

**** سیستم کامپیوتری دارای دو محیط می باشد:**

۱- برون ماشین که شامل تمام وسایل جانبی I/O و ذخیره سازی داده ها می باشد.

۲- درون ماشین که شامل CPU-RAM-ROM می شود.

**** خصوصیات مشترک حافظه ها**

۱- نوشتن و خواندن ۲- نشانی پذیری (address ability) ۳- دستیابی پذیری (access ability)

۴- ظرفیت (capacity) ۵- زمان دستیابی (access time) ۶- نرخ انتقال (transfer rate) یا سرعت انتقال که معمولاً بر حسب بایت بر ثانیه محاسبه می شود.

**** محلی از حافظه که داده ها ذخیره شده اند و آدرس دهی جهت خواندن یا نوشتن آن انجام می گیرد را در حافظه اصلی کلمه (word) و در حافظه جانبی سکتور یا بلاک (sector /block) می باشد.**

**** سلسله مراتب حافظه (storage hierarchy) :**

نوار مغناطیسی → دیسک نوری → دیسک مغناطیسی → حافظه فلاش → حافظه اصلی → حافظه نهان → ثبات

**** در مقابل حسن افزایش ظرفیت حافظه های جانبی ، حسن حافظه های درون ماشین افزایش سرعت دستیابی می باشد البته از معایب آن در مقابل حافظه های برون ماشین افزایش هزینه یک بایت می باشد (cost per byte (cpb) می باشد.**

**** رده بندی حافظه ها (از نظر سرعت دستیابی) :**

رده اول : حافظه درون ماشین . **رده دوم :** دیسک مغناطیسی . **رده سوم:** دیسکهای نوری و نوار مغناطیسی. **رده چهارم:** کارت و نوار منگنه .

**** نواع حافظه برون ماشین از نظر تکنولوژی ساخت:**

تکنولوژی الکترومکانیک : کارت منگنه شدنی و نوار منگنه شدنی

تکنولوژی الکترومغناطیس : نوار مغناطیسی و دیسک مغناطیسی

تکنولوژی الکترواپتیک : دیسک نوری

تکنولوژی الکترومغنااپتیک : دیسک نوری مغناطیسی ((magnetic – optic (mo)

**** دیسکهای نوری مغناطیسی :** سرعت انتقال آنها یک مگا بایت در ثانیه می باشد. نوع DVD آن ۴/۷

تا ۱۷ گیگا بایت ظرفیت دارد. جنس آن تریبیوم یا گادونیوم و آهن و کبالت می باشد. این نوع دیسک اشکال دیسکهای مغناطیسی و نوری را ندارد. خواندن داده ها به صورت مغناطیسی می باشد. جهت نوشتن اشعه لیزر می تابد تا نقطه گرم شود و بعد به صورت مغناطیسی صفر یا یک نوشته می شود و بعد از سرد شدن با میدان مغناطیسی قابل تغییر نمی باشد.

**** دیسکهای باتغییر فاز (phase change (p-c)) از موادی پوشیده شده که با تابش اشعه لیزر قوی**

ذرات آن تغییر جهت می دهند و به این وسیله داده ها ذخیره می شوند.

**** دیسکهای دای پلیمر (dye-polymer (d-p))** از دولایه پلیمر ساخته شده که باتابش لیزر قوی لایه زیرین آن گرم شده ولایه بالاتری متورم می شود و به این وسیله داده ها ذخیره می شوند.

**** طبقه مغناطیسی :** استوانه ای است که پوسته بیرونی آن از مواد مغناطیسی شونده پوشیده شده و داده ها روی آن ذخیره می شوند. شیارها به صورت حلقه هایی موازی دورتادور استوانه فرض می شود که بین آنها گپ وجود دارد. با چرخش استوانه بلاکها از زیر هد عبور می کنند. اگر هد ثابت باشد. هر شیار یک هد خواهد داشت در نتیجه زمان استوانه جویی صفر خواهد بود که این نوع حافظه به خاطر سرعت بالاتر نسبت به نوار و دیسک مغناطیسی در گذشته به عنوان حافظه اصلی استفاده می شده. با هد متحرک آن نیز وجود دارد.

جهت ذخیره برنامه های ثابت که استفاده زیاد داشتند و ذخیره فایل های موقت بسیار فعال استفاده می شده. شیارها از بالا به پایین به ترتیب شماره گذاری می شود.

**** نوار مغناطیسی :**

روش دستیابی : ترتیبی (sequential)

تکنولوژی ساخت : ریل به ریل - نوار کارت ریج - نوار کاست - نوار صوتی تطبیق داده شده با کامپیوتر

نحوه ذخیره اطلاعات : داده ها روی شیارها که به صورت خطوط موازی با فاصله (gap) در طول نوار کشیده شده ذخیره می گردند.

تعداد شیارها یا ۷ عدد و یا ۹ عدد می باشد. بایتها یا کاراکترها در ۷ شیاره به صورت طولی و در ۹ شیاره به صورت عرضی ذخیره می گردند. ترتیب شماره بیتها در ۹ شیاره از بالا به پایین به ترتیب عبارتند از ۰-۱-۲-۳-۴-۵-۶-۷-۸-۹ و در ۷ شیاره از چپ به راست به ترتیب عبارتند از: ۰-۱-۲-۳-۴-۵-۶-۷

بیت توازن یا پاریتی بیت به دو صورت وجود دارد ۱- بیت توازن عرضی که برای هر کاراکتر محاسبه می گردد. ۲- بیت توازن طولی که برای هر بلاک محاسبه می گردد.

**** چگالی به صورت فیزیکی بیت** بر اینچ می باشد اما چون در عمل کوچکترین واحد ذخیره وبازیابی بیت است به بیت بر اینچ مطرح می شود.

GAP: به دو صورت می تواند باشد که با بلاک بندی فقط نوع دوم وجود دارد

۱- فضای خالی بین رکوردها ((inter record gap (irg))

۲- فضای خالی بین بلاکها ((inter block gap (ibg))

**** فضای خالی بین بلاکها را با G یا GAP یا IBG نشان می دهند و برای رسیدن به سرعت مطلوب خواندن و نوشتن می باشد. طول گپ پارامتری است فیزیکی و با تکنیکهای نرم افزاری قابل تغییر نمی باشد ولی تعداد آنها پارامتری منطقی است که با تکنیکهای نرم افزاری تغییر می کند.**

**** سرعت نوار باید مقدار ثابت Vs یا سرعت حس باشد تا بتوان داده ها را خواند یا نوشت .**

**** نیمی از مسافت گپ برای توقف نوار و رسیدن از سرعت حس به سرعت صفر و نیمی دیگر برای رسیدن از سرعت صفر به سرعت حس می باشد. که هر نیمه را با X نشان می دهند.**

** اگر زمان تلف شده هنگام توقف نوار t_o باشد در نتیجه خواهیم داشت $t_o = \frac{G}{v_s} = \frac{IBG}{V_s}$

و همچنین زمان تلف شده هنگام حرکت نوار برابر با t_o خواهد بود.

در نتیجه زمان عبور گپ از زیر هد برابر با $t_o \times 2$ خواهد بود و سرعت در این منطقه تقریباً به طور متوسط

$$\frac{V_s}{2} \text{ می باشد در نتیجه خواهیم داشت : } t_o = \frac{G}{v_s} = \frac{IBG}{V_s} \text{ یا } v_s = \frac{2x}{t_o} \text{ یا } v_s = \frac{G}{t_o}$$

** در قسمت گپ به این علت که سرعت نوار سرعت حس نیست نمی توان داده ای ذخیره کرد.

** نیمی از زمان تلف شده در هنگام عبور گپ برای توقف ونیمی برای حرکت صرف می شود.

** در نوار مغناطیسی فایلها بلاک بندی شده و در قالب رکورد ذخیره می شوند. ابتدای هر فایل با علامت

شروع یا (BOF) Begin Of File و انتهای هر فایل با علامت پایان یا (EOF) End Of File مشخص می شود.

** ظرفیت اسمی نوار در واقع حداکثر ظرفیت نوار است که می توان داده روی آن ذخیره کرد که از طرف

کارخانه سازنده اعلام می شود.

$$S_N = L \times D$$

L : طول نوار به اینچ D : چگالی نوار به بایت بر اینچ S_N : ظرفیت اسمی نوار

** از نوار مغناطیسی جهت تهیه نسخه پشتیبان یا ذخیره فایل ثبت تراکنشها استفاده می شود.

** هر بلاک بایک گپ ذخیره می شود در نتیجه ظرفیت واقعی یا همان فضای مفید استفاده شده کمتر از

ظرفیت اسمی یا کل فضا خواهد بود. که بین ظرفیت واقعی و اسمی رابطه زیر برقرار می باشد:

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

B طول بلاک و G طول گپ بوده و هر دو به یک واحد باید باشند.

** درصد استفاده مفید از نوار نیز توسط رابطه زیر محاسبه می شود:

$$\text{درصد استفاده مفید از نوار} = \frac{B}{B+G} \times 100$$

** در محاسبه درصد استفاده مفید از نوار اگر از رابطه زیر استفاده کنیم در واقع درصد استفاده مفید به طور

دقیق محاسبه شده است : که در آن $W'B$ فضای هدر رفته درون بلاک می باشد.

$$\text{درصد استفاده مفید از نوار به طور دقیق} = \frac{B - W'_B}{B + G} \times 100$$

** زمان انتقال واقعی داده ها در نوار مغناطیسی :

داده ها به دوروش خوانده می شوند

- ۱- **روش بلاکی (block mode)** در این روش بلاکها یکی یکی خوانده و پردازش می شوند در نتیجه به تعداد خواندن بلاکها حرکت و توقف خواهیم داشت.
 T : زمان انتقال N بلاک :

$$T = N \left(\frac{B}{t'} + \frac{G}{t'} + \tau \right)$$

t' : نرخ انتقال اسمی τ : زمان حرکت -توقف درایورنوار N :تعدادبلاکها

- ۲- **روش جریانی (stream mode)** در این روش تعداد N بلاک باهم خوانده می شود مانند کپی کردن داده ها

$$T = N \left(\frac{B}{t'} + \frac{G}{t'} \right) + \tau$$

- مثال: فایلی با ۱۰۰۰۰ رکورد ۸۰ بایتی روی نوار مغناطیسی با چگالی ۱۶۰۰ بایت بر اینچ ذخیره می شود.
بدون در نظر گرفتن بحث بلاکها در این حالت چه طولی از نوار اشغال می شود؟..... اگر در هر بلاک یک رکورد قرار دهیم و طول گپ ۵/۰ اینچ باشد چه مقدار نوار برای گپها مصرف می شود؟.....
در این حالت طول کل فایل بعد از ذخیره چقدر خواهد بود؟.....
درصد استفاده واقعی از نوار در این حالت چقدر خواهد شد؟
اگر $\tau = 0.2ms$ و نرخ انتقال ۱۰۰۰ بایت بر میلی ثانیه باشد مقدار زمان مورد نیاز جهت خواندن فایل به صورت بلاک به بلا کوبه صورت جریانی چقدر خواهد بود؟..... و
حالت دیگر اینکه در هر بلاک ۱۰۰ رکورد ذخیره کنیم ، موارد قبل را محاسبه کنید.

دیسک مغناطیسی :

** نوع دسترسی : مستقیم (direct access device=DASD)

** بیرونی ترین شیار شماره صفر دارد.

** نواحی روی دیسک از بیرون به سمت مرکز دیسک به ترتیب عبارتند از:

- ۱- ناحیه فرود نوک (landing zone) ۲- ناحیه شروع حرکت (tack-off zone) ۳- ناحیه احتیاطی بیرونی (outer guard band) ۴- ناحیه شیارهای ضبط داده ها ۵- ناحیه احتیاطی درونی ۶ - ناحیه احتیاطی نهایی
** شیارها از بیرونی ترین شیار تا درونی ترین شیار از صفر تا n شماره گذاری می شوند.

****** دیسکهای چند صفحه ای را pack می نامند و یک pack با n صفحه دارای $2n-2$ رویه جهت ذخیره داده هاست.

نکته : در دیسک سخت هد مقدار کم در حد میکرومتر از سطح دیسک فاصله دارد اما در فلاپی دیسکها هد با سطح دیسک تماس دارد.

دیسکها از نظر جابجا شدن به دونوع تقسیم می شوند : ثابت مانند دیسک سخت و جابجا شدنی مانند فلاپی دیسک .

هر صفحه دو رویه دارد و در هر رویه یک یا دو لایه جهت ذخیره داده ها می توان استفاده کرد. به تعداد رویه هد وجود دارد.

دیسکهای چند صفحه ایا (multiplatter) را pack گویند.

یک پک با n صفحه دارای $2n$ رویه است که معمولاً از $2n-2$ رویه آن استفاده می شود.

**** تقسیمات دیسک:**

دوایر متحدالمرکزی به نام شیار محل ذخیره داده ها می باشد، که با فاصله ای به نام Gap روی هر رویه وجود دارد. که به آن گپ بین شیار می گویند. می شود و قبل یا بعد از ذخیره داده ها وجود دارد و با تکنیکهای نرم افزاری نمی توان آن را تغییر داد.

شیارهای هم شماره از همه رویه ها را سیلندر یا استوانه گویند.

این دایره ها به قطعات مساوی به نام سکتور تقسیم می شوند. به آنها سکتور سخت افزاری نیز گویند این سکتورها در فرمت سطح پایین توسط کارخانه سازنده ایجاد می شود.

بین سکتورها گپ بین سکتوری در نظر گرفته می شود که برای تشخیص مرز سکتورها می باشد و برای عملیات سخت افزاری استفاده می شود. (در عملیات بلاکینگ این گپ نادیده گرفته می شود)

سکتور نرم افزاری یا فرمت نرم افزاری توسط سیستم عامل ایجاد می شود و همان بلاکها می باشد.

آغاز بلاکها همیشه آغاز سکتور می باشد در نتیجه طول بلاکها مضرب صحیحی از طول سکتور خواهد بود.

طول بلاک می توان از یک سکتور کمتر یا یک سکتور یا چند سکتور باشد.

اگر چگالی شیارهای بیرونی از چگالی شیارهای درونی کمتر باشد آنگاه تعداد سکتورهای شیارهای بیرونی و درونی باهم مساوی می باشد و برعکس.

چگالی دیسک سخت بر حسب بایت بر اینچ مربع بیان می شود.

دیسک را ناحیه (zone) بندی می کنند. یعنی تعدادی شیار نزدیک به هم که اختلاف طول کمتری

دارند را یک ناحیه فرض گرفته و حداکثر تعداد سکتوری که امکان پذیر است را روی آنها درست می کنند. به

این ترتیب : چگالی شیارهای بیرونی و درونی مساوی می شود در نتیجه تعداد سکتورهای شیارهای بیرونی بیشتر از شیارهای درونی می شود؛ و برعکس .

بین شیارها فضای هدررفته گپ وجود دارد که قبل از ذخیره داده ها نیز وجود دارد .

هر بلاک بایک گپ ذخیره می شود که معمولاً جزء خود بلاک می باشد.

شیارهای هم شماره از تمام رویه ها راسیلندریا استوانه می نامند.

** پارامترهای زمانی :

** زمان استوانه جویی یا زمان پیگرد یا **seek time** : با حرف s نشان داده می شود و از لحظه ای کا

فرمان خواندن یا نوشتن صادر می شود تا زمانیکه هدروی سیلندر یا شیار موردنظر قرار می گیرد، می باشد.

** سرعت گردش دیسک : تعداد دور در دقیقه می باشد (rotation per minute (RPM)) که توسط

کارخانه سازنده اعلام می گردد. که از روی آن مدت زمان یک دور کامل را برحسب میلی ثانیه حساب کرده با $2\pi r$ نشان می دهند.

** پس این پارامتر فیزیکی و غیرقابل تغییر با تکنیکهای نرم افزاری می باشد.

$$2\pi r = \frac{60000}{\text{RPM}} \text{ms}$$

** زمان انتظار دوران (درنگ دوران) **rotational latency** : از لحظه قرار گرفتن هد روی شیار موردنظر تا

لحظه ای که آغاز بلاک موردنظر زیرهد قرار گیرد می باشد که با حرف r نشان می دهند.

مقدار این زمان بین صفر تا زمان یک دور کامل می باشد پس r مقدار متوسط $2\pi r$ می باشد.

$$0 \leq r \leq 2\pi r$$

** زمان بختانه یا زمان دسترسی تصادفی (random access time (RAT)) :

از لحظه صدور دستور خواندن یا نوشتن تا زمانیکه هد در آغاز بلاک موردنظر قرار گیرد می باشد.

$$\text{RAT} = s + r$$

** زمان استقرار (setting time) : پس از توقف هد مدت زمانی طول می کشد تا لرزش بازوی هد گرفته

شود. که معمولاً در محاسبات به همراه زمان استوانه جویی محاسبه می گردد.

** نرخ انتقال : تعداد بایت که در واحد زمان جابجا می شود می باشد با واحد بایت بر میلی ثانیه محاسبه

می شود و به دو صورت اسمی و واقعی می باشد. که نرخ انتقال اسمی توسط کارخانه سازنده اعلام می شود و پارامتری فیزیکی می باشد.

اما نرخ انتقال واقعی که کسری از نرخ انتقال اسمی می باشد پارامتری منطقی می باشد.

$$t' = k \times t \text{ اسمی واقعی}$$

که ضریب k نسبت داده ذخیره شده به فضای اختصاص داده شده می باشد.

*** در دیسکها ،بلاکها دوقسمت (بخش پیشوندی و بخش داده ای) دارا می باشند که IBG جزء بخش پیشوندی می باشد. یعنی یک بلاک = B بات داده ها + C بایت پیشوند می باشد.

*** فاکتور شیاربندی (TF tracking factor) : عبارت است از تعداد بلاکهایی که در یک شیار جا می شود.

خارج قسمت تقسیم صحیح طول شیار به بایت بر طول بلاک (بلاک+گپ (C+B)) را فاکتور شیاربندی و باقیمانده آن را که فضای هدررفته روی شیار به ازاء جانشدن بلاک در شیار می باشدرا با W3 نشان می دهند.

$$TF = \left\lfloor \frac{C_{NT}}{B+C} \right\rfloor$$

C_{NT} : ظرفیت اسمی شیار می باشد. B : طول بلاک می باشد. C: طول بخش غیرداده ای بلاک می باشد که شامل بلاک نیز می باشد.

*** درصد استفاده مفید از دیسک : حاصل تقسیم داده ذخیره شده به فضای اختصاص داده شده می باشد. که در صد ضرب شود.

*** در این فرمول داده ذخیره شده همان بلاکهای روی یک شیار است و فضای اختصاص داده شده طول شیار:

$$C_{ET} = \frac{T_F \times B}{C_{NT}} \times 100$$

مقدار B در فرمول فوق فقط بخش داده ای بلاک می باشد.

اگر بخواهیم درصد استفاده مفید به طوردقیق را پیدا کنیم باید به جای داده ذخیره شده ، داده مفیدذخیره شده را بنویسیم

$$C_{ET} = \frac{T_F \times (B - W'_B)}{C_{NT}} \times 100$$

مقدار W'_B در فرمول فوق : فضای هدر رفته درون بلاک می باشد.

*** می توان درصد استفاده مفید از دیسک را باتقسیم داده مفید ذخیره شده در بلاک به فضای اختصاص داده شده به بلاک محاسبه کرد . L_S : طول سکتور به بایت N : تعداد سکتور اختصاص داده شده به بلاک

$$E = \frac{R \times B_F}{N \times L_S} \times 100$$

$$BF: \text{تعداد رکورد در هر بلاک} \quad R: \text{طول رکورد} \quad N = \left\lfloor \frac{B}{L_S} \right\rfloor$$

*** عوامل موثر در طول بلاک ۱- ظرفیت بافر ۲- در محیط چندبرنامه ای که برنامه هاز یک فایل استفاده می کنند ، طول بلاک با طول کوچکترین بافر در برنامه ها برابر می شود. ۳- نوع سیستم عامل ۴- نحوه پردازش فایل (اگر رکورد به رکورد یا تصادفی پردازش می شود بهتر است طول بلاک کوچک باشد) نکته : اگر زمان خواندن یک بلاک را $s + r + b_{tt}$ بدانیم . دو حالت پردازش ممکن است رخ دهد:

۱- پردازش رکوردی : هر بار مراجعه به دیسک برای پردازش یک رکورد از بلاک انجام می گیرد. در نتیجه اگر فایل دارای n رکورد باشد . زمان پردازش فایل می شود:

$$Tp \text{ file}(R) = n(s + r + b_{tt}) + n.C_R$$

۲- پردازش بلاکی: در این حالت هربار مراجعه به دیسک یک بلاک خوانده و پردازش می شود در نتیجه خواهیم داشت :

$$Tp \text{ file}(B) = b(s + r + b_{tt}) + b.C_B$$

**** نرخ انتقال :**

دستیابی مستقیم به بلاک در واقع مجموع زمانهایی است که طول می کشد تا هد به آغاز بلاک برسد و زمانیکه طول می کشد تا بلاک از زیرهد بگذرد ، که به آن b_{tt} گفته می شود.

پس زمان خواندن یا نوشتن یا زمان دستیابی مستقیم به بلاک از این رابطه به دست می آید: $S + r + b_{tt}$ در نتیجه نرخ انتقال که تقسیم داده جابجا شده به مدت زمان جابجایی است از رابطه روبرو محاسبه می شود:

$$t = \frac{B}{s + r + b_{tt}}$$

$$t = \frac{B - W_B}{s + r + b_{tt}} \quad \text{اگر بخواهیم نرخ انتقال واقعی به صورت دقیق حساب کنیم :}$$

می توان گفت اگر به جای B در صورت کسر ، $BF \times R$ بگذاریم در واقع نرخ انتقال مفید را محاسبه کرده ایم.

$$\text{می توان رابطه زیر را برای نرخ انتقال اسمی و واقعی را نوشت:} \quad t_{\text{اسمی}} = \frac{B}{L_S \times N} \quad t_{\text{واقعی}}$$

****** از طریق نرخ انتقال می توان زمان خواندن فایل در صورتیکه به طور تصادفی رکوردها خوانده شوند را به

$$\text{صورت مقابل محاسبه کرد :} \quad TRR = \frac{b \times B(s + r + b_{tt})}{B - WB}$$

$$TRR = b(s + r + b_{tt}) \quad \text{و در صورتیکه از WB صرف نظر شود خواهیم داشت :}$$

بلاکینگ :

****** رکوردها ابتدا در یک بلاک قرارداد می شوند و این بلاکها هستند که ذخیره می شوند. یعنی واحد انتقال داده ها از حافظه اصلی (بافر) به حافظه جانبی و برعکس بلاک می باشد.

****** مزایای بلاک بندی : ۱- کاهش زمان I/O چون در هر بار انتقال ، تعدادی رکورد جابجا می شود.

۲- کاهش فضای هدررفته بین داده ای ؛ چون با بلاک بندی بین هر بلاک یک گپ اما بدون بلاک بندی که رکوردها باید ذخیره شوند، برای هر رکورد یک گپ باید باشد.

**** معایب بلاک بندی :** ۱- نرم افزار پیچیده تر ۲- مصرف بیشتر حافظه اصلی ۳- بالا رفتن میزان خطا

؛ چون هرچه داده جابجا شده بیشتر باشد احتمال خطا بیشتر می شود.

****** طول رکوردها می تواند متغیر یا ثابت باشد. در حالت متغیر در هر رکورد یک اشاره گر که مقدار عددی طول رکورد را دارد قرار می دهند.

**** از دو تکنیک بلاک بندی استفاده می شود:** ۱- بلاک بندی یک پاره: در این حالت اگر رکورد در بلاک جا نشود، فضای آن در بلاک، خالی و هدر رفته باقی می ماند. ۲- بلاک بندی دو پاره: در این حالت اگر رکورد در بلاک جانشود، آن را دو پاره می کنند، یک قسمت را در بلاک جاری و قسمت دیگر را در بلاکی دیگر قرار می دهند، و در بلاک جاری یک اشاره گر قرار می دهند که به آدرس بلاکی که پاره دوم رکورد در آن قرار داده شده اشاره می کند.

**** با توجه به موارد فوق چهار حالت برای بلاک بندی می توان داشت که در هر حالت باید فضای هدر رفته به ازاء هر بلاک و هر رکورد و همچنین روش محاسبه BF را پیدا کرد.**

**** نام گذاری پارامترها:**

W1: همان IBG می باشد.

W2: فضای هدر رفته درون بلاک به ازاء جانشدن رکورد در بلاک می باشد. اگر طول رکورد متغیر باشد آنگاه $W2 = \bar{R}$ می باشد.

اگر طول رکورد ثابت باشد به طور دقیق می توان آن را محاسبه کرد: $W2 = B - B_F \times R$

اگر نوع بلاکینگ دو پاره باشد آنگاه: $W2 = 0$

W3: فضای هدر رفته روی شیار به ازاء جانشدن بلاک در شیار می باشد.

WB: فضای هدر رفته به ازاء هر بلاک می باشد.

WR: فضای هدر رفته به ازاء هر رکورد می باشد.

۱- بلاکینگ یک پاره با رکورد ثابت:

$$WB = W1 + W2 + \frac{W3}{TF}$$

$$WR = \frac{WB}{BF}$$

$$BF = \left\lfloor \frac{B}{R} \right\rfloor$$

مقدار W2 را می توان نوشت: $0 \leq W2 \leq R - 1$

۲- بلاکینگ یک پاره با رکورد متغیر:

$$WB = W1 + W2 + BF \times P + \frac{W3}{TF}$$

$$WR = \frac{WB}{BF}$$

$$BF = \left\lfloor \frac{B}{R + P} \right\rfloor$$

در محاسبه BF می توان مقدار W2 را که در اینجا با W4 نیز نشان می دهند و برابر با متوسط طول رکورد

است را از B کم کرد و آنگاه: $BF = \frac{B - W4}{R + P}$

۳- بلاکینگ دو پاره با رکورد ثابت:

$$WB = W1 + P + \frac{W3}{TF}$$

$$WR = \frac{WB}{BF}$$

$$BF = \frac{B - P}{R}$$

(گرچه این روش غیر عملی می باشد، اما از نظر تئوری قابل بررسی می باشد)

۴- بلاکینگ دوپاره با رکورد متغیر:

$$WB = W1 + W2 + P + BF \times P + \frac{W3}{TF}$$

$$WR = \frac{WB}{BF}$$

$$BF = \frac{B - P}{R + P}$$

*** اگر بلاکینگ دوپاره باشد در محاسبات باید از ظرفیت B به مقدار یک اشاره گر P کم کرد. و اگر طول رکورد متغیر باشد به طول رکورد R به مقدار یک اشاره گر P اضافه کرد.

*** هنگامیکه طول رکورد متغیر باشد در محاسبات متوسط طول رکورد قراردادده می شد. معمولاً در مسائل متوسط طول داده می شود مگر اینکه ماگزیمم طول داده شود که باید بر ۲ تقسیم کرده تا متوسط محاسبه شود.

*** در صورتیکه طول رکورد متغیر باشد: جهت ذخیره رکورد از سه تکنیک استفاده می شود:

۱- درج نشان گر پایان رکورد

۲- درج طول در بخش غیر داده ای رکورد

۳- ایجاد جدول مکان نما (position table) که در این جدول آدرس رکورد نسبت به ابتدای بلاک ذخیره می گردد.

*** مهمترین عوامل موثر در نرخ واقعی انتقال داده ها در دیسک عبارتند از: ۱- نوع بافرینگ ۲- شیوه دستیابی بی به بلاک ۳- CB ۴- نحوه ذخیره بلاکها درون شیارها ۵- نحوه پردازش رکوردها ۶- نحوه دستیابی برنامه به رکورد مورد نظر.

بافرینگ :

*** بافر سخت افزاری: حافظه ای است سخت افزاری که در درایور دیسکها قرار دارد و جهت بهبودی و سرعت دادن به انتقال داده ها از آن استفاده می شود.

*** بافر نرم افزاری: در واقع بخشی از حافظه اصلی می باشد که به عنوان بافر جهت انتقال داده ها استفاده می شود.

*** به مجموعه بافرها Buffers Pool گفته می شود.

*** برنامه کاربر به دو روش می تواند به بافر دسترسی پیدا کند:

۱- روش انتقالی یا حالت حرت (move mode) : در این روش برنامه کاربردی در قسمتی از حافظه اجرا

می شود و در همان محل بافر خود را دارد، بلاک به بافر ورودی در ناحیه دیگری از حافظه اصلی منتقل شده و سپس به بافر برنامه کاربردی منتقل می گردد، چون برنامه به بافر ورودی و خروجی دسترسی ندارد، پس از اجرا به بافر خروجی در ناحیه دیگری از حافظه منتقل شده تا توسط فایل سیستم بلاکبندی شده و به حافظه جانبی منتقل شود.

۲- **روش مکان نمایی یا تعیین محل (locate mode)** : در این روش همه بافرها در یک محل از حافظه اصلی قرار دارند و پس از انتقال بلاک به بافر فقط آدرس آن در اختیار برنامه کاربر قرار می گیرد و برنامه کاربر در همان محل عملیات اجرا را انجام داده سپس بلاکبندی را نیز انجام می دهد، تا فایل سیستم آن را به حافظه جانبی منتقل کند. در این روش انتقال بلاک در حافظه اصلی انجام نمی شود.

**** بافر به اندازه حداقل یک رکورد یا یک بلاک ظرفیت دارد.**

**** بافر به سه روش ایجاد می گردد:**

۱- ناحیه ای از حافظه در برنامه تعریف می شود تا به عنوان بافر باشد (توسط برنامه نویس ایجاد می گردد)

۲- از سیستم عامل تقاضا می شود. (توسط دستور ماکروبی که در برنامه فراخوانی (call) می شود)

۳- سیستم عامل هنگام باز کردن فایل خودش بافر را ایجاد کرده و پس از بسته شدن فایل آنها را پس می گیرد.

**** انواع بافرینگ :** ۱- ساده یا یک بافری (single buffering)

۲- مضاعف (double buffering / buffer swapping)

۳- چند گانه Multiple Buffering یا چرخشی (circular buffering)

در این نوع بافرینگ سیستم از چند نوع بافر استفاده می کند که به هم لینک شده اند.

سیستم می تواند بافر ها را به ترتیب پردازش کند.

**** بافرینگ ساده یا تک بافری:** در این حالت CPU باید منتظر باشد تا بلاک به بافر منتقل شود، سپس پردازش خود را انجام دهد، در این مدت نیز بافر اشغال بوده و امکان انتقال وجود ندارد. در نتیجه در این روش بافر، امکان انجام عملیات پردازش و انتقال به طور همزمان وجود ندارد.

محاسبه نرخ انتقال :

با توجه به روش خواندن بلاکها ، به دوروش نرخ انتقال این نوع بافر محاسبه می شود.

۱- **بافرینگ ساده و مرتب خوانی :** در این حالت وقتی بلاک اول خوانده و پردازش شد حتماً باید

بلاک دوم خوانده شود نه هر بلاکی ، یعنی به ترتیب خوانده شوند. در صورتیکه شرط $CB \leq b_{tt}$

برقرار باشد آنگاه : در این صورت زمان b_{tt} صرف خواندن B_n می شود. پس از پردازش آن که

حداکثر b_{tt} زمان صرف می شود، هدر انتهای B_{n+1} خواهد بود، پس برای خواندن آن باید $2r - b_{tt}$

صبر کرد تا هد به آغاز B_{n+1} برسد و زمان b_{tt} صرف می شود. تا بلاک B_{n+1} خوانده شود در نتیجه

در عملیات خواندن بلاکها ، برای خواندن یک بلاک مدت زمان $b_{tt} + 2r - b_{tt} + b_{tt}$ صرف می شود.

$$t'(1) = \frac{B}{2r + b_{tt}} \quad \text{در نتیجه نرخ انتقال می شود:}$$

**** نیز می توان گفت :** هنگامیکه B_n خوانده شده و در بافر باشد. هد در ابتدای B_{n+1} خواهد بود . پردازش

B_n که آغاز گردد ، هد از ابتدای B_{n+1} می گذرد. در صورتیکه شرط $CB \leq 2r$ برقرار باشد. تقریباً مدت $2r$

باید صبر کرد تا ابتدای B_{n+1} به زیر هد برسد در این لحظه با صرف b_{tt} می توان B_{n+1} را خواند.

در نتیجه برای خواندن یک بلاک $2r+btt$ زمان صرف خواهد شد و خواهیم داشت : $t'(1) = \frac{B}{2r + btt}$

نکته : می توان نوشت $t'(1) = \frac{B - WB}{2r + btt}$

نکته : اگر $CB \leq 2r$ نباشد . مدت انتظار رادودور دیسک محاسبه می کنیم یعنی : $t'(1) = \frac{B}{4r + btt}$

۲- بافرینگ ساده و درهم خوانی : در این حالت پس از خواندن و پردازش یک بلاک ، نیاز نیست حتماً بلاک بعد از آن خوانده شود بلکه هر بلاکی که آغازش زیر هد باشد خوانده می شود. (مانند محاسبه میانگین معدل دانشجویان)

اگر $CB \leq b_{tt}$ باشد آنگاه بلاکها یک درمیان خوانده می شوند پس تقریباً در دور از دیسک همه بلاکها خوانده می شوند. پس جهت محاسبه نرخ انتقال اگر داده جابجا شده ، بلاکهای یک شیار باشد این انتقال در زمان دودور دیسک انجام شده است .

$$t'(1) = \frac{T_f \times B}{2 \times 2r} = \frac{T_f \cdot B}{4r} \quad \text{در نتیجه نرخ انتقال می شود:}$$

**** بافرینگ مضاعف :** در این حالت بافر جای دو بلاک دارد پس وقتی بلاک اول در بافر در حال پردازش است بلاک دوم می تواند به بافر منتقل شود که پس از انتقال بلاک دوم، پردازش بلاک اول تمام شده (با وجود شرط کارایی) پس می توان بلاک سوم را جای آن در بافر قرارداد. به این ترتیب پردازش و انتقال به طور همزمان می تواند انجام شود.

شرط کارایی در این روش چون به طور معمول هنگام خواندن بلاکها ، دو بلاک بدون فاصله زمانی خوانده می شود پس حداکثر زمان پردازش به مقدار زمان عبور گپ از زیر هد اضافه می شود.

$$CB \leq b_{tt} \Rightarrow CB \leq \frac{B+G}{t} \Rightarrow CB \leq \frac{B}{t} + \frac{G}{t} \Rightarrow CB \leq btt + \frac{G}{t} \quad \text{یعنی:}$$

t : نرخ انتقال اسمی می باشد.

$$CR \leq \frac{R + WR}{t} \quad \text{می توان بر حسب رکورد نیز نوشت}$$

در این روش نیز نرخ انتقال به دو صورت محاسبه می گردد:

۱- بافرینگ مضاعف با وجود شرط کارایی:

در این حالت می توان دید که در یک دور دیسک همه بلاکهای یک شیار خوانده و پردازش می شوند.

$$t' = \frac{T_f \times B}{2r} \quad \text{در نتیجه نرخ انتقال می شود:}$$

۲- بافرینگ مضاعف وعدم شرط کارایی :

در این حالت پس از انتقال دو بلاک با صرف زمان $2btt$ ، به مدت یک دور کامل باید صبر کرد تا ابتدای دوبلاک بعدی زیر هد برسد که در این مدت زمان پردازش دوبلاک قبلی تمام شده است. در نتیجه برای انتقال دوبلاک مدت زمان $2r+2btt$ صرف خواهد شد.

$$t'(2) = \frac{2B}{2r + 2btt} \Rightarrow t'(2) = \frac{B}{r + btt} \quad \text{در نتیجه نرخ انتقال می شود:}$$

در این صورت برای خواندن کل فایل یعنی b بلاک ، مدت زمان $b \times (r + btt)$ صرف می شود. مقایسه زمان انتقال حجم مشخصی از داده ها در محاسبات فوق در واقع اثر بافر در سرعت عملیات دیسک می باشد.

** سطوح آدرس دهی:

سه سطح آدرس دهی برای رکوردهای فایل در نظر گرفته می شود:

سطح برنامه کاربر : در این سطح نشانی دهی به صورت زیر می باشد:

۱- نشانی دهی محتوایی (مقداری) یعنی رکوردی را با مقدار فیلدی از آن مورد جستجو قرار داد.

۲- نشانی دهی نسبی ، یعنی همان شماره رکورد که کاربر در فایل برای هر رکورد قائل می شود، می باشد.

این شماره از ۱ شروع می شود، به آن آدرس نسبی رکورد یا

(Relative Record Address) (RRA) گویند.

۳- نشانی دهی نمادی ، هر رکورد با یک نام مشخص می شود مانند فایل.

برنامه کاربر با فایل سیستم منطقی ارتباط دارد که وظیفه آن باز وبسته کردن فایل های بانک اطلاعاتی می باشد. این قسمت از فایل سیستم ، فایل را به صورت آرایه ای از رکوردها می بیند که از شماره ۱ شماره گذاری شده است. و حافظه جانبی را به صورت آرایه ای از بلاکها می بیند که از صفر شماره گذاری شده اند. به این شماره ها آدرس نسبی بلاک (Relative Block Address) (RBA) گفته می شود.

برنامه کاربر تقاضای رکورد نام را از فایل سیستم منطقی می کند ، این شماره I توسط فایل سیستم منطقی به شماره RBA تبدیل می شود و فایل سیستم منطقی از فایل سیستم فیزیکی تقاضای بلاک با آدرس RBA می کند.

فایل سیستم فیزیکی که با حافظه جانبی ارتباط دارد و وظیفه جستجو و آدرس دهی بلاک و خواندن و نوشتن بلاک به روی حافظه جانبی را به عهده دارد. عدد RBA را به آدرس فیزیکی بلاک که همان شماره درایو ، شماره سیلندر در درایو ، شماره شیار یا رویه در سیلندر و شماره سکتور یا بلاک در شیار است تبدیل می کند. تا بتواند بلاک را بخواند.

** نشانی دهی در سطح منطقی و فیزیکی بلاک:

** با توجه به اینکه داده ها روی دیسک سخت ذخیره می شوند نرم افزار کاربردی از این داده ها استفاده می کند؛ بنابراین باید فایل سیستم که وظیفه مدیریت نقل و انتقال داده ها از حافظه جانبی به حافظه اصلی و برعکس را به عهده دارد؛ دو رویه داشته باشد؛ یک رویه که با نرم افزار کاربر ارتباط دارد را فایل سیستم منطقی (logical file system) و رویه دیگر که با رسانه فیزیکی ذخیره سازی ارتباط دارد را فایل سیستم فیزیکی (phisycal file system) می نامند.

** فایل سیستم منطقی :

این رویه از فایل سیستم با نرم افزار کاربر کار می کند. نرم افزار کاربر جهت انجام عملیات روی رکوردی از بانک اطلاعاتی آن را از فایل سیستم منطقی تقاضا می کند. با توجه به اینکه رکوردها در فایل از یک شماره گذاری می شوند، در نتیجه برنامه کاربر تقاضای رکورد i ام از فایل سیستم منطقی می کند. این شماره ها آدرس منطقی رکورد می باشد. پس می توان گفت فایل سیستم منطقی فایل را به صورت آرایه ای از رکوردها می بیند که از یک شماره گذاری شده است .

همچنین فایل سیستم منطقی ، رسانه فیزیکی ذخیره سازی را به صورت آرایه ای از بلاکها می بیند که از صفر شماره گذاری می شود. به این شماره ها آدرس نسبی بلاک یا آدرس نسبی رکورد گویند. یعنی اگر رسانه صفرم ۱۰۰ بلاک و رسانه یکم ۲۰ بلاک و رسانه دوم ۸۰ بلاک داشته باشد. آنها را از صفر تا ۱۹۹ شماره گذاری می کند، بنابراین درایو شماره صفر شامل بلاک با $RBA=0$ تا $RBA=99$ می شود و درایو اول از $RBA=100$ تا $RBA=119$ و درایو دوم $RBA=120$ تا $RBA=199$ که شماره RBA آغاز هر درایو را RBA_{bod} (begin of device) گویند.

تعداد بلاک در درایو را به S نشان می دهند و از رابطه روبرو بدست می آید. $S = c \times t \times b$ که در آن c تعداد سیلندر و t تعداد شیار در سیلندر و b تعداد بلاک در شیار می باشد.

** شماره بایتهای یک فایل را از صفر شماره گذاری می کنند. از رابطه زیر شماره بایت آغاز رکورد نسبت به ابتدای فایل مشخص می شود: R : طول رکورد است

$$\text{Byteoffset}_{\text{REC}} = (i-1) \times R$$

** در صورتیکه بلاکهای یک فایل را از صفر شماره گذاری کنند. از رابطه زیر شماره بلاکی که رکورد i ام در آن قرار دارد، نسبت به ابتدای فایل مشخص می شود:

$$rba_{\text{REC}} = \left\lfloor \frac{(i-1) \times R}{B} \right\rfloor$$

**** شماره بلاکی که رکورد موردنظر در آن قرار دارد، نسبت به ابتدای آرایه ای که فایل سیستم از همه بلاکهای درایوها ساخته است، از رابطه زیر مشخص می شود:**

$RBA_{BOF} = RBA_{REC} - rba_{REC}$: شماره بایت آغاز فایل نسبت به ابتدای فایل است
به RBA_{REC} بدست آمده در رابطه فوق آدرس نسبی بلاک مورد نظر گویند.

**** در صورتیکه چند رسانه فیزیکی موجود باشد، از آنجا که فایل سیستم منطقی همه بلاکها را پشت سرهم مانند یک آرایه فرض می کند؛ بنابراین شماره بلاک آغاز رسانه ممکن است صفر نباشد؛ پس جهت مشخص کردن بلاک مورد نظر نسبت به ابتدای دیسکی که بلاک در آن قرار دارد باید از رابطه زیر استفاده کرد:**
 RBA_{BOD} شماره بلاک آغاز درایو در آرایه بلاکها است

$$RBA = RBA_{REC} - RBA_{bod} = RBA_{BOF} + rba_{REC} - RBA_{bod}$$

یا

$$RBA = RBA_{BOF} + \left\lfloor \frac{(i-1) \times R}{B} \right\rfloor - RBA_{bod}$$

**** فایل سیستم فیزیکی :**

این رویه از فایل سیستم با رسانه فیزیکی ارتباط دارد. که عملیات مربوط به انتقال بلاک به / از حافظه جانبی را به عهده دارد، نوشتن بلاک، خواندن بلاک، آدرس دهی بلاک را به عهده دارد.
آدرس فیزیکی رکورد یا آدرس فیزیکی بلاک شامل چهار قسمت می شود که عبارتند از:
۱- شماره درایو ۲- شماره سیلندر در درایو ۳- شماره شیار در سیلندر ۴- شماره بلاک در شیار
**** جهت پیدا کردن شماره درایو: اگر شماره RBA بلاک مورد نظر از $S-1$ درایوی کمتری مساوی بود، بلاک در این درایو قرار دارد.**
**** جهت مشخص کردن شماره سیلندر یا استوانه :**

$$CYL\# = \left\lfloor \frac{RBA}{(t \times b)} \right\rfloor$$

**** جهت مشخص کردن شماره شیار در استوانه یا رویه در استوانه :**

$$trk\# = MOD \left[\left\lfloor \frac{RBA}{(t \times b)} \right\rfloor / b \right]$$

**** جهت مشخص کردن شماره بلاک در شیار :**

$$BLK\# = RBA \bmod b$$

در روابط فوق :

RBA از رابطه $RBA = RBA_{BOF} + rba_{REC} - RBA_{BOD}$ بدست می آید.
 t : تعداد شیار در هر سیلندر
 b : تعداد بلاک در هر شیار

** چگالی لود اولیه loading density

لود اولیه یعنی ابتدای پر شدن بانک اطلاعاتی توسط رکوردها. معمولاً فایلها به صورت ترتیبی بوده و رکوردها با نظم براساس فیلدی (معمولاً فیلد کلیدی) در بلاکها قرارداده می شوند تا هنگام انجام عملیات روی فایل سرعت بالا باشد. جهت حفظ این نظم ؛ در ابتدا مقداری از بلاکهای فایل را خالی می گذارند تا بعد با رکوردها پر شوند. این فضای خالی مانند حافظه هدر گرفته می باشد. که مقدار فضای پر شده بلاک در ابتدای کار را با L_d نشان می دهند و رابطه $\frac{L_d}{B} < 1$ برقرار است.

مزایا : ۱- لوکالیتی فایل حفظ شده و پراکندگی کمتر شده و در نتیجه زمان دستیابی کمتر می شود.

۲- عملیات درج در فایل را بدون نیاز به شیفت بلاکها امکان پذیر می سازد.

معایب : ۱- حافظه هدر گرفته زیاد شده و در نتیجه زمان خواندن کل فایل زیاد می کند.

۲- اگر توزیع رکوردها در بلاکها یکنواخت نباشد ، بعضی از بلاکها همیشه با فضای خالی باقی می مانند.

** لوکالیتی

** میزان همسایگی فیزیکی رکوردهایی که به طور منطقی (طبق تعریف برنامه پردازشگر) همسایه هستند را لوکالیتی رکوردها گویند.

** رکوردها در برنامه کاربر مطابق پردازش ، صعودی یا نزولی مرتب شده و به این وسیله رکورد بعدی معنی پیدا می کند. این حالت را همجواری یا همسایگی منطقی گویند.

** در صورتیکه فایل ترتیبی باشد که رکوردها بر حسب کلید مرتب شده و در واحدهای تخصیص حافظه (سکتور یا بلاک) مجاور ذخیره شده باشند، درلود اولیه همجواری فیزیکی و منطقی یکی خواهد بود.

** هرچه لوکالیتی رکوردها قوی تر باشد زمان پردازش سریال آنها کمتر می شود.

** **درجات لوکالیتی :** چون لوکالیتی را همسایگی فیزیکی رکوردها تعریف می کنند، بنابراین ۹ حالت از قوی به ضعیف برای این همسایگی وجود خواهد داشت:

۱- رکورد بعدی در همان بلاک رکورد فعلی که در بافر است باشد (I/O ندارد)

۲- رکورد بعدی در بلاک بعدی در همان استوانه باشد (I/O دارد اما $s=r=0$ خواهد بود)

۳- رکورد بعدی در همان استوانه باشد اما در بلاک بعدی نباشد ($r \geq 0, s = 0$ خواهد بود)

۴- رکورد بعدی روی استوانه ای هم شماره با استوانه رکورد فعلی باشد یعنی $s=0$ اما در دیسکی دیگر (RAID)

۵- رکورد بعدی در استوانه همجواری استوانه رکورد فعلی باشد ($r \geq 0, s > 0$)

۶- رکورد بعدی در استوانه ای است که آدرس آن از پردازش رکورد فعلی بدست می آید.

۷- رکورد بعدی در استوانه ای است که آدرس آن با محاسباتی پیدا می شود.

۸- رکورد بعدی دراستوانه ای است که آدرس آن با مراجعه به فایل دیگر مشخص می شود.

۹- رکورد بعدی روی استوانه ای است که هم اکنون روی درایو نمی باشد.

**** تکنیکهای تعیین محدوده طول رکورد در بلاک**

۱- درج نشان گر پایانی در انتهای رکورد.

۲- درج طول در بخش غیر داده ای رکورد.

۳- استفاده از جدول مکان نما (position table) در این جدول به ترتیبی که رکوردها در بلاک قرار می گیرند، آدرس نسبی رکوردها نسبت به ابتدای بلاک نوشته می شود. این جدول پویا بوده یعنی با اضافه شدن رکورد، آدرس آن به جدول اضافه می شود.

تکنیک جدیدی که استفاده می شود ترکیبی از روشهای ۲ و ۳ می باشد. در جدولی طول رکوردها به ترتیب نگهداری می شود.

تخصیص حافظه به بلاکهای فایل

**** حافظه اصلی و حافظه جانبی به سه روش به فایلها اختصاص داده می شود:**

تخصیص پیوسته و تخصیص ناپیوسته ، که با توجه به این دونوع تخصیص ، فایلها به دو صورت در حافظه می نشینند؛ نشست پیوسته و نشست ناپیوسته .

۱- **نشست پیوسته** : در این حالت همه بلاکهای یک فایل در بلاکهای حافظه که مجاور هم هستند، ذخیره می شوند.

مزایا : پیاده سازی آسان ، اگر آدرس اولین بلاک را داشته باشیم ، همه فایل را خواهیم داشت . (بدون نیاز به پیدا کردن آدرس بلاکها) کارایی بالای سیستم ، چون بلاکها در کمترین زمان به حافظه اصلی منتقل می شوند.

معایب : - طول کلی فایل در ابتدای کار باید مشخص باشد تا تعداد بلاک خالی مجاور مورد نیاز پیدا شود.

- اگر بعضی بلاکها را حذف کند ، جای خالی آنها فضای هدر رفته حاصل از پدیده بندبند شدگی

خواهد بود.

۲- **نشست ناپیوسته** : با این روش فضاهای پراکنده به خوبی استفاده می شوند و نیز می توان از آرایه ای از دیسکها (Disk Array) برای ذخیره فایل استفاده کرد، یعنی فایل را بلاک به بلاک یا رکورد به رکورد یا حتی در سطح کاراکتر روی دیسکهای مختلف ذخیره کرد. البته به فایلهایی که به این شکل ذخیره بشوند فایلهای توزیع شده (distributed file) یا فایلهای چند پاره یا اوراق شده (stripped file) گفته می شود. این روش به صورتهای زیر مطرح می شود:

الف (ایجاد لیست پیوندی) ، هر بلاک با اشاره گری به بلاک بعدی مرتبط می شود. از فضای دیسک به خوبی استفاده می شود. اما به علت پراکندگی ، خواندن فایلها کند می شود. (بلاکها با پیمایش زنجیره ای خوانده می شوند)

ب (ایجاد لیست پیوندی مجهز به جدول راهنما) ، در یک جدول یک ستونی ، که شماره خانه های جدول با شماره بلاکها یکی است در نظر گرفته می شود. آدرس اولین بلاک فایل را سیستم دارد و با مراجعه به آن خانه از جدول شماره بلاک بعدی را می خواند، به همین ترتیب تا در خانه آخر که null ذخیره شده. مزایای این روش : با داشتن آدرس اولین بلاک ، همه فایل را خواهیم داشت و دسترسی تصادفی سریعتر خواهد شد.

از معایب آن اینکه تمام جدول باید در حافظه اصلی ذخیره شود.

۳- استفاده از تکنیک گره I (Index Node)

در این روش برای هر فایل جدولی جدا ایجاد می شود که جدول راهنما جزئی از آن است و همچنین اطلاعات دیگر فایل (صفتها) در آن قرار داده می شود. معمولاً برای فایلهای بزرگ از چند جدول که با اشاره گری هم مرتبط شده استفاده می شود. مانند فایل سیستم درسیستم عامل یونیکس .

*** مدیریت بلاکهای آزاد :

در درو حالت می توان لیست بلاکهای خالی را نگهداری کرد.

۱- شماره یا آدرس بلاکها را در یک سری بلاک نگهداری کرد . در این حالت شماره بلاکها که معمولاً

۳۲ بیتی است . در بلاکهای ذخیره می شود (البته یک شماره هم برای اشاره به بلاک بعدی در این بلاک ذخیره می شود) در این صورت تعداد بلاکهای استفاده شده برای لیست بلاکهای آزاد زیاد می شود.

۲- استفاده از روش bit map : به این ترتیب که در هر بلاک از لیست بلاکهای آزاد از ابتدای بلاک هر بیت را متناظر با یک بلاک در سطح دیسک در نظر می گیرند . اگر بلاک خالی باشد این بیت "یک" می شود (البته در هر بلاک به اندازه آدرس هر بلاک ، اشاره گری برای اشاره به بلاک بعدی از لیست بلاکهای آزاد وجود دارد) .

*** تکنیکهای تولید نسخه پشتیبان :

۱- استفاده از نیمه دوم دیسک : به این ترتیب که دودیسک استفاده می شود و در هر دیسک علاوه بر اطلاعات همان دیسک ، اطلاعات دیسک دیگر نیز کپی می شود.

۲- تولید دامپهای تدریجی (incremental dump) : در این روش علاوه بر تولید نسخه پشتیبان ماهیانه و هفتگی ، از قسمتهایی از فایل که در طول روز نیز تغییر کرده ، نسخه پشتیبان تهیه می شود. که حافظه جانبی زیادی مصرف می شود.

۳- آینه سازی (mirroring) : معمولاً از دودیسک استفاده می شود، یکی از آنها برای عملیات ، به طور استفاده می شود و دیگری کپی اولی است ؛ و در صورت خرابی اولی ، دومی می تواند استفاده شود.

** روشهای بهبود کارایی عملیات دیسک:

روش INTERLEAVE: یا درهم چینی جهت کاهش r به کار می رود. اگر سیستم بخواهد بلاکها را پشت سرهم بخواند و پردازش کند، پس از خواندن بلاک اول، در هنگام پردازش آن، بلاک دوم که بعد از بلاک اول ذخیره شده، از زیره عبور می کند، و برای خواندن آن باید در حدود یک دور صبر کند. در این روش بلاکها را n در میان روی شیار می چینند، تا بعد از پردازش بلاک اول، ابتدای بلاک دوم زیره باشد. I ضریب درهم چینی بوده و $I=n+1$ و زمان انتقال با این روش برابر با زمان انتقال در حالت عادی تقسیم بر I .

**** روش TRACK STAGGER:** هنگامیکه هد از شیار به شیار دیگر عبور می کند، اگر روی نقطه آغاز شیار فرود بیاید، می تواند شروع به خواندن بلاکها کند در غیر این صورت باید صبر کند تا آغاز شیار زیر هد برسد.

اگر نقاط آغاز شیارها روی یک خط راست (شعاع) باشد این اتفاق می افتد. چون هنگام حرکت هد از شیار به شیار دیگر که m میلی ثانیه زمان می خواهد، شیار چرخیده و هد در نقطه ای دیگر فرود می آید. برای رفع این مشکل، بین نقاط آغاز دو شیار مجاور زاویه ای در نظر گرفته می شود، که از فرمول زیر محاسبه

$$\theta^\circ = \frac{m \times 360}{2r}$$

می شود:

**** روش فوق نیز جهت کاهش r می باشد.**

**** روشهای زیر جهت کاهش s به کار می روند:**

۱- استفاده از دیسک با هد ثابت

۲- پخش کردن فایل بر روی چند دیسک، که بطور همزمان به بلاکهای فایل دسترسی داشت.

۳- استفاده از روشهایی جهت محیطهای چند برنامه ای:

الف) روش FCFS یا FIFO: تعدادی تقاضا برای خواندن بلاکها یا شیارها مشخص می شود. جهت پاسخگویی (یعنی انتقال آنها از روی دیسک) آن تقاضایی که اول داده شده پاسخ داده می شود.

ب) SSTF (shortest seek time first): با توجه به شماره سیلندری که هد روی آن قرار دارد، آن تقاضا ابتدا پاسخ داده می شود که فاصله کمتری تا موقعیت کنونی هد داشته باشد.

ج) زمان بندی scan: هد از شماره سیلندر صفر حرکت می کند تا به شماره آخر برسد بدون تغییر مسیر سپس از آخر حرکت می کند تا به ابتدا برسد و هم در رفت و هم در برگشت نیازها را پاسخ می دهد.

د) روش c-scan (circular scan): این روش مانند حالت قبل است با این تفاوت که فقط در حالت رفت به تقاضاها پاسخ می دهد.

ه) **روش look** : در این روش مانند روش scan هدا از ابتدا حرکت می کند و تا آخرین شماره تقاضا شده به حرکت ادامه می دهد و سپس برمی گردد. و هم در رفت و هم در برگشت تقاضاها را پاسخ می دهد.

و) **روش c-look (circular look)** : مانند روش look می باشد باین تفاوت که فقط در حالت رفت از شماره پایین به بالا تقاضاها را پاسخ می دهد.

**** تعاریف کلی در سیستم فایل :**

فیلد: مکان ذخیره سازی واحد معنایی داده (semantic data unit) یا یک فقره اطلاع که حالت تجزیه ناپذیری (atomic) دارد می باشد. در فیلد مرکب یک فیلد از چند جزء کوچک تشکیل شده است.

رکورد : رکورد از سه دیدگاه بررسی می شود. ۱- سطح انتزاعی (abstractive level) ۲- سطح برنامه کاربر (منطقی) ۳- در محیط ذخیره سازی .

**** دید انتزاعی** : مفهوم رکورد مستقل از جنبه های کاربردی یا جنبه های ذخیره سازی می باشد.

**** رکورد منطقی** : بحث پیرامون مفاهیم و جنبه های رکورد از دید برنامه کاربردی می باشد.

**** رکورد ذخیره شده** : بررسی جنبه های فیزیکی ذخیره رکورد می باشد.

**** خرد جهان واقع (micro real world)** : محیطی واقعی که با ایجاد برنامه های کامپیوتری چه در جهت ذخیره داده ها (data keeping) و چه در جهت پردازش داده ها (data processing) ، یک سیستم اطلاعاتی برای آن طراحی می شود. و نهایتاً این محیط در قالب فیلد و رکورد دیده می شود.

**** موجودیت** : هر چیزی که به عنوان پارامتر اصلی در محیط واقعی عملکرد مستقل داشته باشد را شئی گویند . که در سیستم اطلاعاتی برای ذخیره وبازیابی اطلاعات آن فضا و مکانیزم طراحی می شود، مانند دانشجو یا کالا و....

وجه تمایز موجودیتها : هر موجودیت دارای اطلاعات بالارزشی است که به آنها صفت گویند (attribute) این صفتها در موجودیتها متفاوت می باشد.

**** اطلاع information** : وقتی حاصل می شود که دو وجه یک صفت خاصه مشخص شده باشد. که عبارتند از اسم صفت خاصه و مقدار صفت خاصه . مانند نام=احمد

**** رکورد از دید برنامه ساز** : یا رکورد منطقی در واقع نظم بخشیدن به همان اطلاع است که بر اساس الگوی خاصی انجام می شود که معمولاً دو الگوی : قالب ثابت مکانی (fixed positional) و قالب غیر ثابت مکانی (non fixed positional)

**** قالب ثابت مکانی** : یک رکورد با طول متغیر یا ثابت به قسمتهایی به نام فیلد تقسیم می شود که اگر ترتیب قرار گرفتن مقادیر در فیلد رکوردها یکسان باشد ، آن را قالب ثابت مکانی گویند. مثلاً فیلد اول همه رکوردها ، مقدار نام را ذخیره کند و

**** قالب غیر ثابت مکانی :** معمولاً طول رکورد متغیر می باشد. چون ترتیب قرار گرفتن فیلدها در رکوردها یکسان نیست ، پس باید نام فیلدرانیز در فیلد ذخیره کرد که با یک علامت = که دوبایت فضا می خواهد به مقدار آن مرتبط می شود.

**** دلایل متغیر شدن طول رکوردها:**

- ۱ - مقدار صفت خاصه فیلدی دارای اندازه های مختلف باشد مانند نام خانوادگی .
- ۲ - هنگامیکه رکورد برای یک موجودیت ایجاد می شود ، ممکن است موجودیت دارای صفتهای مختلفی باشد مانند موجودیت کارمند که ممکن است روزمزد باشد یا دائم .
- ۳ - پدیده اطلاع تکرار شونده ویا گروه اطلاع تکرار شونده (repeating group/item) یا به عبارتی صفت خاصه چند مقداری (multivalued) باشد. یعنی برای یک اسم یا یک فیلد چند مقدار وجود داشته باشد. وگروه تکرار شونده ، از چند فیلد که برای هریک چند مقدار وجود داشته باشد تشکیل می شود که در فیلد مرکب ذخیره می گردد واین فیلد تکرار می شود.

**** وجود پدیده فوق باعث نامسطح شدن فایل می شود. (not flat file)**

**** جهت رفع حالت نامسطح فایل اطلاعات به صورت خطی (linear form) ذخیره می گردد یعنی برای همه مقادیر یک اسم چند فیلد کنار هم در نظر گرفته شود.**

مانند موجودیت کارمند هنگام ذخیره اطلاعات افراد تحت تکفل یا ندارد یا اگر داشته باشد با دیگری متفاوت است که برای هر مورد اطلاعات شناسنامه ای باید ذخیره گردد. به این حالت گروه تکرار شونده یا صفت خاصه مرکب چند مقداری گویند.

مانند دانشجویانی که هریک ممکن است درسهای متفاوت هم در نام وهم در تعداد گرفته باشند.

**** رکوردها هنگام ذخیره توسط فایل سیستم دارای دو بخش می باشند. بخش داده ای و بخش غیر داده ای**

**** بخش غیر داده ای از دید برنامه های کاربردی و کاربر مخفی می باشد و به آن بخش پیشوندی**

(prefix section) یا بخش کنترل یا بخش سیستمی یا متا بخش (meta section) یا متا پار نیز گویند، و شامل فیلدهای زیر می باشد:

طول رکورد- نوع رکورد- نشانه روها (اشاره گریا pointer) - فلکهای عملیاتی - فلکهای حفاظتی- اطلاعات خاص.

۱- **طول رکورد :** برای رکوردهای با طول متغیر کاربرد دارد.

۲- **نوع رکورد :** بعضی فایلها دارای چند نوع رکورد می باشند که به آنها فایلهای چند نوعی گفته می شود (multi type) پس رکوردها در این فایل دارای این فیلد هستند.

۳- **نشانه روها :** جهت اشاره به داده بعدی است و انواع زیر را دارد:

رکورد به رکورد - رکورد به بلاک - بلاک به رکورد - گروهی بلاک به گروهی دیگر - فایل به فایل

۴- **پرچم یا فلاگ (flag) عملیاتی** : فلاگ در فایل سیستم عبارت است از بیتها یا بایتهایی که راهنمای عملیات خاص روی رکوردها می باشد. وبه دو صورت می باشد فلاگ عملیاتی که انجام شده مانند ثبت تاریخ آخرین عملیات تصحیح فایل .

فلاگ عملیاتی که بعداً انجام می شود مانند رکوردی که علامت پاک شدن میگیرد تا هنگام بازسازی فایل پاک شود.

۵- **فلاگ حفاظتی** : یا قفل در سطح رکورد (record level lock) مانند حق دستیابی برای خواندن و نوشتن

حق نوشتن مانند درج رکورد جدید - حذف رکورد - بهنگام سازی رکورد.

این حفاظتها در سطوح مختلف انجام می شود : سطح رکورد - سطح بلاک - سطح گروهی بلاکها - سطح فایل یا گروهی فایل وحتى در سطح فیلد نیز مطرح می شود.