\text{اندازه کل بلاک}}{\text{چگالی}} = \frac{2,000,000}{6250} = 0.016 \text{ inch}$$

$$= 1,000,000 + (0.3 + 0.16) = 316000 \text{ inch}$$

$$\frac{1,000,000}{50} = 20,000$$

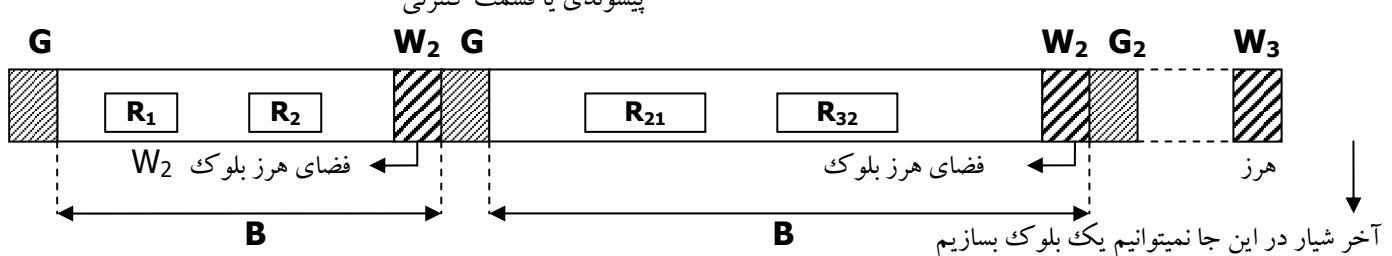
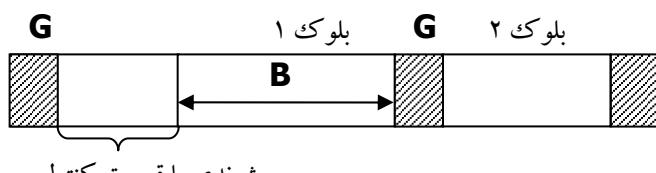
بایت تعداد کل بلوک های فایل

تعداد رکوردهای یک بلاک = Bf

اندازه یک رکورد : R

اندازه یک رکورد = B

$$B_f = \left[\frac{B}{R} \right] = \text{فرمول}$$



حالی که قسمت پیشوندی نداریم از این فرمول استفاده می کنیم:

$$W_B = G + \frac{R}{2} + \frac{W_3}{T_F} \quad \text{فرمول :}$$

فضای هرز بین بلوک ها : W_2

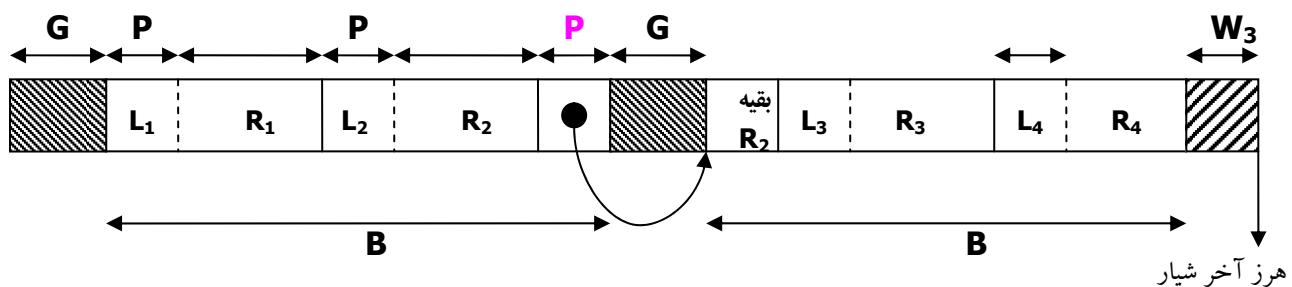
فضای هرز آخر شیار : W_3

تعداد کل بلاکهای یک شیار: T_F

اندازه حافظه هزر به ازای هر بلوک: WB

اندازه حافظه هرز به ازای هر رکورد: WR

* روش اول



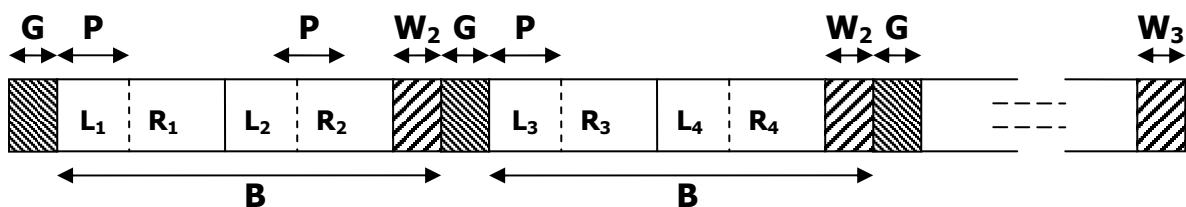
$$W_B = G + P + B_{F*P} + \frac{W_3}{T_F} = \text{فرمول}$$

در حالی که قسمت پیشوندی داریم

* روش دوم

$P = \text{طول} \text{ های یک شیار}$

$$W_R = \frac{W_B}{B_F}$$



* روش سوم

$$W_B = G + \frac{R}{2} + B_{F*P} + \frac{W_3}{T_F} = \text{فرمول}$$

مقایسه روش های بلاک بندی (روش ها یا فرمول ۳ قبلی) :

روش اول : پیاده سازی ساده ولی انعطاف پذیری کمی دارد به طوری که اگر طول رکورد تغییر کند بایستی فایل را از ابتدا بسازیم.

روش دوم : از نظر مصرف حافظه نسبت به روش اول و سوم بهتر است ولی نرم افزار پیچیده ای را نیاز دارد اندازه یک رکورد به اندازه یک بلوک نیست بلکه می تواند اندازه آن از یک بلوک هم بیشتر شود. ممکن است برای خواندن یک بلاک نیاز به فرآخوانی دو یا چند بلاک باشد.

روش سوم : خواندن کل فایل بیشتر از روش دوم زمان می برد . (به علت فضاهای هرز بین بلوک ها (W2) حداکثر طول رکورد می تواند برابر با اندازه یک بلاک باشد (یعنی W2 وجود نداشته باشد)

باکت بندی

به اجتماع چندین بلوک که از دیدگاه سیستم فایل اطلاعات به صورت باکتی رد و بدل می شود باکت بندی گویند مثلا با دستور Read سیستم فایل چند بلوک را همزمان به بافر انتقال می دهد لذا اندازه بافر و باکت همواره برابر می باشد .

لوکالیتی (locality)

به میزان همسایگی رکورد منطقی بعدی نسبت به رکورد فعلی روی حافظه های جانبی را لوکالیتی می گویند .

رکورد منطقی بعدی

رکوردي است که از دید برنامه باید پس از رکورد فعلی مورد پردازش قرار گیرد مانند قرار گرفتن معدل دانشجویان به صورت نزولی

ظروفیت واقعی نوار :

$$\text{درصد واقعی استفاده از نوار} = \frac{B}{B+G} * 100$$

$$\text{میزان استفاده واقعی از نوار (بر حسب بایت)} = \frac{B}{B+G} * Sn$$

مثال :

فایلی با ۲۰۰۰ رکورد ۴۰ بایتی روی نوارهای با چگالی 1000B/inch می خواهد ذخیره شود اگر طول گپ ها برابر یک صدم

اینج باشد مطلوبست :

۱) در صد استفاده واقعی از نوار ؟

۲) طول واقعی فایل (همراه با گپ) بر حسب بایت ؟

۳) طول داده های اصلی فایل (بدون گپ) بر حسب بایت ؟

۴) طولی از نوار بر حسب اینچ که برای ذخیره کردن فایل استفاده می شود ؟

با فرض اینکه $B_f = 1$ می باشد ؟

۲۰۰۰ رکوردي

$\Leftarrow \text{اندازه بلوك} = \text{اندازه رکورد}$

$R=40$

$$R = 40 \Rightarrow B = 40$$

۱۰ بایت = چگالی = 1000 bpi

اینج

$$1) \Rightarrow \frac{B}{B+G} * 100 = \frac{40}{40+10} * 100 = 80\%$$

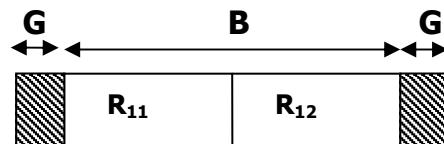
$$2) \Rightarrow b * (B+G) = 2000 * (40+10) = 10000 \text{ بایت}$$

$$\text{بایت } 3) = \text{طول فایل} = b * B = 2000 * 40 = 80000 \text{ بایت}$$

$$4) \frac{B}{B+G} * 100 = \frac{0.4}{0.4 * 0.01} * 100 = 80\%$$

اگر اندازه بلوك ضریب صحیحی از اندازه رکورد باشد می توان از دو رابطه زیر استفاده کرد :

$$\left\{ \begin{array}{l} \text{اندازه فایل} \\ b = \frac{n}{B_f} \\ n * R = b * B \end{array} \right.$$



تعداد رکوردهای یک فایل:

روش های خواندن نوار

۱- روش بلوکی

۲- روش جریانی

روش بلوکی : نوار پس از خواندن هر بلوک می ایستد .

روش جریانی : نوار پس از خواندن تعدادی بلوک می ایستد .

زمان خواندن یک بلوک از نوار بدون در نظر گرفتن گپ :

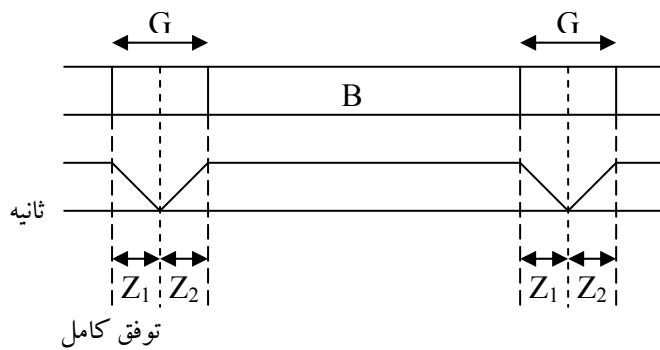
$$b_{tt} = \frac{B}{t}$$

t = سرعت نوار

b_{tt} = زمان خواندن یک بلوک از نوار بدون گپ

Z = زمان حرکت و توقف

t' = نرخ انتقال واقعی نوار



مکانیزم خواندن روش بلوکی

نرخ انتقال واقعی در روش بلوکی

$$T' = \frac{B}{b_{tt} + Z}$$

$$\text{زمان خواندن واقعی یک بلوک نوار} = \frac{B}{t} + z = b_{tt} + z$$

$$\text{زمان خواندن واقعی بلوک N} = Nb_{tt} + N \frac{G}{t} + z$$

تعداد بلوکی که می خوانیم بخوانیم = N

$$\text{زمان لازم برای خواندن گپ های بین N بلوک با سرعت t} = N \frac{G}{t}$$

$$\text{نرخ انتقال در روش جریانی} = \frac{Nb_{tt}}{Nb_{tt} + N \frac{G}{t} + z} * t$$

$$\frac{\text{اندازه بلاک}}{\text{فضایی که بلاک در دیسک می گیرد}} = \frac{*100}{\text{درصد واقعی استفاده از دیسک}} = \frac{B_f \cdot R}{*N} * 100$$

$$N = \left[\frac{B}{\text{طول سکتور}} \right]$$

در دیسک تعداد سکتورهایی که برای ذخیره یک بلوک فایل لازم است: N:

مثال:

فایلی داریم با رکوردهای به طول $B=400$ ، طول هر سکتور $B_f=4$ گر و نیز تعداد سکتورهایی که برای ذخیره هر بلوک مورد نیاز می باشد را پیدا کنید.

بایت $512 = \text{طول یک سکتور}$

بایت $R=400$

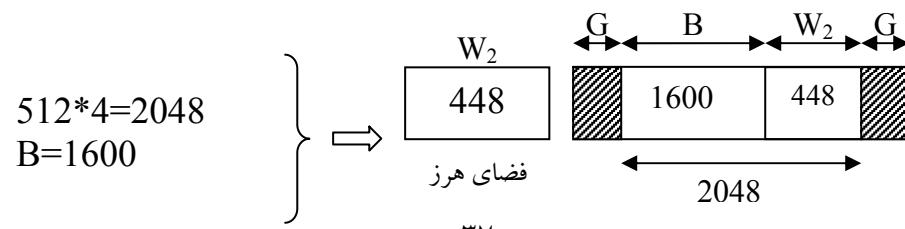
$B_f=4$

در صد واقعی استفاده از دیسک

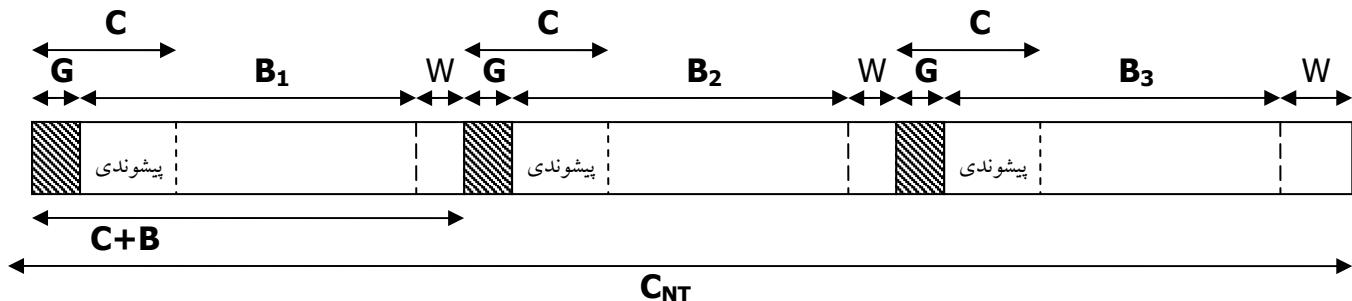
بایت $B_f \cdot R = 4 * 400 = 1600 = \text{طول بلاک}$

$$N = \left[\frac{B}{\text{طول سکتور}} \right] = \left[\frac{1600}{512} \right] = [3.124] = 4 \quad \text{حد بالای جزء صحیح}$$

$$\text{درصد واقعی استفاده از دیسک} = \frac{1600}{512 * 4} * 100 \approx 78\%$$



هر چقدر اندازه ی بلوک در نوار بیشتر باشد (B) به همان ترتیب فضاهای بین بلوک ها (W2) و نیز فضای هرز آخر نوار یا شیار (W3) نیز به علت نگنجیدن یک رکورد و یا یک بلوک بیشتر خواهد بود.



C_{NT}: ظرفیت اسمی شیار

C: ظرفیت کنترلی دیسک (پیشوند + گپ)

B: اندازه یک بلوک

T_f: تعداد بلوکهای شیار

$$T_f = \left[\frac{C_{NT}}{C + B} \right]$$

$$= \frac{T_f * B}{C_{NT}} * 100 = \text{درصد استفاده واقعی از شیار}$$

(B \Rightarrow B - W) به جای B عملیات

مثال:

طول شیار در دیسک ۲۰ هزار بایت است فایلی را با رکوردهایی به طول 100Byte را می خواهیم بر روی این دیسک ذخیره کنیم اگر گپ کنترلی در ابتدای هر بلوک ۲۰ بایت باشد در صورتی که تعداد رکوردهای موجود در هر یک بلوک برابر 5 باشد درصد استفاده واقعی از شیار را بیابید.

$$C_{NT}=20,000 \text{ byte}$$

$$R=100 \text{ byte}$$

$$c=20 \text{ byte}$$

$$B_f = \left[\frac{B}{R} \right] \Rightarrow B = 5 * 100 = 500 \text{ byte}$$

$$T_f = \left[\frac{C_{NT}}{C + B} \right] = \left[\frac{20,000}{20 + 500} \right] = 38$$

$$= \frac{T_f + B}{C_{NT}} * 100 = \frac{38 * 500}{20000} * 100 = 95\% = \text{درصد واقعی استفاده از شیار}$$

پارامترهای موثر بر نرخ انتقال دیسک :

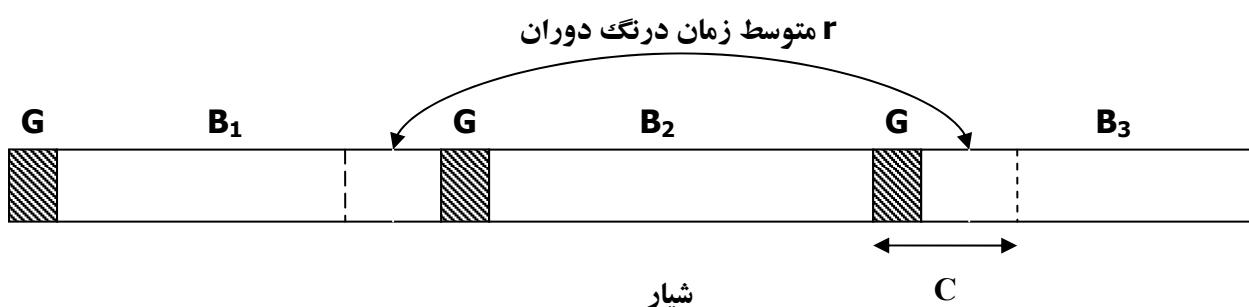
۱. زمان استوانه جویی : S

۲. زمان درنگ دوران : r

۳. زمان لازم برای انتقال یک چند بلاک به طور همزمان به حافظه یا بر عکس از حافظه به دیسک منتقل شود
(اندازه بافر حافظه = تعداد بلاک هایی که همزمان منتقل می شوند).

متوجه زمان درنگ دوران

متوجه زمانی است که طول می کشد تا هد به ابتدای بلاک مورد نظر در یک شیار قرار گیرد.



$$r = \frac{60 * 1000}{2 * rpm} \text{ (ms)}$$

مثال :

اگر سرعت چرخش دیسکی برابر 3600 دور در دقیقه باشد متوجه زمان درنگ دورانی چقدر خواهد بود.

$$r = \frac{60 * 1000}{2 * 3600} = 8.3 \text{ ms}$$

زمان دستیابی تصادفی :

مدت زمان بین لحظه‌ی دستور خواندن یا نوشتن داده می شود و لحظه‌ای که آغاز بلاک زیر هد قرار گیرد در اینجا محل فعلی نوک خواندن و نوشتن نامعین است ولی بلوک که قرار است خوانده شود معین می باشد.

$$b_{tt} \leq 2r$$

b_{tt} زمان انتقال بلوک به چه عواملی بستگی دارد؟

۱. به سرعت چرخش دیسک

۲. چگالی ظبط داده ها

$$\left. \begin{array}{l} \text{نرخ انتقال واقعی دیسک} \\ t = \frac{B}{b_{tt}} \\ \text{زمان دستیابی تصادفی} \\ s+r \end{array} \right\} \Rightarrow \begin{array}{l} \text{نرخ انتقال واقعی} \\ t = \frac{B}{s+r+b_{tt}} \\ \text{نرخ انتقال خیلی دقیقتر} \\ t = \frac{B-W}{s+r+b_{tt}} \end{array}$$

روش های خواندن یک فایل:

۱. ترتیبی (انبوه) مثال: نوار

۲. تصادفی مثال: هارد دیسک، دیسکت ها

b = تعداد بلوک ها در یک دیسک

$b(s+r+b_{tt})$ = زمان خواندن بلوک به طور تصادفی

b'_{tt} : در حالت ترتیبی زمان انتقال بلوکی

t' : نرخ انتقال واقعی در حالت ترتیبی

$$t' = \frac{B}{b'_{tt}}$$

$$b'_{tt} > b_{tt}$$

نرخ انتقال در حالت تصادفی $t' < t$ نرخ انتقال در حالت ترتیبی

مثال: در یک نوع از دیسک های با نام Ipm 3380 اطلاعات به قرار زیر است اندازه بلاک $b_{tt}=0.8$ ، $R=2400\text{byte}$ و $c=600\text{byte}$ ، $r=8.3\text{ms}$ و $t'=2857\text{ms}$ نرخ انتقال ترتیبی $t=3000\text{byte/ms}$ ، $N=88$ ، $s=16\text{ms}$ تعداد سیلندر، $m=3\text{ms}$ ، 10 بلوک =

(الف) خواندن ترتیبی پشت سرهم (انبوه)

(ب) خواندن تصادفی

$$s + r + b * b'_{tt}$$

$$16 + 8.3 + 10 * 0.84 = 32.7\text{ms}$$

$$b(s+r+b_{tt})$$

$$= 10 * (16 + 8.3 + 0.8) = 251\text{ms}$$

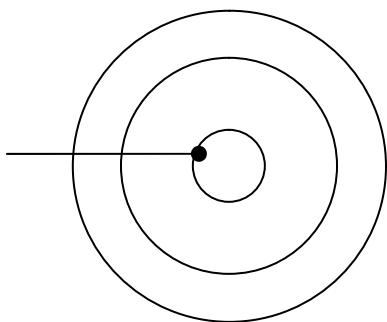
با توجه به اینکه حرکت از یک شیار به شیار دیگر زمان اندکی را جت خاموش و روشن کردن هد صرف می کند این زمان را در مسایل در نظر نمی گیریم.

انواع مکان هندسی قرار گرفتن شیارها (تکنیک قرار گرفتن شیارها) :

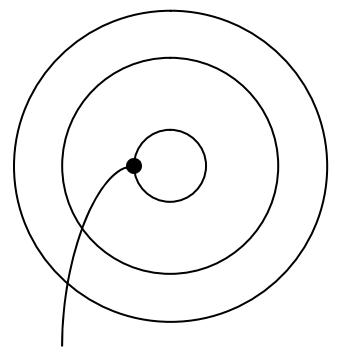
۱. روش معمولی

۲. Track stagger

در روش معمولی نقطه شروع شیارها بر روی شعاعی از صفحه ئیسک قرار گرفته است ولی در روش Track stagger نقطه شروع هر شیار عبارت نسبت به شیار قبلی قدری جلوتر است و این فاصله به اندازه ای است که گام حرکت هد از یک شیار به شیار مجاورش درست در اول آن شیار قرار بگیرد. لذا زمان درنگ دوران فقط برای اولین سیلندر در نظر می گیرند.



معمولی



شكل Track Stagger

$$b' = s + r + b * b_{tt} \Rightarrow b * b_{tt}$$

$$b = b * (s + r + b_{tt})$$

$$n = n * (s + r + b_{tt})$$

$$B_f = \frac{B}{R} \quad \text{دوپاره} \quad B_f = \left[\frac{B}{R} \right] \quad \text{یکپاره}$$

$$b = \frac{n}{B_f} \quad \text{فرمول در هم یکپاره و هم دوپاره}$$

$$n * R = b * B \quad (\text{طول داده های فایل بر حسب بایت (بدون گپ)})$$

(در آخر هر بلاک فضای هرز نباشد) $W3 = \text{طول بلاک} \times \text{مضرب صحیحی از طول رکوردها}$ باشد.

مثال :

فایل شامل ۳۰۰۰۰ رکورد است و هر رکورد ۱۶۰۰ بایت طول دارد طول هر بلاک نیز ۲۴۰ بایت می باشد. زمان خواندن کل فایل را در دو حالت زیر به دست آورید.

الف) رکوردها در بلاک دوپاره باشند

ب) رکوردها بتوانند دوپاره در بلاک ها ذخیره شوند.

n= 3000 عدد

B=2400byte

R=1600byte

$$B_f = \left[\frac{B}{R} \right] = \left[\frac{2400}{1600} \right] = 1 \quad \text{الف) }$$

$$\text{ثانیه} = \frac{n}{B_f} * b'_n = \frac{30000}{1} * 0.84 = 25200ms = 25.2$$

$$b) n * R = b * B \Rightarrow b = \frac{30000 * 1600}{2400} = \frac{30000 * 2}{3} = 2000$$

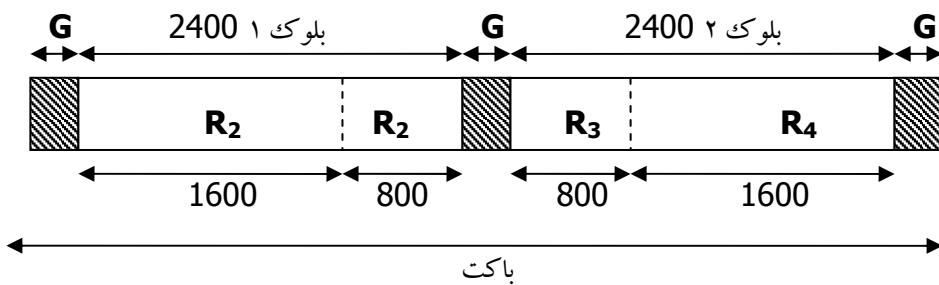
$$\text{ثانیه} = b * b'_n = 20000 * 0.84 = 16800ms = 16.8\text{ sec}$$

نتیجه :

با توجه به اعداد حاصله برای خواندن کل فایل در حالت دوپاره و حالت یکپاره نتیجه می‌گیریم که در حالت دوپاره و هم از فضای دیسک بیشترین استفاده را می‌بریم و هم زمان خواندن کل فایل کمتر از حالت یکپارچه می‌باشد.

اثر باکت بندی در سرعت عملیات دیسک:

فرض کنید فایلی با رکوردهای ۱۶۰۰ بايت و اندازه بلوک ۲۴۰۰ بايتی در رکوردها می‌توانند دوباره در این حافظه ذخیره شوند.



به خاطر اینکه ما رکورد دوم را بخوانیم مجبور هستیم بلوک ۱ به همراه بلوک ۲ در حافظه بار کنیم. یعنی عمل باکت بندی صورت گرفت لذا اندازه بافر موجود در حافظه با اندازه باکت برابرند.

بافر و باکت :

مجموع چند بلاک پشت سر هم در دیسک تشکیل باکت و مجموع چند بلاک پشت سر هم در حافظه تشکیل بافر می‌دهد.

تفاوت بین اندازه باکت با اندازه بلاک :

اندازه باکت توسط ساختار فایل و برنامه ریز تعیین می گردد حال آنکه اندازه بلاک اغلب توسط ساختار برنامه معین نمی شود و بلاک حداقل رد و بدل داده ها بین سیستم عامل و دیسک می باشد . اندازه بلاک در دیسک اغلب برابر اندازه page در سیستم عامل هاست .

مثال :

فایلی با رکوردهای ۱۶۰۰ بایتی در دیسک IBM3380 با بلوک های ۲۴۰۰ بایتی به صورت دوپاره ذخیره شده است اگر هر باکت شامل دو بلاک پی در پی باشد زمان خواندن یک باکت با توجه به مشخصات دیسک IBM3380 زمان خواندن یک باکت را بنویسید .

$$زمان خواندن یک باکت در حالت دوپاره = 8.3 + 16 + 0.8 + 0.84 = 25.94ms$$

حل روش دوم :

زمان خواندن یک باکت که شامل k بلاک

$$= 8.3 + 16 + 2 * 0.84 \approx 25.94ms$$

تذکر :

باکت بندی در زمان خواندن (تریبی) پی در پی فایل اثر ندارد باکت بندی بر روی زمان خواندن تصادفی فایل اثر می گذارد .

چرا که مجبور هستید برای خواندن b بلاک پشت سر هم زمان $b * b'_{tt}$ را صرف نماییم .

$$\text{زمان خواندن تصادفی فایل بر اساس باکت} = b * (r + s + d_{tt})$$

تعداد باکت های فایل = bk

، زمان انتقال یک بلاک : d_{tt}

زمان خواندن تصادفی فایل رکورد به رکورد = n(r+s+d_{tt})

نتیجه :

$$\text{زمان خواندن تصادفی باکت ها} = bk(r+s+d_{tt}) = bk(r+s) + bk * d_{tt}$$

در هنگام خواندن تصادفی باکت ها با توجه به فرمول بالا بزرگتر بودن باکت باعث کاهش مقادیر bk و در نتیجه کاهش مقدار bk(r+s) می شود ولی تاثیری در bk * d_{tt} ندارد چرا که با بزرگتر کردن باکت d_{tt} بیشتر می شود و در نتیجه اثری بر مقدار bk * d_{tt} ندارد .

لذا بزرگتر کردن طول باکت زمان خواندن تصادفی فایل را به صورت باکتی کاهش می دهد ولی خواندن تصادفی فایل به صورت رکوردي را افزایش می دهد .

فایلی با ۳۰۰۰۰ رکورد ۱۰۰ بایتی در دیسکی با مشخصات زیر ذخیره شده است.

$$b'_{tt} = 0.84, b_{tt} = 0.8 \text{ ms}, r = 8.3 \text{ ms}, B = 2400, S = 16 \text{ ms}$$

اگر رکوردها به صورت دوپاره در بلاک ذخیره شده باشند زمان خواندن تصادفی باکت‌ها و رکوردهای فایل را برای دو حالت زیر حساب کنید:

الف) هر باکت دو بلاک داشته باشد.

ب) هر باکت یک بلاک داشته باشد

$$\text{الف) } d_{tt} = b_{tt} + b'_{tt} = 0.8 + 0.84 = 1.64 \text{ ms}$$

$$\text{زمان خواندن تصادفی رکوردها} = n(r+s+d_{tt}) = 30000(16+8.3+1.64) = 778200 \text{ s}$$

$$B_f = \frac{2400}{100} = 24 \quad b = \frac{30000}{24} = 1250$$

$$bk = \frac{1250}{2} = 625 \quad \text{تعداد باکت‌ها}$$

$$\text{زمان خواندن تصادفی بلاک‌ها} = bk(r+s+d_{tt}) = 625(16+8.3+1.64)$$

$$= 16212.5 \text{ ms} \approx 16 \text{ ثانیه}$$

ب) هر باکت یک بلاک است پس:

$$d_{tt} = b_{tt} = 0.8 \text{ ms}$$

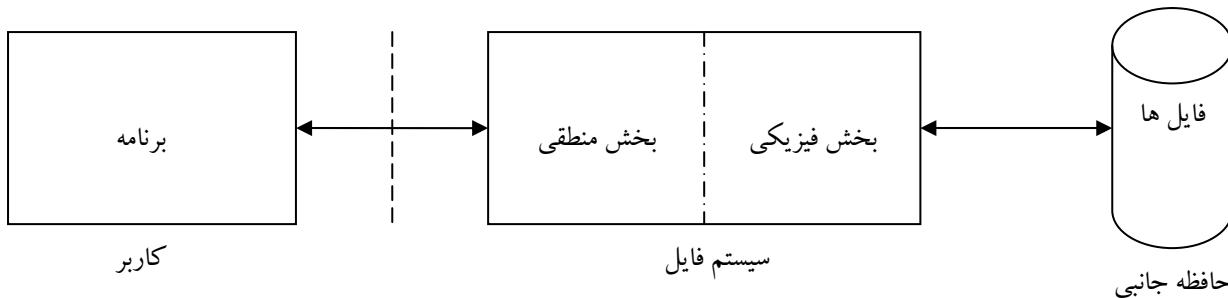
$$\text{زمان تصادفی خواندن رکوردها} = n(r+s+d_{tt}) = 30000(16+8.3+0.8) = 753000 \text{ ms} \approx 753 \text{ s}$$

$$Bk = b = 1250 \quad \text{تعداد باکت‌ها}$$

$$\text{زمان تصادفی خواندن باکت‌ها} = bk(r+s+d_{tt}) = 1250(16+8.3+0.8) = 31375 \text{ ms} \approx 31 \text{ s}$$

سیستم فایل :

بخشی از سیستم عامل است که با فایل سروکار دارد و هر سیستم فایل را می توان به دو بخش فیزیکی و منطقی تقسیم کرد که بخش منطقی آن مربوط به دستورات کاربر است نظیر باز و بسته کردن و خواندن و نوشتن فایل ها اما قسمت فیزیکی آن به طور غیر مستقیم به فایل های موجود بر روی حافظه جانبی سروکار دارد . در واقع این قسمت دستوراتی را که از بحث منطقی سیستم فایل وارد شده را به دستوراتی جهت صدور به رسانه ذخیره سازی تبدیل می کند .



عملیاتی که سیستم فایل ها انجام می دهند :

- ۱- مکان یابی
- ۲- خواندن
- ۳- نوشتن بر روی حافظه های جانبی به قسمت فیزیکی سیستم فایل AM (Access Method) نیز می گویند .

سطوح مختلف آدرس دهی :

- ۱- آدرس دهی در برنامه :
 - الف) محتوایی یا مقداری
 - ب) آدرس دهی نسبی
 - ج) آدرس دهی نمادین (سمبلیک)
- ۲- آدرس دهی در سطح فیزیکی سیستم عامل :
 - الف) محتوایی یا مقداری :

کاربر مقدار یک چیز یا چند صفت خاصه را جهت جستجو می دهد این صفت خاصه می تواند کلید باشد یا نباشد .

ب) آدرس دهی نسبی :

در این حالت کاربر فایل را به صورت یک ساختار خطی می بیند که هر رکورد یک شماره دارد و اغلب این شماره با عدد صفر آغاز می گردد .

ج) آدرس دهی نمادین (سمبلیک) :

در این حالت رکورد دلخواه توسط یک نام شخص آدرس دهی می شود .

۲- آدرس دهی منطقی در سطح سیستم عامل :

در این روش آدرس دهی کل فضارسانا و فایل به صورت یک آرایه از بلاک ها می باشد که هر بلاک آدرس منحصر بفردی دارد. که اصطلاحا به آن RBA (Relative Block Address) یا آدرس نسبی بلاک گفته می شود. در واقع در این سطح آدرس تولید شده در سطح برنامه دریافت گردیده و به آدرس RBA تبدیل می شود و سپس به قسمت فیزیکی سیستم فایل می فرستد.

۳- آدرس دهی در سطح فیزیکی سیستم عامل :

در بخش فیزیکی سیستم فایل آدرس RBA را از بخش منطقی دریافت کرده و آن را به آدرس دهی خاص رسانه ذخیره سازی تبدیل می کند مثلا در مورد دیسک آدرس RBA را به شماره هد، سیلندر و شماره سکتور تبدیل می کند.

مثال :

کاربری در سطح برنامه خود دستور فرضی زیر را وارد می کند .

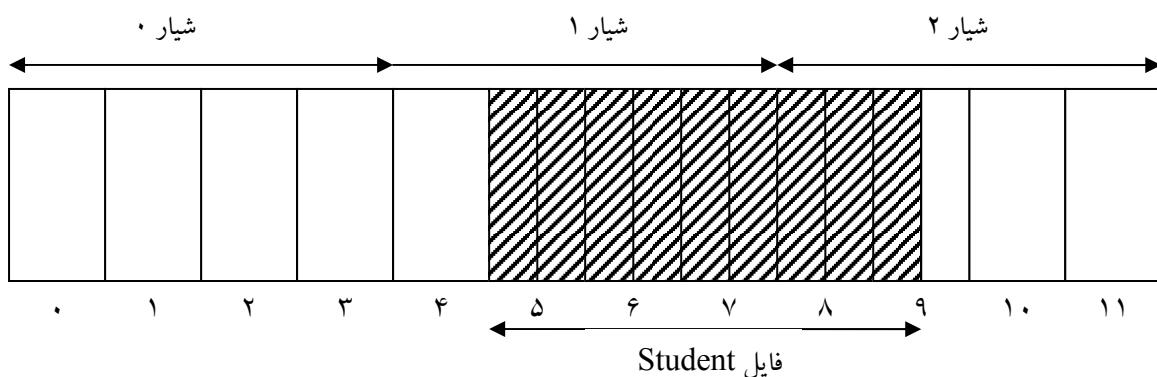
Read from student Record=6

یعنی رکورد ششم از فایل Student خوانده شود. تعداد رکوردهای منطقی فایل ۹ عدد بوده و طول هر رکورد منطقی ۱۰۰ بایت و طول هر بلاک ۲۰۰ بایت است. فایل به صورت پیوسته بر روی دیسک ذخیره شده و هر شیار حاوی ۴ بلاک باشد آدرس RBA شروع فایل بر روی دیسک برابر ۵ باشد. مطلوبست: (شماره RBA از صفر آغاز می شود.)

الف) رکورد ششم فایل از کدام بایت آغاز می شود ؟

ب) RBA حاوی رکورد ششم نسبت به ابتدای فایل چه می شود ؟

ج) RBA حاوی رکورد ششم نسبت به اولین بلاک دیسک چه می شود ؟



الف) رکورد ششم از بایت شماره 500 فایل آغاز می شود :

$$(6-1)*100=5*100=500$$

ب) با توجه به شکل مشخص است که RBA رکورد ششم نسبت به ابتدای فایل برابر ۲ است به عبارتی دیگر رکورد ششم در سومین بلاک فایل قرار دارد.

ج) با توجه به شکل رکورد ششم فایل در $RBA = 7$ نسبت به اول دیسک قرار دارد از طریق محاسبه داریم :

(رکورد مورد نظر در فایل) $RBA = RBA + (اول فایل)$ نسبت به اول دیسک

$$RBA = 5 + 2 = 7$$

رکورد ششم آدرس دهی در سطح برنامه است آدرس دهی در سطح برنامه است بخش منطقی سیستم فایل این آدرس را گرفته و به $RBA = 7$ تبدیل می کند . بخش فیزیکی این فایل عدد 7 را گرفته و در می یابد که کدام قسمت فیزیکی دیسک را باید بخواند (بلوک چهارم از شیار شماره یک) . سیستم فایل بلوک مورد نظر را (بلوک) خوانده و در بافر قرار می دهد سرانجام سیستم فایل داده ها را از بلاک بندهی درآورده (رکورد به رکورد) و رکورد دوم این بلاک را به ناحیه کاری کاربر می فرستد .

فرمول ها :

$$\text{آفست رکورد } i \text{ ام نسبت به اول فایل بر حسب بایت} = (i-1) * R$$

$$= (6-1) * 100 = 500$$

$$\text{آفست بلوک حاوی رکورد } i \text{ ام نسبت به ابتدای فایل} = \left\lceil \frac{(i-1) * R}{B} \right\rceil = \left\lceil \frac{500}{200} \right\rceil = 2$$

آفست بلوک حاوی رکورد نسبت به اول فایل + شماره بلوک آغاز فایل = شماره بلوک حاوی رکورد نسبت به ابتدای دیسک

$$cyl\# = \left\lceil \frac{RBA}{t * b} \right\rceil \quad \text{تعداد شیارهای موجود در هر سیلندر :}$$

تعداد بلوک های موجود در هر شیار : b

$$t * b = \text{ظرفیت هر استوانه بلاک}$$

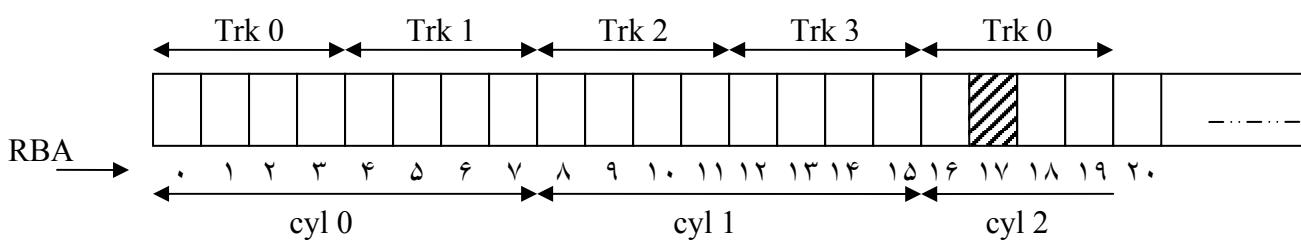
آدرس فیزیکی : $(cyl\#, trk\#, blk\#)$

$$Trk\# = \left\lceil \frac{RBA \mod (t * b)}{b} \right\rceil$$

$$blk\# = RBA \mod b$$

مثال :

دیسکی داریم که هر شیار آن شامل چهار بلاک است و هر سیلندر آن دو شیار دارد . آدرس فیزیکی بلوکی $RBA = 17$ را به دست آورید .



B=14

t=2

RBA = 17

$$\text{Blk}\# \text{RBA} \bmod b = 17 \bmod 4 = 1$$

$$trk\# = \left[\frac{RBA \bmod (t * b)}{b} \right] = \left[\frac{17 \bmod (2 * 4)}{4} \right] = \left[\frac{1}{4} \right] = 0$$
$$cyl\# = \left[\frac{RBA}{t * b} \right] = \left[\frac{17}{2 * 4} \right] = 2$$

جدول راهنمای فایل : (file directory

در سیستم فایل جدول راهنمای فایل وجود دارد که در آن اطلاعات مربوط به فایل از قبیل نام فایل ، آدرس اولین بلاک آن ، طول رکورد ، تعداد رکوردها ، طول فایل ، تاریخ ایجاد فایل ، تاریخ آخرین بروز رسانی فایل و ... ذخیره می گردد .

بافر :

جهت ایجاد هماهنگی بین سرعت وسیله I/O و پردازنده سیستم ناحیه ای در حافظه‌ی بنام بافر در نظر گرفته شده است اگر فایل بلاک بندی شده باشد حداقل یک بلاک در بافر قرار داده می شود (یعنی خوانده می شود) و اگر فایل بلاک بندی نشده باشد در هر بار عملیات I/O یک رکورد در بافر قرار می گیرد . سیستم‌های I/O همواره حداقل دو بافر دارند یکی جهت ورودی و دیگر برای خروجی در غیر این صورت کارایی سیستم I/O کاهش می یابد .

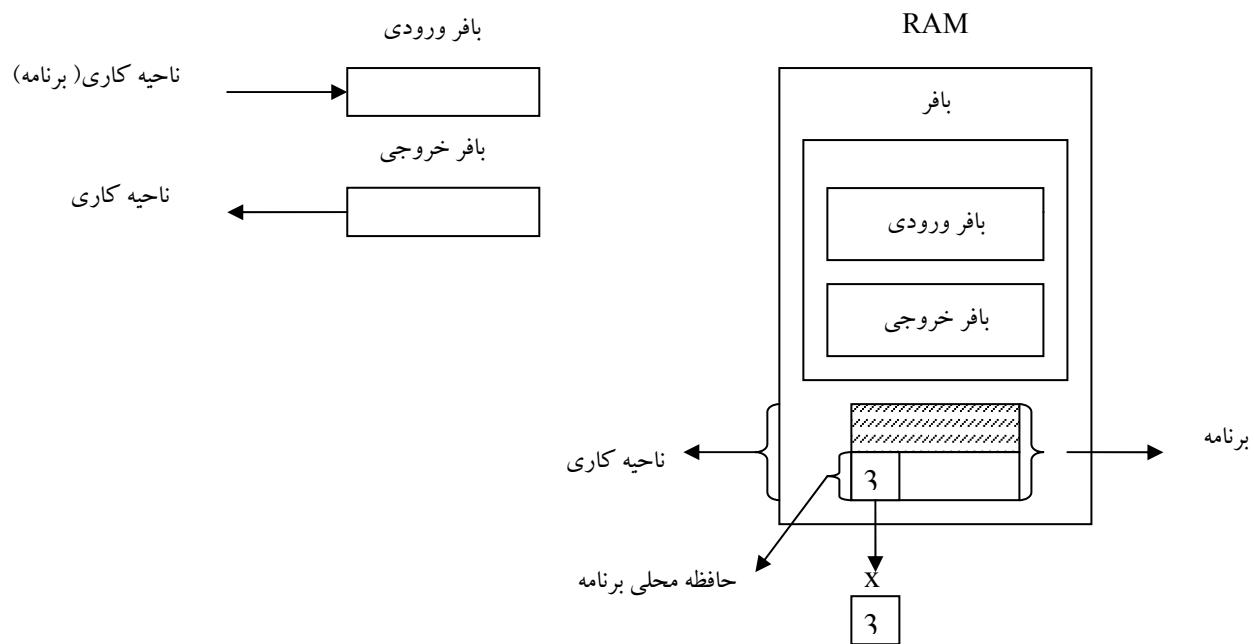
روش‌های دستیابی برنامه کاربر به محتويات بافر :

۱. روش انتقالی یا حرکتی (move mode)

۲. روش مکان نمایی یا تعیین محل (locate mode)

روش انتقالی یا حرکتی :

در این حالت ناحیه بافر جدا از حافظه مربوط به برنامه (محل حافظه‌ای برنامه در آن ذخیره شده ناحیه کاری) می باشد . و برنامه مستقیماً به بافر سیستم دسترسی ندارد بلکه بافر ویژه خود را دارد . در این روش بلاک از ناحیه کاری به بافر خروجی یا از بافر ورودی به ناحیه کاری فرستاده می شوند . لذا در این روش داده‌ها قبل از دستیابی به محلی در حافظه به محل دیگر انتقال می یابند که این زمان نسبتاً زیاد می باشد .



روش مکان نمایی یا تعیین محل :

به دو روش می توان از حرکت موجود در تکنیک انتقالی اجتناب کرد . اول اینکه داده های ورودی و خروجی مستقیما بین وسیله ورودی و خروجی و ناحیه کاری برنامه رد و بدل شود . راه دوم مدیریت فایل از بافرهای سیستم برای همه ورودی و خروجی ها استفاده می کنند .

روش های دستیابی به بافر :

۱. روش انتقالی در ورودی و خروجی
۲. روش مکان نمایی در ورودی و خروجی
۳. روش انتقالی در ورودی و مکان نمایی خروجی
۴. روش مکان نمایی در ورودی و انتقالی در خروجی

روش مدیریت بافر توسط سیستم عامل انجام می گیرد و خیلی کم توسط برنامه نویس انجام می شود .

مدیریت بافر به روش استخراج بافری :

وقتی که به یک بلوک نیاز پیدا می کنیم سیستم ابتدا بلوک مورد نظر را در بین بلوک های موجود در بافر جستجو می کند در صورت عدم وجود این بلوک در بافرها (بافر ورودی ، خروجی) یکی از بافرهای بلااستفاده را پیدا کرده و بلوک مورد نظر را از حافظه جانبی به آن بافر بار می کند . در صورتی که بافر خالی نباشد برای تصمیم گیری به این موضوع راه های متفاوتی توسط مدیریت بافر (سیستم عامل) اتخاذ می شود . مثلا بافری که آخرین بار استفاده شده انتخاب می شود .

*توجه :

مدیریت بافر به گونه ای عمل می کند که کارایی حافظه کم شود . مثلا اندازه بافرها را بیش از اندازه بزرگتر در نظر نمی گیرد چرا که هر چقدر تعداد بافرها بیشتر باشد سیستم عامل زمان بیشتری را برای مدیریت آنها بایستی صرف کند .

انواع بافرینگ (بافردهی) :

بافردهی یعنی تعداد بافرهایی که به برنامه اختصاص داده می شود و سه نوع هستند :

۱- بافر دهی ساده

۲- بافردهی دو گانه

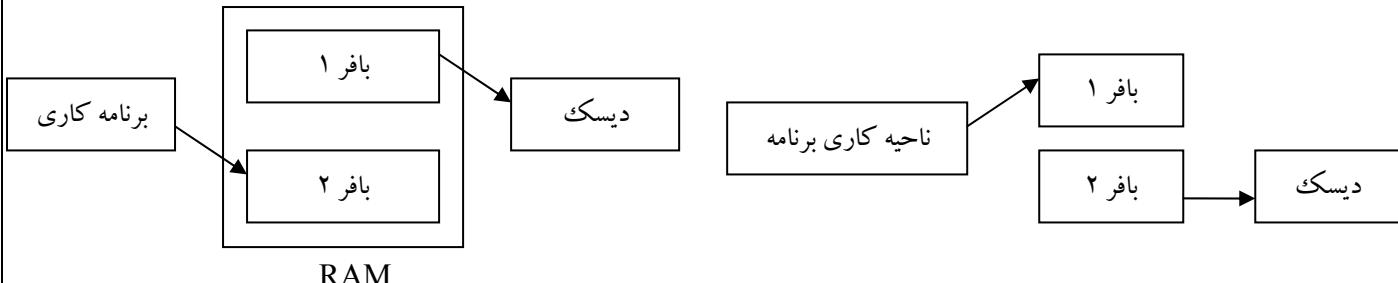
۳- بافردهی چند گانه

۱- بافردهی ساده یا یگانه : single buffering

در این روش از یک بافر ورودی و خروجی استفاده می شود لذا کارایی برنامه کاهش می یابد چرا که بافر اگر در حال پرشدن باشد cpu بایستی منتظر مانده (بیکاریcpu) و بعد از عمل پرشدن بافر می تواند از محتویات آن استفاده نماید . در سیستم عامل به این زمان بیکاری cpu زمان انتظار (Idle) می گویند . که در سیستم های چند برنامه ای از این زمان برای اجرای برنامه های دیگر استفاده می شود .

۲- بافردهی دو گانه یا مضاعف : Double Buffering

برای از بین بردن شکل روش اول (زمان انتظار) از دو بافر برای عملیات I/O و cpu استفاده می شود . لذا پردازنده می تواند در هنگام ارسال یک بافر به دیسک بافر دیگر را پر کند .



در این روش برای اینکه کارایی بافر خوب باشد بایست همواره $C_B \leq b_{tt}$ باشد .

$$C_B \leq b_{tt} \equiv C_B \leq \frac{B+G}{t}$$

$$C_R \leq \frac{R+W_R}{t}$$

زمان پردازش یک رکورد

اگر $C_B > b_{tt}$ باشد بافرینگ مضاعف کارایی ندارد .

وقتی که پردازش فایل به صورت ترتیبی (انبوه) باشد . حتما بایستی از بافرینگ مضاعف استفاده شود در غیر این صورت کارایی

برنامه کاهش می یابد . رابطه دقیق تر از $C_B \leq b_{tt}$ برابر $C_B \leq b_{tt} + \frac{G}{t}$ می باشد .

۳- بافردهی چند گانه : Multiple Buffering

با استفاده از بافردهی چند گانه موقع پردازش پی در پی و ابوه فایل می توانیم بلوک های فایل را جلوتر (از پیش) خوانده و د فایل ذخیره نماییم که به این روش صفت بندی نیز می گویند چرا که سیستم رکوردهای بعدی را که هنوز cpu آن را نخواسته به بافر انتقال داده است به بافردهی چند گانه صفت چرخشی هم (بافر چرخشی) می گویند .

تذکر :

هر چند که با فردی هی چند گانه تا حدود زیادی از زمان انتظار CPU می کاهد اما در صورتی که سرعت پردازش خیلی بیشتر از سرعت عملیات با فرینگ باشد باز پردازنده مجبور است انتظار بکشد (در سیستم های چند برنامه ای این زمان CPU به فرآیندهای دیگر انتقال می یابد).

روش بیس برای بافردهی (Base) :

اگر سیستم از بافر استفاده نکند نمی توان عملیات بلاک بندی را انجام داد. لذا مجبور هستیم رکورد داده ها را به ناحیه کاری برنامه انتقال دهیم که اصطلاحاً به این عمل روش بیس (پایه) می گویند.

تذکر : اگر برنامه از روش کاری مکان نمایی استفاده نماید (یعنی ناحیه کاری نداشته باشد) بلاک های فایل را نمی توان جلوتر (از پیش) خواند.

عواملی که به کارایی سیستم فایل بستگی دارند :

۱- اندازه بلاک بندی و نحوه بلاک بندی

۲- لوکالیتی رکوردها

۳- پارامترهای رسانه (ظرفیت واقعی نوار، نرخ انتقال واقعی نوار و)

۴- نحوه آرایش سلسله مراتب حافظه

۵- نحوه بافردهی و مدیریت آن

۶- مدیریت فایل ها در سیستم عامل

۷- روش ها کاهش زمان پیگرد در درنگ دورانی

۸- استفاده از فایل با کارایی بیشتر

اثر بافرینگ در سرعت عملیات:

نرخ انتقال انبوه یا ترتیبی به عوامل زیر بستگی دارد

۱- نوع بافردهی (بافرینگ)

۲- طول بافر

۳- CB زمان لازم پردازش محتویات یک بلاک

۴- نوع پردازش (ترتیبی یا تصادفی)

۱- بافردهی ساده و پردازش ترتیبی :

در این نوع بافرینگ نمی توانیم عملیات پردازش و انتقال را همزمان انجام دهیم چرا که در پردازش ترتیبی حتماً بلاک های فایل باشیستی به ترتیب خوانده شوند. لذا در این روش زمان خواندن هر بلاک برابر با $2r+btt$ می باشد. برای مثال برای لیست کردن رکوردها به ترتیب شماره آنها پس از آن که بلاک اول خوانده شد (انتقال بلاک مورد نظر از دیسک به حافظه اصلی) CPU

مشغول پردازش آن می شود و در این هنگام چرخش دیسک هم ادامه دارد. لذا با توجه به اینکه پردازش ترتیبی می باشد برای اینکه بلوک بعد از آن خوانده می شود. آن بلوک دوم از زیر هد رده است لذا مجبور هستیم که دیسک یک چرخش کامل را در زمان $2r$ بزنده همین دلیل می گوییم که زمان خواندن هر بلوک در بافرینگ ساده به روش پردازش ترتیبی $2r + b_{tt}$ می باشد. لذا برای خواندن ترتیبی فایل در همین روش با شرط $CB < 2r$ باشد.

$$C_B < 2r \quad t' = \frac{B}{2r + b_{tt}}$$

نرخ انتقال انبوه (ترتیبی) برای فایل بافرینگ ساده به روش ترتیبی

* توجه :

در صورتی که $CB < 2r$ باشد زمان خواندن هر بلوک برای بافرینگ ساده با پردازش ترتیبی $2r + b_{tt}$ و نرخ انتقال انبوه برای

$$t' = \frac{B}{2r + b_{tt}}$$

خواندن ترتیبی فایل چنان چه زمان پردازش یک بلوک $CB < 4r$ باشد نرخ انتقال واقعی برای

$$t' = \frac{B}{4r + b_{tt}}$$

۲- بافرینگ ساده بدون پردازش ترتیبی :

منظور از پردازش بدون ترتیبی فایل خواندن تمامی رکوردها بدون در نظر گرفتن ترتیب خواندن آن رکوردها می باشد در این حالت اگر $CB \leq b_{tt}$ باشد کل بلاک های موجود در یک شیار در دور زدن دیسک (در زمان $4r$) می توان خواند پس نرخ

$$t' = \frac{T_f \cdot B}{4r} \quad \text{انتقال انبوه در این نوع بافردهی و پردازش بدون ترتیبی با شرط } CB \leq b_{tt} \text{ مساوی با}$$

اندازه یک شیار: $T_f \cdot B$

۳- بافرینگ مضاعف با شرط کارایی با پردازش ترتیبی یا بدون ترتیبی :

$$\left(C_B \leq \frac{B+G}{t} \right)$$

کارایی در پردازش ترتیبی و بدون ترتیبی در بافرینگ مضاعف با شرط $C_B \leq \frac{B+G}{t}$ یعنی تمام بلاک های موجود در یک شیار چه در پردازش های ترتیبی و چه در پردازش تصادفی با یک دور چرخیدن دیسک خوانده می شود لذا نرخ انتقال انبوه

$$t' = \frac{T_f \cdot B}{2r} \quad \text{(ترتیبی) و تصادفی در بافرینگ مضاعف با شرط کارایی به صورت } 2r \text{ می باشد.}$$

۴- بافرینگ مضاعف بدون شرط کارایی و پردازش ترتیبی :

$$C_B > \frac{B+G}{t}$$

در این روش چون کارایی را نداریم پس $C_B > \frac{B+G}{t}$ می باشد. یعنی دو بلاک در بافر خوانده شده و پردازش نیز همزمان صورت می گیرد و قتنی که هد بر سر بلاک سوم می رسد نمی توانند خوانده شوند چرا که به علت عدم کارایی بافر اول هنوز پردازش نشده است. لذا برای خواندن دو بلاک بعدی (بلاک شماره ۳ و ۴) باید دیسک یک دور کامل ($2r$) را بزنده یعنی برای هر اندازه بلوک B بایت زمان $2b_{tt} + 2r$ صرف می شود لذا نرخ انتقال انبوه با بافرینگ مضاعف بدون شرط کارایی

$t' = \frac{2B}{2b_{tt}' + 2r} = \frac{B}{b' + r_{tt}}$

می باشد . که در این فرمول زمان خواندن کل فایل در بافرینگ مضاعف بدون شرط کارایی با پردازش ترتیبی $= b^*(b_{tt}' + r)$ زمان خواندن کل فایل می باشد .

*توجه :

در فرمول های گفته شده فضای هرز موجود در هر بلاک (WB یا WR) در نظر گرفته نشده اند بهترین روش برای خواندن و

$MAX(t) = \frac{T_f \cdot B}{2r}$

پردازش و بافرینگ موقعي است که حداکثر نرخ انتقال یعنی در يك دور زدن ديسك بتوانيم تمام بلاک های شiar را خوانده و پردازش کنيم .

اثر طول بافر و سرعت عملیات ديسك :

استفاده از بافری به طول يك شiar تاثير زيادي را می تواند در افزایش سرعت عمل ديسك داشته باشد .

مثال :

سيستمي از بافرينگ مضاعف استفاده می کند که طول هر بافر آن برابر با طول يك بلاک می باشد چنان چه در هر شiar ۲۰ بلاک موجود باشد زمان پردازش يك بلاک (CB) با شرط $C_B > b_{tt}'$ و عدم کارايي با بافرينگ مضاعف ۳ برابر زمان خواندن داده های آن بلاک می باشد . زمان خواندن و پردازش ترتیبی کل فایل چه ميزان می باشد .

بلاک اول در بافر اول خوانده می شود پس بلاک دوم در بافر دوم خوانده شده و همزمان پردازش بافر اول انجام می شود هنگامی که هد بر سر بلاک دوم می رسد هنوز پردازش بلاک اول تمام نشده است (زمان پردازش يك بلاک ۳ برابر زمان خواندن داده هاست) و چون ديسك دائم در حال چرخش است . لذا با اين شرایط هیچ عمل O/I صورت نگرفته و لذا هد از روی بلاک سوم ردي شود با اتمام پردازش بافر اول (بلاک اول) هد بر سر بلاک پنجم می رسد . به همين دليل بایستي يك دور كامل زده شود تا هد دوباره بر سر بلاک سوم برسد . که در اين هنگام هم می توان بافر دوم را پردازش نمود و به همين ترتيب می توان گفت بلاک های سوم و چهارم خوانده می شود اگر ديسك يك دور كامل زده شود ولی ديگر در اين لحظه نمی توانيم بلاک های پنجم و ششم را بخوانيم و برای خواندن آن بلاک ها نياز هست که ديسك يك دور كامل نيز بزند .

بنابراین خواندن و پردازش هر دو بلاک پشت سرهم به زمان $2b_{tt}' + 2r$ نياز هست . و به همين جهت خواندن و پردازش ترتیبی کل فایل با b بلاک $b^*(b_{tt}' + r)$ می باشد .

زمان modifying یا زمان اصلاح يك رکورد :

می خواهیم رکوردي را خوانده بعد از ویرایش و انجام اصلاحات لازم دوباره آن را به مکان قبلی خود قرار دهیم (خواندن ذخیره

کردن) زمان خواندن بلاک حاوی آن رکورد $r + s + d_{tt}$ می باشد . پس از آن که هد اين بلاک را خواند زمان $-b_{tt} - 2r$ نياز هست تا هد به اول آن بلاک مورد نظر برسد . با فرض اينکه در اين مدت اصلاحات لازم بر روی رکوردي مورد نظر

انجام شده است لذا مدت زمان b_{tt} نیاز هست تا رکورد مورد نظر بر سر جای خود نوشته شود. لذا می توان زمان خواندن، اصلاح و نوشتن یک رکورد برابر با $(s + r + b_{tt}) + b_{tt} + (2r - b_{tt})$ می باشد. لذا می توان گفت که اصلاح با 3^3 عمل خواندن، پردازش و نوشتن صورت می گیرد.

زمان اصلاح یک بلوک :

زمانی که $C_B \leq b'_{tt}$ می باشد و اندازه بافر با طول شیار برابر است. زمان اصلاح تقریباً دو برابر زمان خواندن ترتیبی فایل می باشد. ($2 * b * b'_{tt}$ می باشد. یعنی $b * b'_{tt}$ طول می کشد تا b بلاک خوانده شود و پردازش گردند و نیز b'_{tt} طول می کشد تا بلاک های صالح شده دوباره روی دیسک نوشته شوند.

مثال :

فایلی با ۱۰ هزار بلاک روی دیسک IBM 3380 که دارای مشخصات زیر می باشند :

$$b'_{tt} = 0.84ms \quad b_{tt} = 0.8ms \quad B = 2400byte \quad r = 8.3ms \quad b = 10000$$

= ۲۰ تعداد بلاک ها در هر شیار

مطلوبست محاسبه زمان اصلاح کل فایل :

الف) بافری به طول یک بلوک و $C_B < 2r$

ب) بافری به طول دو بلوک و $C_B < r$

ج) بافری به طول ۱۰ بلوک و $C_B < \frac{r}{5}$

د) بافری به طول شیار و $C_B < b'_{tt}$

ه) بافری به طول شیار و $C_B < b'_{tt}$ برای زمان خواندن ترتیبی فایل

الف) بلاک اول در زمان b_{tt} خوانده می شود و بعد از آن در بافر شروع به پردازش می شود. در این حال هد از روی بلوک ۲ رد می شود لذا رای خواندن بلال ۲ باید $2r$ زمان صبر کنیم و بعد از یک دور کامل می توان بلاک یک که اصلاح شده را بر سر جای خود (هد بعد از یک دور کامل بر سر بلاک یک می رسد) نوشت و بلافارسله بعد از آن بلاک ۲ به درون بافر خوانده می شود. لذا می توان با شرایط یک بلاک زمان خواندن، پردازش و نوشتن هر بلاک $b_{tt} + 2r$ می باشد لذا زمان اصلاح کل فایل

$$\text{زمان اصلاح کل فایل} = b * (b_{tt} + 2r) = 1000 * (0.8 + 2 * 8.3) = 174000ms = 174s$$

می باشد شرط $C_B < 2r$ به این خاطر داده شده است تا مطمئن شویم که پس از خواندن یک بلاک هنگامی که دیسک یک دور می زند پردازش بلوک خوانده شده تمام گردیده و می توان آن را بر روی دیسک نوشت.

ب) با استفاده از بافرینگ مضاعف اصلاح هر دو بلاک پشت سر هم به زمان $2b'_{tt} + 2r$ نیاز می باشد . لذا زمان اصلاح کل $= (2b'_{tt} + 2r) * \frac{b}{2} = (2 * 0.84 + 2 * 8.3) * \frac{1000}{2} \approx 91s$ فایل زمان اصلاح کل فایل

می باشد $C_B < r$ یا همان $2C_B < 2r$ برای آن داده شده تا مطمئن باشیم که بعد از یک دور چرخیدن دیسک دو بلاک خوانده شده پردازش شده اند .

ج) چون بافر دارای ۱۰ بلاک طول می باشد . لذا اصلاح هر ۱۰ بلاک پشت سر هم به اصلاح کل فایل $10 * b'_{tt} + 2r$ زمان نیاز دارد لذا زمان

$$= (10 * b'_{tt} + 2r) * \frac{b}{10} = (10 * 0.84 + 2 * 2.3) * \frac{10000}{10} = 25 \text{ ثانیه}$$

شرط $C_B < \frac{r}{5}$ به این خاطر داده شده که در یک دور زدن دیسک باید ۱۰ بلاک تمام شده باشد نتیجه می

$$C_B < \frac{r}{5}$$

د) طبق فرض مسئله تعداد بلاک ها در هر شیار مساوی ۲۰ است لذا می توان گفت اندازه بافر برابر با ۲۰ بلاک می باشد (به طول شیار) لذا زمان اصلاح کل فایل برابر است با

$$= (20 * b'_{tt} + 2r) * \frac{b}{20} = (20 * 0.84 + 2 * 8.3) * \frac{10000}{20} = 16.7s$$

$$= b * b'_{tt} = 1000 * 0.84 = 8.4s$$

نتیجه می گیریم که با استفاده از بافری به طول شیار زمان اصلاح کل فایل تقریبا ۲ برابر زمان خواندن ترتیبی فایل می باشد .

فایل یا pile :

در بررسی پایل فرض می شود که طول هر باکت برابر یک بلاک می باشد فایل های پایل یا پراکنده بر خلاف فایل های ترتیبی ممکن است بعضی از فیلدهای آن مقدار نداشته باشند . (Null)

Sn	Family	name
1321	Alizade	Ali
1777	Ahmadi	Ahmad
1821	Hosseini	Hossein
1515	Asadi	Ali

فایل متراکم

Sn	Family	name
	Alireza	Ali
1777		Ahmad
	Hosseini	Hossein
1515	Asadi	Ali

فایل پایل (پراکنده)

این فایل را به دو دسته تقسیم می کنیم :

۱- متراکم (Dense)

۲- غیر متراکم (پراکنده Sparse)

همچنین دستیابی به این فایل ها به دوسته تقسیم کرده ایم :

۱- ترتیبی

۲- تصادفی

افروزگی (Redundance) :

اگر داده بیش از یک جا در بانگ اطلاعاتی ذخیره گردند افروزگی به وجود می آید . مثلاً علی کریمی در تهران ساکن است در چندین جای بانگ وجود داشته باشد . لذا در موقع بهنگام سازی بایستی در تمامی جاهای که این اطلاعات تکرار شده باشند باشند بهنگام سازی منتشر شود (propagating update) در غیر این صورت ناسازگاری (Inconsistent) رخ می دهد .

فسرده سازی فایل های اطلاعاتی :

در برخی موارد ممکن است بخواهیم فایل های اطلاعاتی را به صورت فشرده بر روی دیسک ذخیره کیم که از دلایل مهم آن می توان انتقال فایل در شبکه ، کم بودن ظرفیت دیسک و یا بالا بودن افروزگی فایل . باید توجه داشت که فشرده سازی عملیات را کاهش می دهد برای فشرده سازی نرم افزارها و روش های مختلفی وجود دارد از جمله روش Huffman در ساختمان داده ها و یا یک روش دیگر آن نمایش تعداد کاراکترهای تکراری پشت سرهم با یک عدد می باشد . مثلاً داده AAAABBCBBBF را فشرده کنید .

4A2BC3BF فشرده شده .

* برای نمایش بهتر فایل اطلاعاتی با فیلد های چند مقداری از روش ماتریس بیتی استفاده می کنیم .

نام استاد	نام درس	سخت افزار	ذخیره	ساختمان	برنامه سازی	اسمبلی	سیستم	زبان فنی	مبانی
اکبری	۱	۱	۰	۰	۰	۰	۰	۰	۰
شیخ زاده	۰	۰	۱	۱	۰	۰	۰	۰	۰
مرادی	۰	۰	۰	۰	۱	۱	۱	۱	۰
اسماعیل زاده	۰	۱	۰	۱	۰	۰	۰	۰	۱

انواع افزونگی به دو صورت می باشد :

- ۱- طبیعی
- ۲- تکنیکی

در مثال فوق افزونگی از نوع طبیعی می باشد چرا که در تعداد از نمونه رکورد بعضی از فیلدهای تکراری هستند افزونگی تکنیکی جهت استراتژی های خاص مثلاً کلید خارجی یا اندیس گذاری استفاده می شود .

پارامترهای ارزیابی ساختار فایل :

۱. Tf : زمان واکشی یک رکورد را گویند . یعنی پیدا کردن و خواندن رکورد .
۲. TN (Next) : زمان واکشی رکورد بعدی بر اساس ترتیب معین
۳. TZ (Insert) : زمان درج یک رکورد جدید 4 (update) Tu زمان به روز رسانی یک رکورد
۴. TD (Delete) : زمان حذف یک رکورد TX (exhau) زمان خواندن کل فایل
۵. Ty : زمان سازماندهی مجدد فایل

زمان دوباره نویسی یک بلاک (TRW) :

در حالت کلی زمان نوشتن یک بلاک برابر با زمان خواندن آن بلاک می باشد لذا نوشتن یک بلاک به صورت تصادفی

$$b'_{tt} = \frac{B}{t'} \quad \text{زمان می برد و نوشت بلک به صورت انبوه}$$
$$\frac{R}{t'} \quad \text{صورت رکورد به رکورد باشد . به صورت}$$

در نظر گرفت . چنانچه زمان پردازش یک بلوک کمتر از زمان $C_B < 2r - b_{tt}$ باشد . در آن صورت می توانیم بلاکی را که خوانده ایم در همان دور پس از اصلاح در سر جای اولیه اش ذخیره کنیم . لذا می توان نوشت زمان بازنویسی یک بلاک یا رکورد برابر با زمانی است که طول می کشد هد به اول بلاک برسد .

زمان انتقال بلاک از بافر به دیسک+زمانیکه طول می کشد به اول بلاک برسد = TRW

$$T_{RW} = 2r - b_{tt} = 2r : C_B < 2r - b_{tt}$$

با شرط

در صورتی که $C_B < 2r - b_{tt}$ باشد در آن صورت $TRW = 4r$ یعنی بایستی دو دور چرخش صورت گیرد .

مشخصات فایل های پایل (پرآکنده) :

فایل پایل ساده ترین ساختار را داشته و رکوردهای آن بر اساس فیلدهای مرتب نیست . طول رکوردها متغیر بوده ، تعداد فیلدها و مکان آنها در رکورد و نمونه های مختلف ممکن است نتفاوت باشد که در این صورت مقدار هر فیلد در مقابل آن نوشته شده است .