فصل سوم

بررسی پارامتر های دیسک و نوار مغناطیسی

پارامتر های اساسی نوار

1) پارامتر های ظرفیتی : (ظرفیت اسمی ، ظرفیت چگالی ، ظرفیت واقعی و .... .)
2) پارامتر های زمانی : (نرخ انتقال واقعی و نرخ انتقال اسمی ، ...)

(a) درصد واقعی نوار : اگر بخواهیم B بایت اطلاعات را روی نوار ذخیره کنیم به B+G بایت فضا نیاز داریم : بنابراین

(درصد ظرفیت واقعی) $x=\frac{B}{B+G}\times 100\%$

(b) ظرفیت اسمی نوار مغناطیسی : اگر Lt طول نوار مغناطیسی بر حسب اینچ و D چگالی نوار بر حسب بایت در اینچ باشه آنگاه ظرفیت اسمی نوار T برابر میشه با

(ظرفیت اسمی نوار) $T = l_t \times D \rightarrow \frac{byte}{inch}$

byte inch

درصد ظرفیت واقعی × ظرفیت اسمی = ظرفیت واقعی نوار

پس :

ظرفیت واقعی نوار = $L_t \times D \times \frac{B}{B+G}$

مثال: در یک نوار مغناطیسی با چگالی 800 بیت در هر اینچ 120 هزار بایت اطلاعات قابل ذخیره‌سازی است طول این نوار کدام است؟

$$L_T = \frac{\text{کل اطلاعات}}{D} = \frac{120000}{100} = 1200\,\text{inches} \quad \text{(ظرفیت واقعی نوار)}$$

$$D = 800\ bpi = 100\frac{Byte}{inch}$$

کل اطلاعات $= 120000\ \text{Byte}$

مثال : در یک نوار دارای چگالی 1600 بایت در هر اینچ که 80% اطلاعات ذخیره شده روی آن مفید میباشد ، اطلاعات در قالب بلاک های 480 بایتی ذخیره شده اند طول گپ بین بلاک ها چند اینچ است ؟

$$\frac{B}{B+G} = \frac{80}{100}$$

$$\frac{480}{480+G} = \frac{8}{10} \rightarrow \frac{480}{480+G} = \frac{8}{10} \rightarrow 3840+8G=4800$$

8G=4800-3840
8G=960
G=120 Byte

$$G = LG \times D \Rightarrow L_G = \frac{G}{D}$$

حالا

$$\rightarrow \frac{120}{1600} = \frac{3}{40} = 0.75\ inch$$

رابطه های مورد نیاز برای حل مسایل فصل :

D=1600
G = 120
LG = x

طول گپ بر حسب بایت

$$G = L_G \times D \rightarrow L_G = \frac{G}{D}$$

طول گپ بر حسب اینچ

$$B = L_B \times D \rightarrow L_B = \frac{B}{D}$$

$$T = L_T \times D \rightarrow L_T = \frac{T}{D}$$

$$F = L_F \times D \rightarrow L_F = \frac{F}{D}$$

مثال : یک فایل شامل 1000 رکورد 60 بایت با ضریب بلاکینگ 10 در نصف نوار مغناطیسی دارای چگالی 800 بایت در اینچ ذخیره شده است . اگر گپ بین بلاک ها $1/2$ اینچ باشد طول نوار مغناطیسی چند اینچ است؟

اگر l_t طول نوار بر حسب اینچ و $L_B + L_G$ طول بلاک همراه باشد آنگاه تعداد بلاکهای نوار برابر است با :

$$b = \frac{L_T}{L_B + L_G}$$

$$B_F = \frac{n}{b} \rightarrow n = B_F \times b$$

فرمول نهایی $\rightarrow$ $\boxed{n = B_F \times \frac{L_T}{L_B + L_G}}$

نکته چون 1000 رکورد در نصف نوار قابل ذخیره سازی است بنابراین 2000 رکورد بر روی کل نوار قابل ذخیره سازی است

$$n=B_F \times \frac{Lt}{(LB)+LG}$$

$$B = BF \times R = 10 \times 60 = 600$$
$$LB = \frac{B}{D} = \frac{600}{800} = \frac{3}{4}$$

$$2000=10 \times \frac{Lt}{\frac{3}{4}+\frac{1}{2}} \rightarrow 200 = \frac{Lt}{\frac{3+2}{4}} \rightarrow \qquad L_T = 200 \times \frac{5}{4} = 250$$

پارامترهای زمانی نوار مغناطیسی

1. نرخ انتقال اسمی نوار : میزان اطلاعات منتقل شده در واحد زمان در حالت ایده آل را نرخ انتقال اسمی گویند که با t کوچک نمایش داده میشود.

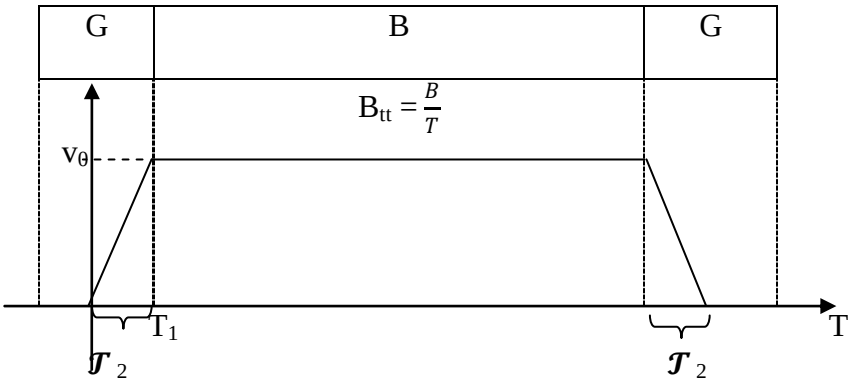
2. نرخ انتقال واقعی : میزان اطلاعات منتقل شده در واحد زمان را نرخ انتقال واقعی گویند که با t' نمایش داده میشود که همواره t'<t

عوامل کاهنده نرخ انتقال واقعی عبارتند از :

1. زمان طی کردن گپ یا G_{tt} $\qquad G_{tt} = \frac{G}{t}$
2. زمان حرکت / توقف یا $\boldsymbol{\mathcal{T}}$ $\qquad \boldsymbol{\mathcal{T}} = \boldsymbol{\mathcal{T}}_1+\boldsymbol{\mathcal{T}}_2$

به طور کلی یک نوار مغناطیسی را به دو روش میتوان خواند :

1. روش بلاکی یا Blocking Mode :که در این روش هد پس از خواندن هر بلاک متوقف میشود.



پس زمان واقعی خوندن برابر میشود با :

زمان واقعی خوندن یک بلاک = $\mathcal{T}_1 + B_{tt} + \mathcal{T}_2$

از اونجایی که $\mathcal{T}=\mathcal{T}_1+\mathcal{T}_2$

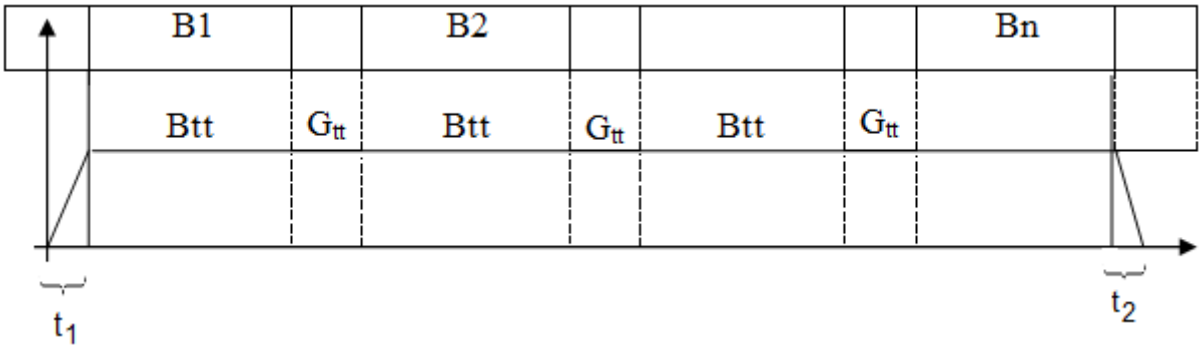
پس

زمان واقعی خوندن یک بلاک = $B_{tt} + \mathcal{T}$

نرخ انتقال واقعی = $\frac{\text{میزاناطلاعات منتقل شده}}{\text{کل زمان}}$

$t' = \frac{B}{Btt + \mathcal{T}}$

2) روش جریانی stream mode : هد پس از خواندن n بلاک می ایستد.



زمان واقعی خواندن N بلاک = $\mathcal{T}_1$+ NBtt + NGtt+ $\mathcal{T}_2$

زمان خواندن واقعی➔NBtt + NGtt+ $\mathcal{T}$

$$t' = \frac{NB}{NBtt + NGtt + T} \quad (1)$$

چون $B_{tt} = \frac{B}{t}$ است بنا بر این رابطه به صورت زیر قابل تغییر است:

$$t' = \frac{NB}{N\frac{B}{t} + N\frac{G}{t} + \mathcal{T}}$$

$$t' = \frac{NB}{\frac{NB + NG + \mathcal{T}}{t}}$$

$$t' \neq \frac{\frac{NB}{1}}{\frac{NB+NG+\mathcal{T}}{T}}$$

$$t' = \frac{NB}{NB+NG+\mathcal{T}} \times t \qquad t' = \frac{NBtt}{NBtt+NGtt+\mathcal{T}} \times t$$

از آنجایی که $B_{TT} = \frac{B}{t}$ است پس $B = B_{TT} \times t$ است، بنابراین اگر در رابطه (1) بجای $B = B_{TT} \times t$ را جای گزین کنیم آنگاه :

$$t' = \frac{}{N}$$

(3)

مثال: در یک نوار مغناطیسی 20000 اینچی که در هر اینچ آن 800بایت اطلاعات ذخیره شده است رکورد هایی 80 بایتی با پارامتر بلاک بندی 8 ذخیره شده اند اگر در هر ثانیه 200 بایت اطلاعات از نوار منتقل شود. مطلوب است :

الف زمان طی کردن گپ 2.0 اینچی چند میلی ثانیه است ؟

را حل:

$$G_{TT} = \frac{G}{G}$$

$$G = L_G$$

ب: درصد اطلاعات واقعی نوار چیست ؟

راه حل:

$$x = \frac{B}{B}$$

$$B = BF$$

ج: تعداد رکورد های ذخیره شده در سطح کل نوار را بدست آورید .

راه حل :

$$n = BF \times \frac{L_T}{L_B+L_G} = 8 \times \frac{2000}{\frac{8}{10}+\frac{2}{10}} = 160000$$

$$L_B = \frac{B}{D}$$

مثال: 75 درصد یک نوار مغناطیسی شامل اطلاعات مفید فایل F است. این فایل دارای رکوردهای 100 بایتی بوده که در سطح شیارهای نوار مغناطیسی به طول 2000 اینچ و چگالی 600 بایت در اینچ ذخیره شده است. بین هر دو بلاک $\frac{1}{3}$ اینچ گپ وجود دارد. تعداد رکوردهای فایل کدامست؟

$$L_G = \frac{1}{3} inch \quad \Rightarrow G = L_G \times D = \frac{1}{3} \times 600 = 200\ Byte$$

$$\frac{B}{B+G} = \frac{75}{100} \Rightarrow \frac{B}{B+200} = \frac{75}{100} \qquad \Rightarrow 100B = 75B + 15000$$
$$\Rightarrow 25B = 15000 \Rightarrow B = 600 Byte$$

$$B_f = \frac{B}{R} = \frac{600}{100} = 6$$

$$n = B_f \times b = B_f \times \frac{L_T}{L_B + L_G} = 6 \times \frac{2000}{1 + \frac{1}{3}} = \frac{6 \times 2000}{\frac{4}{3}} = \frac{6 \times 2000 \times 3}{4} = 9000\ \mathrm{Re}cords$$

$$L_B = \frac{B}{D} = \frac{600}{600} = 1inch$$

مثال: در یک نوار مغناطیسی دارای زمان حرکت توقف 200 میلی ثانیه و نرخ انتقال اسمی 25 بایت در میلی ثانیه زمان انتقال واقعی بلاک 400 بایتی 18 میلی ثانیه طول میشکد طول گپ بین بلاک ها چند بایت است.

نکته : زمان واقعی خواندن بلاک از دو روش قابل محاسبه است :

(روش بلاکی) زمان واقعی خواندن= $\frac{B}{t} + \boldsymbol{\mathcal{T}}$

(روش جریانی) زمان واقعی خواندن= $N\frac{B}{t} \times N\frac{G}{t} + \boldsymbol{\mathcal{T}}$

جون مسئله فضای گپ را می خواهد پس از روش جریانی باید عمل کرد چون تو روش بلاکی G نداریم

در روش جریانی اگر تعداد بلاکها یعنی N=1 است بنابراین :

زمان خواندن واقعی= $\frac{B}{T} + \frac{G}{T} + \boldsymbol{\mathcal{T}}$

$\frac{1\ msec}{x} \times \frac{1000\ \mu sec}{200\ \mu sec} \rightarrow 0.2$

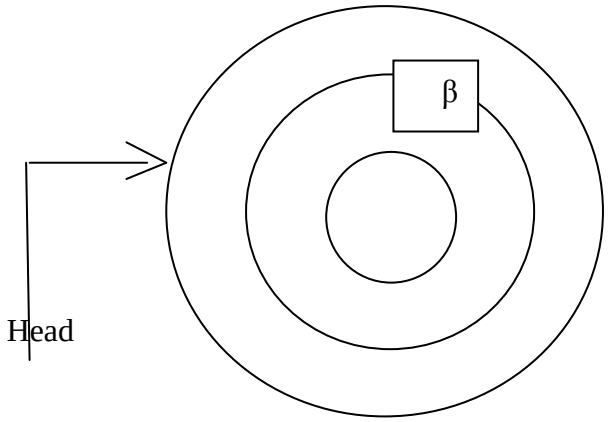
$18 = \frac{400}{25} + \frac{G}{25} + \frac{2}{10} \rightarrow 18 = \frac{800 + 2G + 10}{50} \rightarrow 900 = 810 + 2G \rightarrow 900 - 810 = 2G$

G = 45 byte

پارامترهای دیسک

(1) پارامترهای ظرفیتی (تعداد رویه، تعداد شیار در هر رویه، تعداد سکتورهای هر شیار، اندازه‌ی سکتور و چگالی دیسک)

(2) پارامترهای زمانی (زمان جستجو، زمان درنگ دورانی و زمان انتقال)



فرض مسأله: یک بلاک B بایتی در شیاری از دیسک قرار داشته و هد دیسک در یک محل اولیه (شیار) قرار دارد.

حکم مسأله: مراحل دسترسی به بلاک B بایتی

(a) زمان جستجو استوانه‌جویی (Seek time): مدت‌زمانی که لازم است تا نوک هد دیسک از محل جاری به سیلندر موردنظر برسد را زمان استوانه‌جویی گویند و متوسط آن با S نشان داده می‌شود.

(b) زمان درنگ دورانی (Rotational time): مدت‌زمانی که لازم است تا آغاز داده‌ی یک شیار در زیر هد خواندن ـ نوشتن قرار بگیرد را گویند. متوسط این زمان را با r نشان می‌دهند.

نکته 1: یک دور چرخش کامل دیسک بطور متوسط 2r میلی‌ثانیه طول می‌کشد که برابر زمان بازنویسی اطلاعات است. بعبارت دیگر اگر فیلدی غیر از فیلد اصلی تغییر کند زمان ثبت مقدار جدید بجای مقدار قبلی آن فیلد $T_{RW} = 2r$ میلی‌ثانیه طول می‌کشد.

نکته 2: تعداد دورهای یک دیسک در دقیقه با PRM(Rotation Per Minute) نمایش داده شود که با زمان یک دور دیسک ارتباط دارد.

$$\Rightarrow r = \frac{60000}{2\times rpm} = \frac{30000}{RPM}(m\sec) \frac{\text{زمان (میلي ثانيه)}}{60\times 1000} \frac{\text{دور}}{\text{RPM}}$$

$$r = \frac{30}{RPM}(Sec)$$

ارزیابی دقیق‌تر زمان درنگ دورانی

نکته: اگر در هر شیار به طور متوسط T_f بلاک دیسک ذخیره شده باشد. سیستم پس از شناسایی نشانگر آغاز شیار، بطور متوسط از شروع شیار تا آغاز بلاک موردنظر $\frac{T_f - 1}{2T_f}$ از شیار را طی کرده و زمان درنگ دروانی به r' افزایش می‌یابد.

$$r' = r + \frac{1}{2} \times \frac{t_f - 1}{t_f}(2r) \Rightarrow r' = r + \frac{T_f - 1}{T_f}(r) \Rightarrow r' = r(1 + \frac{T_f}{T_f} - \frac{1}{T_f}) \Rightarrow r' = r(2 - \frac{1}{f})$$

نکته: زمان $S + r$ مدت‌زمان بین‌لحظه‌ای که دستور خواندن / نوشتن داده شده و لحظه‌ای که آغاز بلاک موردنظر زیر هد دیسک قرار می‌گیرد. این زمان را زمان دستیابی تصادفی Random Access Time گویند.

(c) زمان انتقال (b_{tt}): مدت‌زمانی که لازم است تا پس از رسیدن نوک هد به بلاک Bبایتی آن بلاک با نرخ t از حافظه جانبی به حافظه اصلی منتقل شود. $b_{tt} = \frac{B}{t}$

رابطه‌ی سه پارامتر زمانی

زمان انتقال $\geq$ زمان جستجو $\geq$ زمان درنگ دورانی

نکته: از آنجا که اندازه‌ی بلاک هرگز از اندازه‌ی شیار بزرگتر نیست لذا $b_{tt} \leq 2r$ است.

مثال: در دیسک دارای PRM=2000 یک دور چرخش دیسک چند ثانیه طول می‌کشد.

$$r = \frac{30000}{RPM} = \frac{30000}{2000} = 15 m\sec = 0.015 Sec$$

یک دور چرخش کامل دیسک $= 2r = 2 \times 0.015 = 0.03\ Sec$

مثال: در دیسک دارای هد ثابت که در هر دور چرخش آن 40 میلی‌متر ثانیه طول می‌کشد و در هر ثانیه 60000 بایت را به حافظه اصلی منتقل می‌کند. زمان بازیابی بلاک 300 بایتی چند میلی‌ثانیه است؟

نکته: در دیسک دارای هد ثابت چون بازوی هد حرکت نمی‌کند زمان جستجو صفر است. $S = \circ !$

$$2r = 40 \Rightarrow r = 20\, m\sec$$

$$t = 60000 \frac{Byte}{Sec} = 60 \frac{Byte}{m\sec}$$

$$\Rightarrow \quad Access\ Time = S + r + \frac{B}{t} = 0 + 20 + \frac{300}{60} = 25 m\sec$$

ارزیابی پارامتریک دیسک

(1) زمان دسترسی و نرخ انتقال واقعی دیسک:

در صورتی که در هر شیار T_f بلاک وجود داشته باشد.

اگر در هر شیار T_f بلاک وجود داشته باشد آنگاه:

(1) $Access\ Time = S + r + b_{tt}$

(2) $Access\ Time = S + r' + b_{tt}$

خواندن تصادفی فایل $= b(s + r + b_{tt})$ $\Rightarrow$ نرخ انتقال واقعی دیسک $= \frac{B}{AccessTime}$

نرخ انتقال واقعی $= \frac{\text{میزان اطلاعات مفید}}{\text{زمان کل انتقال اطلاعات}}$

در دیسک داریم $\Rightarrow t' = \frac{B}{Access\ Time}$

$Access\ Time = S + r + b_{tt}$

$Access\ Time = s + r' + b_{tt}$

زمانی که در هر شیار T_f بلاک وجود داشته باشد.

خواندن تصادفی کل فایل شامل b بلاک $T_x = b_N(s + r + b_{tt})$

نرخ انتقال انبوه : (Bulk Transfer Rate)

اگر قرار باشد تعداد بلاک پشت سرهم (پی‌درپی) همراه فضای Gap بین آنها خوانده و به حافظه اصلی منتقل گردد. از طرفی به ازای هر رکورد نیز W_R فضای تلف شده وجود داشته باشد زمان انتقال افزایش یافته و میزان اطلاعات منتقل شده در واحد زمان کاهش می‌یابد که به آن نرخ انتقال انبوه Bulk Transfer Rate گفته می‌شود. اگر هر سیلندر K شیار و هر رویه شیار n رکورد Rبایتی داشته باشد بطوری‌که به ازای هر رکورد W_R بایت فضای تلف شده داشته باشد. طبق تعریف نرخ انتقال انبوه در دو حالت زیر بررسی می‌گردد:

(A) تمام بلاک‌های فایل در یک سیلندر ذخیره شده‌اند در این حالت پس از خواندن فایل و انتقال یک بلاک نیازی به زمان جستجو (استوانه‌جویی) برای یافتن محل سیلندر (بلوک بعدی) نیست.

$$t' = \frac{\text{اطلاعات مفید}}{\left(\frac{\text{کل اطلاعات}}{\text{نرخ انتقال}}\right)} \Rightarrow \frac{KnR}{\frac{kn(W+W_R)}{t}} = \frac{R}{(\frac{R+W_R}{t})} = \frac{R.t}{R+W_R} \quad (1)$$

(B) اطلاعات فایل از سیلندرهای مختلف پخش شده‌اند. در اینصورت پس از خواندن اطلاعات بلاکی در یک سیلندر دیسک، برای یافتن بلاک بعدی به زمان جستجوی (Si) سیلندر حاوی بلاک بعدی نیاز است. در رابطه (1) بفرم کلی زیر تبدیل می‌شود.

$$t'' = tc = \frac{R}{\frac{R+W_R}{t} + Si}$$

از طرفی Si زمان جستجو به ازای یک رکورد می‌باشد یعنی $Si = \frac{2r}{n} = \frac{R+W_R}{t}$ و لذا نرخ انتقال انبوه در این حالت عبارتست از:

$$tc = \frac{RC}{2(R+W_R)} = \frac{t'}{2}$$

نکته 1: نرخ انتقال‌های tc و t'، این شرط بدست آمده‌اند که زمان پردازش یک بلاک موجب به تأخیر انداختن خواندن بلاک (b_{tt}) نشود و از زمان مذکور کمتر باشد یعنی $CB < b_{tt}$

نکته 2: در دستیابی پی‌درپی و پشت سرهم به رکوردها و بلاک‌ها t' دخالت داده می‌شود و زمان انتقال بلاک $b_{tt} = ebt = \frac{B}{t'}$ خواهد بود و $B = B_f \times R$.

نکته 3: در دسترسی تصادفی به بلاک‌ها t نفش دارد. شرط اینکه تعداد زیادی بلاک را بصورت تصادفی پردازش کنیم این است که $CB < (S + r + b_{tt})$ باشد.

نکته 4: از آنجا که $tc = \frac{t'}{2} < t' \le \frac{Rt'}{R+W_R} < t$ و نرخ انتقال رابطه معکوس با زمان انتقال دارد لذا $ebt > b_{tt}$ است. می‌دانیم $ebt = \frac{B}{t'}$, $b_{tt} = \frac{B}{t}$ و چون $\begin{matrix} t' > t' \\ t' < t \end{matrix}$ است لذا رابطه‌ی $ebt > b_{tt}$ صحیح است.

مثال 1: زمان خواندن چهار سکتور در دیسک دارای هد ثابت و نرخ انتقال $1\frac{MB}{Sec}$ که در هر دور چرخش آن 30 میلی‌ثانیه طول می‌کشد چند میلی‌ثانیه است.

$B = 512\ Byte$

$2r = 30 \Rightarrow r = 15\ m\sec$

$S = 0$

T_x = زمان خواندن چهار سکتور $= 4(S + r + \frac{B}{t}) = 4(0 + 15 + \frac{512}{1000}) = 62.048\ m\sec$

$t = 1\frac{MB}{Sec} \Rightarrow x = 1000\frac{Byte}{m\sec}$

مثال 2: تعداد 16 هزار رکورد 75 بایتی در 2 هزار بلوک در سطح یک سیلندر دیسکی دارای زمان جستجوی 60 میلی‌ثانیه که در هر دقیقه 3000 دور می‌چرخد و نرخ انتقال آن 30 بایت در میلی‌ثانیه بوده ذخیره شده‌اند به ازای هر رکورد 15 بایت فضای هرز وجود داشته و هر شیار 5 بلاک دارد، هر شیار 6 کیلوبایت است؟

الف) زمان بازیابی هر بلاک چند میلی‌ثانیه است؟

ب) درصد استفاده مفید از شیار چیست؟

$n = 16000$

$R = 75\ Byte$

(2) $t' = \frac{R * t}{R + W_R} = \frac{75 * 30}{75 + 15} = 25\frac{Byte}{m\sec}$

$b = 2000$

$S = 60 m\sec$

(3) $b_{tt} = \frac{B}{t'} = \frac{600}{25} = 24 m\sec$

$PRM = 3000$

$t = 30\frac{Byte}{m\sec}$

$r = \frac{30000}{RPM} = 10 m\sec \Rightarrow r' = r(2 - \frac{1}{T_f}) = 10(2 - \frac{1}{5}) = 18 m\sec$

$W_R = is\ Byte$

$T_f = 5$

(5) $At = S + r' + ebt = 60 + 18 + 24 = 102\ m\sec$

(6) درصد استفاده واقعی از شیار $\frac{TF \times B}{\text{اندازه شیار}} = \frac{600 \times 5}{6 \times 1024} = \frac{500}{1024} = 48\%$

$$\begin{cases} b_f = \frac{n}{b} = \frac{16000}{2000} = 8 \\ B = R * B_f = 75 \times 8 = 600\ \text{Byte} \end{cases}$$

مثال: رکوردهای 80 بایتی فایل f روی سیلندرهای دیسک دارای rpm= 2000 و نرخ انتقال ایده‌آل 25 بایت در میلی‌ثانیه ذخیره شده‌اند با توجه به آنکه نرخ انتقال انبوه به 10 بایت در میلی‌ثانیه کاهش یافته است. فضای هرز هر رکود فایل چند بایت است؟

$$t'c = \frac{R_t}{2(R+W_R)} \Rightarrow \frac{80 \times 25}{2(80+W_R)} \Rightarrow 20(80+W_R) \Rightarrow W_R = 20\,Byte$$

نکته: زمان خواندن b بلاک بصورت ترتیبی (پشت سرهم) از رابطه‌ی زیر بدست می‌آید:

با زیاد شدن تعداد بلاک‌ها اثر زمان S + r در مقابل $T_x = S + r + b * ebt$ کمرنگ می‌شود. بنابراین می‌توانیم از زمان S + r صرف‌نظر کنیم.

مثال 4: در دیسک دارای 1000= rpm و $20\frac{byte}{m\sec}$ = نرخ انتقال اسمی و هد ثابت زمان بازیابی پی‌درپی بلاکهای فایل حاوی رکوردهای 40 بایت که به ازای هر رکورد 10 بایت بلااستفاده وجود دارد. چند میلی‌ثانیه است؟ $(B_f = 8)$

$$\left.\begin{array}{l} r = \frac{30000}{rpm} = \frac{30000}{1000} = 30ms \\ t = 20\frac{8}{m\sec} \Rightarrow t' = \frac{R_t \times t}{R+W_R} = \frac{40 \times 20}{40+10} = 16\frac{b}{ms} \end{array}\right\} \Rightarrow At = S + r + \frac{B}{t'} = 0 + 30 + \frac{320}{16} = 50m\sec$$

$$BF = 8 \Rightarrow B = B_f \times R = 8 \times 40 = 320\,Byte$$

ظرفیت واقعی دیسک

در ارزیابی ظرفیت واقعی، نحوه فرمت‌بندی شیار تکنیک بلاک‌بندی نقش مهمی دارند بطوریکه با استفاده از آنها فضای بلااستفاده در یک شیار را محاسبه و آنرا از ظرفیت اسمی شیار کم می‌کنیم بدین‌ترتیب ظرفیت واقعی (شیار) بدست می‌آید.

یکی از راه های افزایش ظرفیت واقعی دیسک این است که اندازه‌ی بلاکها ضریب صحیحی از اندازه‌ی سکتورها باشد بعبارتی بلاکها در تعداد صحیحی از سکتورها ذخیره شوند. تا آخرین سکتور فضای هرزی نداشته باشد. .

$$\text{درصد استفاده واقعی از دیسک} = \frac{B}{L_S \times N}$$

$$\text{یا تعداد سکتور در هر بلاک} = \left[\frac{B}{L_S}\right]$$

حداکثر اندازه بلاک به چه عواملی بستگی دارد

(1) امکانات بافرینگ

(2) نحوه پردازش فایل (اگر فایل تصادفی باشد طولانی بودن بلاک، زمان پردازش را افزایش می‌دهد)

(3) استفاده چند برنامه از یک فایل (در این حالت اندازه‌ی بلاک = اندازه کوچکترین بافر تخصیصی)

(4) نوع سیستم عامل. اندازه‌ی بلاک بستگی به اندازه‌ی صفحه دارد (page) در VSAM از Page=2KB و در سیستم‌عامل NMS، Page = 32KB است.

نتیجه‌ی کلی: میزان استفاده واقعی از هر شیار به نحوه‌ی بلاک‌بندی و فرمت‌بندی شیار بستگی دارد: در سیستم IBM370، دو نوع فرمت برای بلاک وجود دارد:

(a) فرمت داده‌ها (B)، اطلاعات کنترلی (c) $\Leftarrow \quad T_f = \left[\frac{C_{Nt}}{B+C}\right]$

(b) فرمت داده‌ها (B)، اطلاعات کنترلی (c)، کلید(K) $T_f\left[\frac{T_{Nt}}{B+C+K}\right]$

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

نکته: کاهش در ظرفیت واقعی عمدتاً ناشی از وجود گپ‌هاست و تاحدی نیز به طول بلاک ظرفیت شیار و نحوه فرمت آن بستگی دارد.

مثال: اطلاعات فایل f حاوی رکوردهای 80 بایتی با پارامتر بلاک‌بندی، در سطح شیارهای 4 کیلوبایتی ذخیره شده‌اند، در ابتدای هر بلاک بخش 20 بایتی وجود دارد درصد استفاده واقعی از شیار را بدست آورید.

$$T_f = \left[\frac{C_{Nt}}{B+C}\right] = \left[\frac{4096}{500}\right] = 8$$

$$\left.\begin{array}{l} B = B_f \times R = 6 \times 80 = 480 \\ C = 20 \Rightarrow \end{array}\right\} B + C = 500$$

$$P = \frac{T_f \times B}{C_{Nt}} = \frac{8 \times 480}{4096} \times 100\% = 93.75\%$$

مثال: در دیسک سکتوربندی شده حاوی بلاکهای 2 کیلوبایتی و سکتورهای 512 بایتی میزان استفاده واقعی از حافظه دیسک 25 درصد است تعداد سکتور در بلاک این دیسک چقدر است؟

$$B = 2KB = 2 \times 2^{10} = 2^{11} Byte$$

$$\Rightarrow L_S = 512 = 2^9 Byte \quad \text{درصد استفاده واقعی} = \frac{B}{L_S \times N}$$

$$\Rightarrow \frac{25}{100} = \frac{2^{11}}{2^9 \times N} \Rightarrow \frac{1}{2^2} = \frac{2^{11}}{2^9 N}$$

$$\Rightarrow 2^9 N = 2^{13} \Rightarrow N = 16$$

مثال: دیسک دارای شیار $cnt + 32KB$ و اطلاعات بلاک $B = 2000$ و $C = 150$ و $rpm = 1000$ زمان درنگ دورانی را بدست آورید.

$$r = \frac{30000}{RPM} = \frac{30000}{1000} = 30ms$$

$$T_f = \left[\frac{C_{Nt}}{B + C} \right] = \left[\frac{32 \times 1024}{2150} \right] = 15$$

$$r' = r(2 - \frac{1}{T_f}) = 30(2 - \frac{1}{15}) = 58$$

مثال) زمان خواندن مستقیم یک بلاک 2 ثانیه است در صورتی که طول بلاک B=10000 byte و مقدار حافظه هرز بلاکی 2000 byte با شد نرخ انتقال واقعی چند k byte \ sec است ؟

$$TD = \frac{B - W_B}{s + r + B_{tt}} \Longrightarrow \frac{10000 - 2000}{2} \Longrightarrow 4000bps = \frac{4000}{1024} kbps$$

مثال) اگر تعداد دور در دقیقه یک دیسک برابر 1200 RPMباشد متوسط زمان درنگ دورانی چند ثانیه است ؟

$$R = \frac{1}{2} \qquad \frac{(60 * 1000)}{1200} = \frac{300}{12} = 25ms = 0.025s$$

فصل چهارم

«تکنیک‌های بهبود کارآیی عملیات دیسک»

همانطور که در فصل قبل مطرح شد مهمترین پارامترهای دیسک عبارتند از:

(1) زمان استوانه‌جویی (S)

(2) زمان درنگ دورانی، (r)

(3) زمان انتقال بلاک (b_{tt}) بودند که تأثیر مستقیم بر زمان دسترسی (بازیابی) به اطلاعات بلاک‌های ذخیره شده روی دیسک دارند. در این فصل راه‌های تسریع در بازیابی بلوک و بعبارتی تکنیک‌ها کاهش زمان دسترسی به بلاک بررسی می‌گردد.

4-1- روش‌های کاهش زمان استوانه‌جویی

عمده‌ترین روش‌ها جهت کاهش زمان جستجو عبارتند از:

(الف) استفاده از دیسک‌های دارای هد ثابت.

(ب) توزیع فایل روی سیلندرهای هم‌شماره از چند دیسک

(ج) بکارگیری الگوریتم‌های سیستم‌عامل جهت زمان‌بندی دیسک.

(د) جایگاه مناسب رکوردها در فایل.

در ادامه به شرح جامع هر یک از چهار روش فوق پرداخته می‌شود.

4-1-1- استفاده از دیسک دارای هد ثابت

همچنان که در فصل قبل گفته شد در این دیسک‌ها از آنجا که بازوی هد حرکتی ندارد به ازای هر شیار از رویه دیسک، یک هد خواندن ـ نوشتن به بازو وصل بوده و بازو حرکتی ندارد یعنی $S=0$. لذا در دیسک‌های دارای هد ثابت، زمان دسترسی به اطلاعات کمتر و سریعتر از دیسک‌های با هد متحرک است.

4-1-2- توزیع فایل روی سیلندرهای هم‌شماره چند دیسک

این روش، تکنیکی نرم‌افزاری در جهت شبیه‌سازی تکنیک اول است. به‌عبارتی با ذخیره‌سازی فایل سیلندرهای هم‌شماره از چند دیسک (همان‌طور که در بحث لوکولیتی مطرح شد) زمان جستجو صفر خواهد شد زیرا نوک هد حرکتی ندارد.

4-1-3- بکارگیری الگوریتم‌های سیستم‌عامل جهت زمان‌بندی دیسک

در یک محیط چند برنامه‌ای که در آن سیستم باید به درخواست‌های ورودی ـ خروجی چند برنامه پاسخ دهد نیاز به الگوریتمی است که بر اساس آن بازوی هد حرکت کرده و به سیلندرهای متقاضی پردازش، سرویس لازم را ارائه نماید.

فرض (ورودی) این الگوریتم‌ها، ترتیبی از سیلندرهاست که درخواست خواندن و پردازش اطلاعاتشان را به سیستم‌عامل داده‌اند و همین‌طور محل ابتدایی نوک هد خواندن ـ نوشتن و زمان حرکت بازوی هد از یک سیلندر به سیلندر بعدی از ورودی‌های این الگوریتم‌ها بشمار می‌رود. خروجی این الگوریتم‌ها ترتیب خواندن اطلاعات سیلندرهای متقاضی و کل زمان حرکت بازوی هد می‌باشد.

شرح کامل این الگوریتم‌ها چنین است:

الف) الگوریتم FIFO یا FCFS (First Comed First Serviced):

در این روش سیلندرهای متقاضی به ترتیب درخواست‌شان سرویس‌دهی می‌شوند. بعبارتی هر درخواست در صف اجرا قرار می‌گیرد. ساده‌ترین روش است اما کارآیی چندانی ندارد.

ب) الگوریتم SSTF (Shortest Seel Time First):

در این روش، هر لحظه سیلندر متقاضی که به محل هد در آن لحظه نزدیکتر باشد مورد پردازش قرار می‌گیرد. معمولاً این روش حداقل زمان جستجو را به همراه دارد. اما ضعف عمده‌ی آن احتمال قحطی‌زدگی (Starvation) (برای سیلندرهایی که دور از محل هد هستند) می‌باشد.

ج) الگوریتم SCAN (آسانسور):

این روش خود به دو دسته تقسیم می‌شود.

1. تکنیک آسانسور با بیت بالا (UP): در این روش هد از محل اولیه خود رو به جلو حرکت کرده و اطلاعات سیلندرهای متقاضی را می‌خواند. سپس مانند آسانسور به طرف سیلندرهای ابتدایی برگشته و سیلندرهای متقاضی باقی‌مانده که اطلاعاتشان خوانده نشده سرویس‌دهی می‌شوند.

2. روش آسانسور با بیت پائین (Down): در این روش ابتدا هد از محل اولیه‌اش به سمت سیلندرهای ابتدایی حرکت و سرویس‌دهی می‌کند. سپس مانند آسانسور رو به جلو حرکت کرده و مابقی سیلندرهای متقاضی که سرویس‌دهی نشده‌اند را می‌خواند.

د) الگوریتم C – SCAN:

مانند روش آسانسور به دو روش cScan-Up و cScan-Down تقسیم می‌شود. وقتی هد به انتهای دیسک رسید مجدداً از اولین سیلندر دیسک شروع کرده و سیلندرهای متقاضی باقی‌مانده را سرویس‌دهی می‌کند. یعنی مانند آسانسور به عقب برنمی‌گردد. در روش cScan-Down هم وقتی هد از محل اولیه خود رو به سیلندرهای پائین حرکت کرد و به نخستین سیلندر رسید به انتهای سیلندرهای متقاضی رفته و از آنجا نیز رو به عقب برگشته سیلندرهای درخواست‌کننده را سرویس‌دهی می‌نماید.

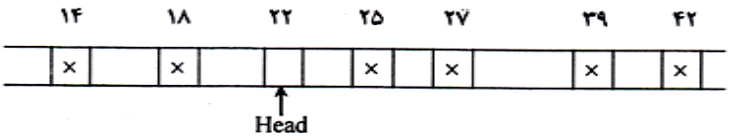
به عبارتی در روش C- SCAN ترتیب صعودی یا نزولی پردازش اطلاعات سیلندرها حفظ می‌گردد.

نکته 1: در روش SSTF، اگر دو سیلندر متقاضی از محل کنونی هد فاصله یکسانی داشتند زمان‌بندی دیسک باید به صورت ترکیبی تعریف شود. مثلاً ترکیبی از SSTF و FCFS یا SSTF و SCAN-UP. در این صورت از آن دو سیلندر متقاضی که به یک فاصله از محل کنونی قرار گرفته‌اند یکی زودتر سرویس‌دهی می‌شود.

مثال: درخوست پردازش سیلندرهای 25 و 42 و 18 و 27 و 14 و 39 به ترتیب چپ به راست به سیستم عامل داده شده و در ابتدای کار نوک هد برروی سیلندر 22 واقع است. اگر انتقال بازوی هد از یک سیلندر به سیلندر بعدی 4 میلی‌ثانیه طول بکشد، در هر یک از الگوریتم‌های زمان‌بندی، تعداد حرکت بازوی هد و زمان کل حرکت بازو را بدست آورید.

روش	ترتیب خواندن سیلندر (چپ به راست)	تعداد کل حرکات بازوی هد (سیلندر)	زمان کل حرکت بازو
1. FIFO :	۳۹, ۱۴, ۲۷, ۱۸, ۴۲, ۲۵	$۱۷+۲۵+۱۳+۹+۲۴+۱۷=۱۰۵ \Rightarrow t_{cyl}=۱۰۵\times۴=۴۲۰ m sec$	
2. SSTF :	۲۵, ۲۷, ۱۸, ۱۴, ۳۹, ۴۲	$۳+۲+۹+۴+۲۵+۳=۴۶ \Rightarrow t_{cyl}=۴۶\times۴=۱۸۴ms$	
3. SCAN – UP	۲۵, ۲۷, ۳۹, ۴۲, ۱۸, ۱۴	$۳+۲+۱۲+۳+۲۴+۴=۴۸ \Rightarrow t_{cyl}=۴۸\times۴=۱۹۲ms$	
4. SCAN – Down :	۱۸, ۱۴, ۲۵, ۲۷, ۳۹, ۴۲	$۴+۴+۱۱+۲+۱۲+۳=۳۶ \Rightarrow t_{cyl}=۳۶\times۴=۱۴۴ms$	
5. C – SCAN – UP	۲۵, ۲۷, ۳۹, ۴۲, ۱۴, ۱۸	$۳+۲+۱۲+۳+۲۸+۴=۵۲ \Rightarrow t_{cyl}=۵۲\times۴=۲۰۸ms$	
6. C – SCAN – Down :	۱۸, ۱۴, ۴۲, ۳۹, ۲۷, ۲۵	$۴+۴+۲۸+۳+۱۲+۲=۵۳ \Rightarrow t_{cyl}=۵۳\times۴=۲۱۲ms$	

در نقشه کلی، سیلندرهای متقاضی مشخص شده‌اند:



در این مسأله الگوریتم آسانسور با بیت پایین (SCAN – DOWN) کارآیی بهتری نسبت به سایرین دارد.

4-1-4- قرار دادن مناسب رکوردها

علاوه بر تکنیک‌های مطرح شده می‌توان براساس ساختار فایل و وضعیت داده‌های ذخیره شده، با اعمال مواردی زمان جستجو را کاهش داد به عنوان نمونه سعی می‌شود جایدهی رکوردها روی دیسک براساس تعداد دسترسی به آن‌ها صورت گیرد تا زمان جستجو کاهش یابد. بدین‌ترتیب رکوردهایی که زیادتر مورد دسترسی قرار می‌گیرند در سیلندرهای میانی و رکوردهایی که کمتر استفاده می‌شوند را در سیلندرهای بیرونی‌تر ذخیره می‌کنند. به عنوان مثال، در محیط عملیاتی دانشگاه، فایل حاوی اطلاعات دانشجویان که مرتباً مورد مراجعه قرار می‌گیرد در سیلندرهای میانی و فایل حاوی اطلاعات پایان‌نامه دانشجویان که کمتر مورد استفاده قرار می‌گیرد در سیلندرهای بیرونی بهتر است ذخیره گردد.

مثال: کدام تکنیک کاهش زمان استوانه‌جویی دیسک سخت‌افزاری نیست؟

1) استفاده از دیسک دارای هد ثابت

2) توزیع فایل روی سیلندرهای هم شماره چند دیسک

3) استفاده از دیسک سریع برای ذخیره کردن رکوردهایی که تعداد دستیابی به آن‌ها بالاست.

4) هر سه صحیح است.

☑ پاسخ: گزینه «2» یعنی توزیع روی چند دیسک

مثال: درخواست خواندن سیلندرهای 14، 41، 24، 19، 36 و 30 از چپ به راست (اول 30) به دیسک داده شده و در ابتدا نوک هد در سیلندر 26 واقع است. تعداد حرکت بازو در حالتی که از سیاست SSTF استفاده شود کدامست؟

1) 39 2) 41 3) 37 4) 26

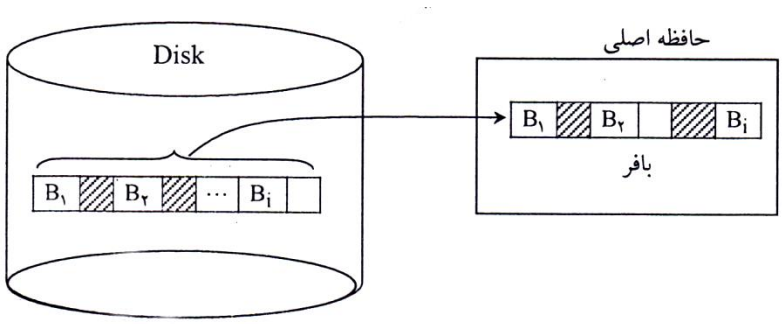
☑ پاسخ: گزینه «1»

39=5+6+16+5+5+2= تعداد حرکت بازو ⟹ 41 , 36, 30 , 14 , 19 , 24 → ترتیب پردازش سیلندرها در این مسأله، SSTF و آسانسور با بیت پائین پاسخ یکسانی دارد.

4-2- کاهش زمان انتقال به کمک Disk Caching

یکی از روش‌های کاهش زمان انتقال (b_{tt})، که موجب کاهش زمان بازیابی بلاک می‌گردد، تکنیک Disk Catching است. در این روش از یک حافظه نهان (Cache)، که درحقیقت بافر بسیار بزرگ است، استفاده می‌شود.

بدین‌ترتیب وقتی سیستم بخواهد بلاک حاوی رکورد موردنظر برنامه کاربر را خوانده و جهت پردازش به بافر در حافظه اصلی منتقل کند، همراه با آن بلوک; بلوک پس از آن را نیز به بافر منتقل می‌کند تا برای پردازش بلوک بعدی مجدداً ارجاع به دیسک صورت نگیرد.



شکل 4-1 تکنیک Disk Caching جهت کاهش زمان انتقال

یعنی برای پردازش n بلوک پشت سرهم، بجای n عمل ورودی ـ خروجی یک بار این کار صورت می‌گیرد.

نکته 1: برای پیاده‌سازی این بافر بزرگ، یک راه ایجاد بافری بزرگ در حافظه اصلی و در منطقه بافرهاست. راه دوم این است که بافر بزرگ یا حافظه Cache بصورت سخت‌افزاری توسط کارخانه سازنده دیسک در کنترلر آن تعبیه شود.

نکته 2: اگر پردازش فایل بصورت تصادفی باشد تکنیک Disk Caching راه‌حل مناسبی نیست. زیرا در این حالت بی‌جهت تعدادی بلاک به بافر منتقل شده که یکی از آن‌ها فقط استفاده (پردازش) می‌شود. پس این تکنیک زمانی مفید است که پردازش فایل ترتیبی باشد.

نکته 3: از آنجا که زمان انتقال (b_{tt}) با نرخ انتقال (t) رابطه معکوس دارد، هرچه نرخ انتقال کمتر از حالت ایده‌آل باشد زمان انتقال افزایش می‌یابد. همان‌طور که در فصل قبل ذکر شد اگر فضای بلااستفاده در بلوک افزایش یابد نرخ انتقال کاهش می‌یابد. لذا روش دیگر کم کردن زمان انتقال این است که فضای بلااستفاده به ازای هر رکورد فایل به حداقل ممکن برسد.

مثال: از حافظه نهان (Cache) با نرخ برخورد 80 درصد و دسترسی 30 میلی‌ثانیه برای کاهش زمان دسترسی اطلاعات استفاده شده است. اگر زمان دسترسی در حافظه 200 میلی‌ثانیه باشد زمان دسترسی مؤثر به اطلاعات به کمک Catche چند ms خواهد شد؟

1) 64 2) 70 3) 90 4) 80

☑ پاسخ: گزینه «2»

زمان دسترسی به کش

$$t_e = \underset{\text{نرخ برخورد}}{h}t_c + (۱-h)(t_C + t_M) = \frac{۸۰}{۱۰۰} \times ۳۰ + \frac{۲۰}{۱۰۰}(۲۳۰) = ۲۴ + ۴۶ = ۷۰ \text{ ms}$$

4-3- کاهش زمان درنگ دورانی

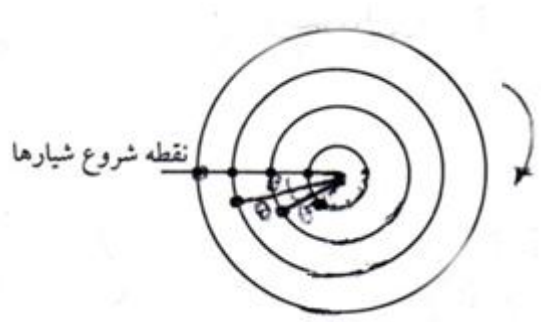
در اینجا هدف این است که با حداقل دیسک نوک هد به بلوک‌های واقع در شیار که باید پردازش شوند رسیده و عمل پردازش بر حسب نوع بافرینگ انجام پذیرد.

در روش سخت‌افزاری (Track Stagger) و نرم‌افزاری (Inter Leaving) برای کاهش زمان درنگ دورانی مطرح شده‌اند که در ادامه به شرح آن‌ها خواهیم پرداخت.

4-3-1- روش Track Stagger:

بطورمعمول نقطه شروع شیارها در یک رویه دیسک چنان است که مکان هندسی آن‌ها، شعاعی از صفحه دیسک است. شکل 4-2 بیانگر همین مطلب است.

$\max(si)$



شکل 4-2 نقطه شروع شیارها در تکنولوژی معمولی

اما در اغلب دیسک‌های امروزی سعی شده مکان هندسی شروع شیارها بجای خط راست یک منحنی تشکیل دهند تا هد از یک شیار به شیار مجاور ورد. بدین‌ترتیب با کاهش زمان درنگ دورانی، نرخ انتقال انبوه نیز بهبود می‌یابد. این روش یک تکنیک سخت‌افزاری است که در آن نقطه شروع هر شیار با شیار بعدی نسبت به مرکز دیسک زاویه تشکیل می‌دهد که از رابطه $\theta = \frac{180}{r} \times Max(Si)$ بدست می‌آید. S_i در این رابطه زمان لازم برای رفتن به سیلندر همجوار بعدی است که برحسب میلی‌ثانیه می‌باشد.

$360°$ $2r$ $\Longrightarrow$ $2r\theta = 360 \times Max(si)$

θ $Max(si)$

1-2-3-4- بافر و اثر آن در سرعت عملیات دیسک

بافر بخشی از حافظه اصلی است که برای ایجاد هماهنگی بین سرعت عملیات حافظه‌های جانبی و پردازنده (CPU) بکار می‌رود. به عبارتی بلاک خوانده شده از حافظه جانبی به داخل آن منتقل شده و یا بلوک داخل در آن روی حافظه جانبی نوشته می‌شود (مجموعه‌ای از بافرها که سیستم برای عملیات ورودی و خروجی درنظر گرفته و ناحیه‌ای پیوسته در حافظه را تشکیل می‌دهد منطقه بافرها (Buffer Pool) نام دارد که می‌تواند به یک یا چند فایل اختصاص داده شود.

درباره بافرها موارد و نکات مهمی مطرح است که برخی از آن‌ها در ادامه شرح داده شده است.

نحوه ایجاد منطقه بافرها

به سه طریق می‌توان منطقه بافرها را ایجاد کرد:

الف) سیستم عامل با باز شدن فایل، اقدام به ایجاد منطقه بافر می‌کند.

ب) با اجرای یک ماکرو که از سیستم‌عامل درخواست ایجاد منطقه بافر می‌نماید.

ج) برنامه‌نویس با ایجاد ناحیه‌ای از حافظه در برنامه و اجرای ماکرویی که محتوای بافر را با فایل‌های تحت پردازش مرتبط می‌کند بافر را ایجاد می‌کند.

انواع بافر از نظر محل تعبیه آن

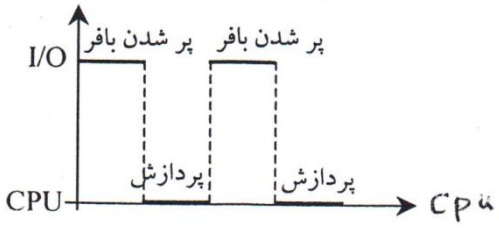
از این دیدگاه بافرها به دو نوع نرم‌افزاری و سخت‌افزاری تقسیم می‌شوند. بافر نرم‌افزاری ناحیه‌ای در حافظه اصلی است که توسط سیستم عامل طبق الگوریتم‌های مدیریت حافظه در اختیار فایل پرداز قرار داده می‌شود. بافر سخت‌افزاری بافری در کنترلر دستگاههای جانبی متصل به کامپیوتر است که به‌ویژه در دستگاه‌های کندتر مثل کارت‌خوان و چاپگر استفاده می‌شود.

انواع بافر از نظر تعداد تخصیص به برنامه

از این دیدگاه سه نوع بافر ساده، مضاعف و چندگانه وجود دارد.

الف) بافرینگ ساده

در این نوع بافرینگ، یک بافر در اختیار برنامه پردازشگر (کاربر) قرار داده می‌شود. در حین پر شدن بافر، پردازنده معطل بوده و لذا کارآیی سیستم پائین می‌آید. البته در محیط چندبرنامه‌ای می‌توان از این زمان برای سایر برنامه‌ها استفاده کرد. شکل زیر بیانگر همین مطلب است.



شکل ۳ـ۴ـ نحوه عملکرد بافرینگ ساده

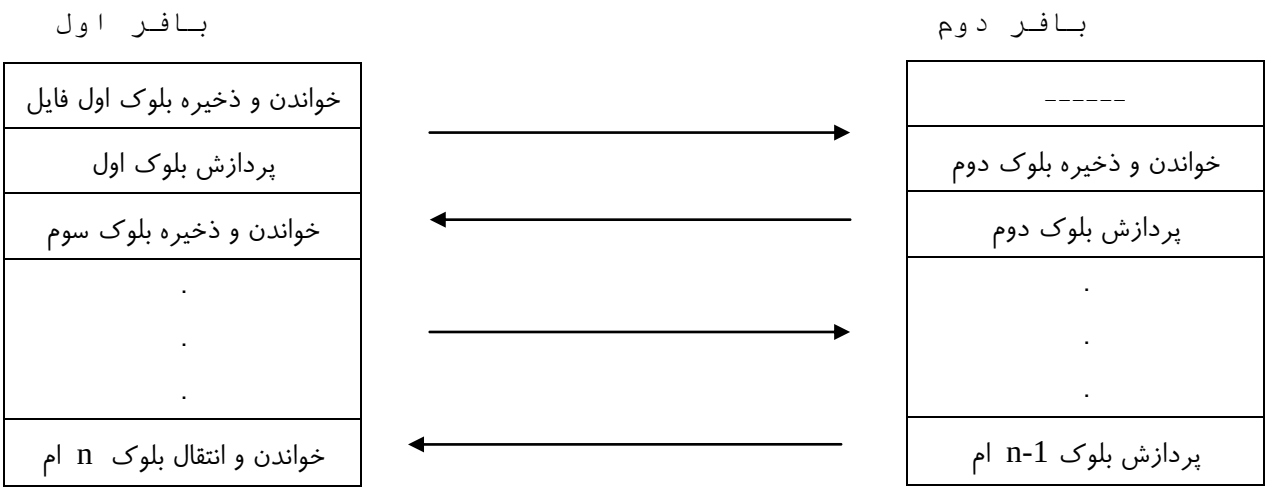
مهم: در این نوع بافرینگ، پس از خواندن بلوک توسط هد و ذخیره آن در بافر، نوک هد و اندازه C_B (زمان پردازش بلاک) از محل انتهای بلاک در شیار عبور می‌کند.

ب) بافرینگ مضاعف

در این روش، به کمک دو بافر عمل خواندن (I/O) بلوک و انتقال آن به یک بافر و پردازش محتوای بافر دوم همروند انجام می‌شود.

در پردازش پی‌درپی (ترتیبی) و انبوه تمام بلاک‌ها لازم است دو بافر در اختیار برنامه قرار گیرد تا سرعت عملیات افزایش یابد.

در مقابل نحوه عملکرد بافر مضاعف آمده است:



نکته: شرط کارآیی بافرینگ مضاعف آن است که زمانی که CPU جهت پردازش محتوای یک بافر (C_B) مصرف می‌کند کمتر از زمانی باشد که پردازنده‌ی ورودی ـ خروجی و کنترلر دیسک برای انتقال بلاک به بافر (با زمان b_{tt} لازم دارند، یعنی $C_B < b_{tt}$ یا

$$C_B < \frac{B+G}{t} \quad \text{یا} \quad C_R < \frac{R+W_R}{t}$$

در این رابطه C_R پردازش یک رکورد، C_B زمان پردازش محتوای بافر (یک بلاک) است.

2-2-3-4- ارزیابی نرخ انتقال انبوه با توجه به نحوه بافرینگ

الف) بافر ساده و پردازش ترتیبی

در این حالت به دلیل آنکه عمل انتقال (I/O) و پردازش همزمان انجام نمی‌شود پس از خواندن و انتقال بلاک اول در حالیکه CPU مشغول پردازش آن است، هد از انتهای بلاک دوم گذشته است و بنابراین برای انتقال آن بلاک باید دور کامل (2r میلی‌ثانیه) صبر کرد و لذا زمان خواندن هر بلوک $۲r+b_{tt}$ میلی‌ثانیه طول می‌کشد. پس نرخ انتقال انبوه در خواندن ترتیبی فایل با شرط $C_B \le ۲r$ خواهد شد:

زمان پردازش بلاک از زمان گردش یک دور دیسک است.

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

تذکر: اگر $2r \le C_B < 4r$ باشد در مخرج کسر t' بجای $2r+b_{tt}$ عبارت $4r+b_{tt}$ درج می‌شود.

ب) بافر ساده و پردازش بدون نظم

منظور از پردازش بدون ترتیب فایل این است که فقط خواندن تمام بندها مهم می‌باشد ولی ترتیب خواندن آنها مهم نیست. در این حالت تمام بلاکهای موجود در یک شیار را میتوان با 2 دور خواند. پس نرخ انتقال انبوه در این حالت با شرط $C_B \le B_{tt}$ برابر است با: $t' = \frac{t_f \times B}{2r}$

ج) بافرینگ مضاعف با وجود شرط کارآیی($C_B \le b_{tt}$) و پردازش دلخواه

همچنانکه در بافر مضاعف گفته شد شرط کارآ بودن بافرینگ مضاعف $C_B < B_{tt}$ یا $C_B < \frac{B+G}{t}$ یا $C_R < \frac{R+W_R}{t}$ است. در این حالت تمام بلاکهای موجود در شیار چه در پردازش ترتیبی و چه در پردازش تصادفی با یک دور چرخش دیسک خوانده می‌شوند و لذا نرخ انتقال انبوه برای بافر مضاعف با وجود شرط کارآیی آن خواهد بود:

$$t' = \frac{T_f \times B}{2r}, CB < \frac{B+G}{T}$$

د) بافرینگ مضاعف با عدم شرط کارآیی و پردازش ترتیبی

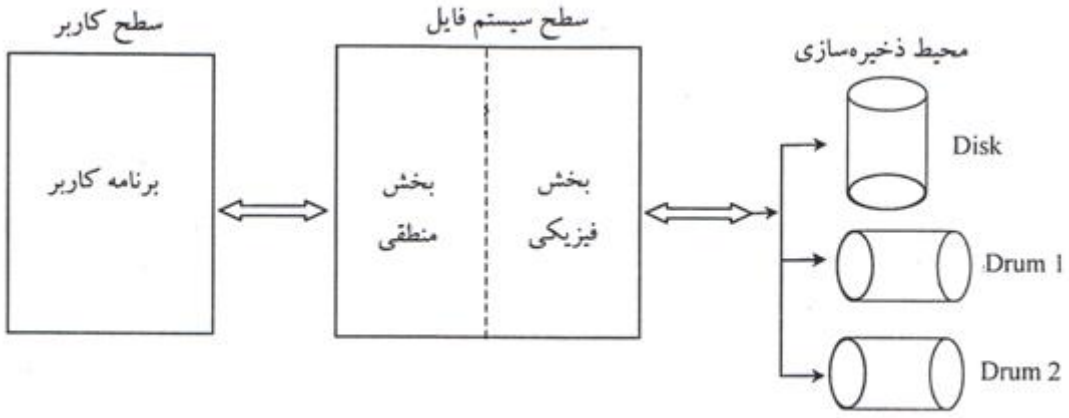
اگر شرط کارا بودن بافر مضاعف وجود نداشته باشد یعنی $C_B \ge b_{tt}$ یا $C_B \ge \frac{B+G}{t}$ یا $C_R \ge \frac{R+W_R}{t}$، در این حالت در هر بافر یک بلاک قرار گرفته و همزمان پردازش می‌شوند. اما وقتی هد به ابتدای بلاک سوم می‌رسد به علت عدم کارآیی هنوز پردازش بافر اول تمام نشده و لذا بلوک سوم نمی‌تواند در بافر قرار گیرد. پس برای هر 2B بایت اطلاعات $۲r + ۲b_{tt}$ زمان لازم است. بدین‌ترتیب نرخ انتقال انبوه برای بافر مضاعف بدون شرط کارآیی بافرینگ مضاعف برابر است با:

$$t' = \frac{2B}{2r + 2b_{tt}} = \frac{2B}{2(r + B_{tt})} = \frac{B}{r + B_{tt}} \qquad \text{با شرط} \quad b_{tt} < C_b < r$$

4-5- سیستم فایل و آدرس‌دهی فایل‌ها

سیستم فایل بخشی از سیستم عامل است که کلیه عملیات مربوط به فایل مانند ایجاد، حذف، کپی و تخصیص فضای حافظه را مدیریت می‌کند. از طرفی آنچه در فایل آدرس‌دهی می‌شود داده‌ی موردنظر (برنامه) کاربر است که در جایی از حافظه جانبی ذخیره شده است. نحوه آدرس‌دهی آن به سطح برخورد با محیط ذخیره‌سازی خارجی بستگی دارد.

بطورکلی سیستم فایل از دو بخش منطقی و فیزیکی (A.M: Access Method) تشکیل شده که جهت دسترسی به داده‌ی ذخیره شده روی محیط ذخیره‌سازی (شامل چند حافظه جانبی) استفاده می‌گردد. در شکل زیر بیانگر همین مطلب است.



شکل ۴ـ۴ـ سطوح مختلف برخورد با محیط ذخیره‌سازی بکمک سیستم فایل

در شکل فوق روال کار چنین است: برنامه پردازشگر فایل (برنامه کاربر) به کمک دستورات یک زبان برنامه‌سازی می‌خوهد به داده‌ای در محیط فیزیکی دسترسی می‌یابد. نرم‌افزار واسط «سیستم فایل» عمل دریافت دستور از برنامه کاربر و محاسبه آدرس فیزیکی محل داده‌ی موردنظر را انجام می‌دهد.

نحوه عملکرد سیستم فایل بدین‌ترتیب است که بخش منطقی درخواست‌های برنامه کاربر مبنی بر باز کردن، بستن، نوشتن یا خواندن فایل را دریافت می‌کند. این بخش از تعداد، نوع و ظرفیت حافظه‌های جانبی موجود در محیط ذخیره‌سازی و همچنین تعداد بلاک‌های فایل مطلع است. درنتیجه کل فضای ذخیره‌سازی را بصورت مجموعه‌ای بلاک می‌بیند که از صفر شماره‌گذاری شده‌اند. به عبارتی بخش منطقی به هر بلاک یک آدرس نسبی بلاک (RBA: Relative Block Address) اختصاص می‌دهد.

اما وظیفه بخش فیزیکی سیستم فایل دستیابی فیزیکی به فایل‌ها در محیط ذخیره‌سازی است. برای این منظور باید درخواست دریافتی (RBA) از بخش منطقی را به فرامینی جهت صدور به کنترلر رسانه تبدیل کند. بخش فیزیکی سیستم فایل به عمل مکان‌یابی (Seek)، خواندن و نوشتن روی حافظه جانبی را انجام داده و از مشخصات فیزیکی هر حافظه جانبی موجود در محیط ذخیره‌سازی مثل تعداد شیار در هر سیلندر یا تعداد بلاک، در هر شیار مطلع است. بدین‌ترتیب سیستم فایل می‌تواند به محل داده‌ی موردنظر برنامه دسترسی پیدا کند.

4-5-1- سطوح آدرس‌دهی:

همانطور که مطرح شد سه سطح برخورد با فایل بصورت کاربری، منطقی و فیزیکی سیستم فایل وجود دارد. لذا سه سطح آدرس‌دهی مختلف نیز مطرح می‌گردد که در ذیل آمده است:

الف) آدرس‌دهی در سطح کاربر بصورت محتوایی، نسبی و سمبلیک می‌باشد

1) آدرس‌دهی محتوایی: در این نوع، کاربر مقدار کلید اصلی رکورد موردنظرش راجهت بازیابی اعلام می‌کند.

2) آدرس‌دهی نسبی: در این حالت، کاربر کل محیط ذخیره‌سازی را بصورت ساختار خطی دیده که هر رکورد شماره مشخصی داشته که از صفر شروع می‌شود. کاربر جهت جستجو آدرس نسبی رکورد موردنظر را می‌دهد.

3) آدرس‌دهی سمبلیک یا نمادی: در این حالت حتی فایل نیز به کمک یک اسم سمبلیک در برنامه پردازشگر مشخص و آدرس‌دهی می‌گردد. به همین ترتیب کاربرد رکورد موردنظرش را نیز آدرس‌دهی می‌کند.

ب) آدرس‌دهی در سطح منطقی سیستم فایل

در این سطح، آدرس‌دهی نسبی استفاده می‌شود ولی آدرس نسبی با آدرس نسبی از دید کاربر، در محدوده یک فایل مشخص صورت می‌پذیرد، تفاوت دارد. همان‌طور که گفته شد بخش منطقی سیستم فایل با توجه به اطلاعاتش درباره فایل‌ها، به کل فضای ذخیره‌سازی را بصورت مجموعه‌ای از بلاک دارای آدرس نسبی $RBA(0 \leq RBA \leq S-1)$ می‌بینند. S تعداد کل بلاک‌های موجود در محیط ذخیره‌سازی است.

بطورکلی بخش منطقی سیستم فایل باید آدرس داده شده توسط برنامه کاربر را به آدرس نسبی بلاک (RBA) تبدیل کند. اگر آدرس در برنامه کاربر (در سطح کاربر) سمبلیک باشد، برای تبدیل فوق از یک جدول Look up Table استفاده می‌شود و اگر آدرس در سطح کاربر نسبی باشد بخش منطقی آن را به RBA تبدیل می‌نماید که جهت این امر از B (طول بلاک)، R (طول رکورد فایل) پارامتر بلاک‌بندی (Brf)، آدرس شروع فایل، تعداد بلاک‌های فایل (b) و تعداد رکوردهای آن (n) استفاده می‌شود.ژ

ج) آدرس‌دهی در سطح فیزیکی سیستم فایل

بخش فیزیکی سیستم فایل آدرس RBA حاصل از بخش منطقی سیستم فایل را به آدرس فیزیکی تبدیل می‌کند. آدرس فیزیکی شامل شماره درایو (Volume)، شماره استوانه، شیار از استوانه، بلاک از شیار و شماره رکورد در حالت بلاک‌بندی نشده است.

✍**مثال**: نشانی‌دهی در سطح منطقی سیستم فایل چگونه است؟

1) آدرس‌دهی نسبی 2) آدرس‌دهی نمادی 3) آدرس‌دهی محتوایی 4) هر سه مورد

☑ پاسخ: گزینه «1» در سطح منطقی سیستم فایل، آدرس‌دهی معمولاً بصورت نسبی است.

✍**مثال**: اگر آدرس در برنامه پردازشگر فایل بصورت ... باشد در بخش منطقی سیستم فایل از جدول نمادها (Look up) برای تبدیل آدرس استفاده می‌شود.

1) آدرس نسبی 2) آدرس نمادی 3) آدرس محتوایی 4) هیچکدام

☑ پاسخ: گزینه «2»

✍**مثال**: بخش فیزیکی سیستم فایل:

1) آدرس نمادی را به RBA تبدیل می‌کند

2) RBA را به آدرس نسبی تبدیل می‌کند.

3) آدرس سمبلیک بلاک را به آدرس فیزیکی تبدیل می‌کند.

4) آدرس نسبی بلاک را به آدرس فیزیکی تبدیل می‌کند

☑ پاسخ: گزینه «4» بخش منطقی سیستم فایل، آدرس‌نسبی را به PBA و بخش فیزیکی نیز RBA را به آدرس فیزیکی تبدیل می‌کند.

2- 5-4- تبدیل آدرس در سیستم فایل

با توجه به مشخصات حافظه‌های جانبی موجود در محیط ذخیره‌سازی و آدرس نسبی بلاک (RBA) و حافظه جانبی حاوی بلاکی که RBA آن مشخص شده یافته و محل آن نسبت به شروع حافظه جانبی محاسبه می‌گردد درنهایت با توجه به مشخصات سخت‌افزاری آن حافظه جانبی، آدرس فیزیکی داده‌ای موردنظر بدست می‌آید.

✍مثال1: در یک محیط ذخیره‌سازی، حافظه‌های جانبی زیر موجود است.

الف) حدود RBA هر حافظه جانبی را بدست آورید.

ب) آدرس فیزیکی داده‌ای دارای 22476= RBA را بدست آورید.

ج) داده‌ای در سیلندر شماره 4، شیار شماره 12 بلوک شماره 5 حافظه xDisk واقع است. آدرس نسبی بلوک (RBA) آن را محاسبه کنید.

تعداد	نوع	تعداد سیلندر در واحد	تعداد شیار در سیلندر	تعداد بلوک در شیار
۲	Drum	۱	۱۲۸	۱۶
۱	Disk 1	۱۰۰	۴۰	۸
۱	Disk 2	۱۰۰	۲۰	۸

☑ پاسخ: ابتدا تعداد بلاک هر حافظه جانبی را بدست می‌آوریم: (تعداد بلاکهای هر شیار x تعداد شیار x تعداد سیلندر = تعداد بلاکها)

Drum=1*128*16=2048
Disk1=100*40*8=3200
Disk2=100*20*8=1600

تعداد کل بلاکها(s)=2*2048+32000+16000=52096

نام حافظه		
Drum1	0	2047
Drum2	2048	2048+2047=4095
Disk1	4096	4096+31999=36095
Disk2	36096	36096+15999=52095

برای بدست آوردن قسمت ب می بایست محل بلاک از شروع Disk1 معلوم شود .

که بدین منظور محل بلاک مورد نظراز ابتدای رسانه RBA_i خواهد بود که از رابطه $RBA_I = RBA_{REC} + RBA_{BOD}$ بدست می آید.

در مرحله بعد باید از ویژگی‌های Disk1 (که داده‌ی دارای شماره‌ی 22476 در آن قرار دارد) برای یافتن محل فیزیکی داده‌ی موردنظر استفاده کرد. چون هر سیلندر Disk1 دارای 40 شیار و هر شیار 8 بلوک دارد. یعنی هر سیلندر 320 بلوک جا می‌گیرد و لذا:

$$C\# = \left[\frac{RBAi \bmod (ttb)}{b}\right] = \left[\frac{18380}{320}\right] = 57$$

برای بدست آوردن شماره شیار میبایست از رابطه زیر استفاده کرد:

$$t\# = \left\lfloor$$

همچنین برای بدست آوردن شماره بلاک از رابطه زیر استفاده میشود:

$$blk\# =$$

18380 | 320
- 18240 | 57 ====> شماره سیلندر
140

140 | 8
-136 | 17 ====> شماره شیار
4 ======> شماره بلاک

داده‌ی موردنظر در بلوک شماره 4 (بلوک پنجم) واقع در شیار شماره 17 (شیار هجدهم) سیلندر شماره‌ی 57 (پنجاه‌و هشتم) حافظه Disk1 واقع است.

ج) عکس مراحل فوق طی می‌شود.

توجه شود هر سیلندر Disk2 دارای 160 بلوک بوده و هر شیار 8 بلوک دارد.

A | 8
5 | 12

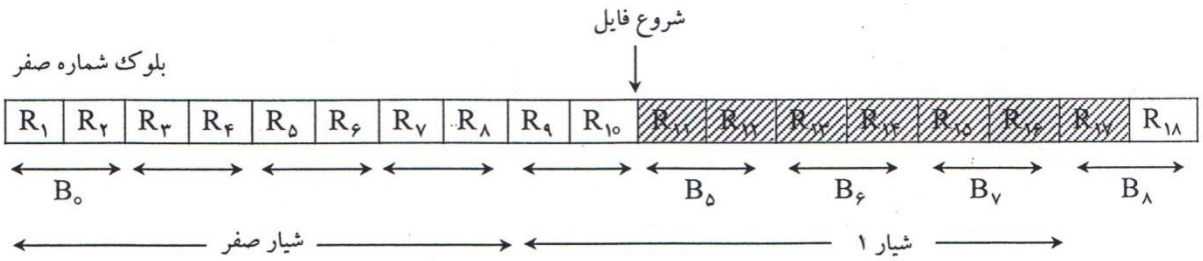
b | 160
101 | 4

A= (8*12)+5=101
B=(4*160)+101=741
RBA (REC)=RBA(XDISK)+RBA
36096+741=36837

مثال2: فایل f شامل7 رکورد منطقی 500 بایتی است که در قالب بلوک‌های 1000 بایت در سطح دیسک ذخیره شده‌اند. از طرفی در هر شیار 4 بلوک جای می‌گیرد و فایل در ناحیه‌ای پیوسته از حافظه ذخیره شده است. اگر آدرس اولین بلوک فایل (RBA شروع فایل) 5 باشد. مطلوبست:

الف) بایت شروع رکورد ششم. ب) RBA حاوی رکورد ششم نسبت به ابتدای فایل.

☑ پاسخ: الف) $\text{Bfr} = \frac{B}{R} = ۲$ هر بلوک 2 رکورد دارد.



در شکل فوق، رکوردهای هاشورخورده مربوط به فایل f می‌باشند.

رکورد ششم فایل f از بایت 2500 آغاز می‌شود.

وع رکورد

(الف) $\Longrightarrow (6-1)\times 500 = 2500$

$BF =$

ب) با توجه به شکل رسم شده و رکورد ششم (رکورد شماره 5) در بلوک شماره 2 (بلوک سوم) یعنی B_2 واقع است و لذا:

$RBA_I =$

ج) باز هم از شکل نتیجه می‌شود رکورد ششم در بلوک شماره 7 یعنی $B_۷$ واقع است.

RBA_{rec}

نکته 1: فایل f حاوی رکوردهای R بایتی در قالب بلوک‌های B بایتی در سطح شیارهای دیسک ذخیره شده‌اند. اگر RBA شروع قابل برابر a و در هر شیار T_f بلوک وجود داشته باشد. داریم:

$(i-1)\times R$ بایت شروع رکورد iام

$RBA = \left\lceil \frac{i-1}{Bfr} \right\rceil = RBAi$ حاوی رکورد i

RBA_{rec} حاوی رکورد iام نسبت به اولین بلوک دیسک RBA=$RBA_{BOD} + RBAi$

نکته 2: (روش دیگر حل مثال 1) اگر تعداد رویه (شماره موجود در هر سیلندر) برابر t و هر شیار b_i بلوک داشته باشد. آدرس فیزیکی یک بلوک به فرم $(C\neq, t\neq, b\neq)$ یعنی (شماره بلوک نسبت به اول آن شیار، شماره شیار نسبت به اول آن سیلندر، شماره سیلندر) از رابطه‌های زیر بدست می‌آید:

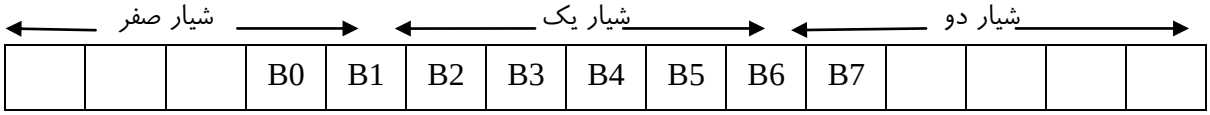
$$C\#\left\lceil \frac{RBA_i}{(t\times b)} \right\rceil \qquad t\#\left\lceil \frac{RBA_i \bmod (t\times b)}{b} \right\rceil, blk\#PBAi \bmod b$$

مثال: هر شیار دیسک دارای 5= بلوک 400 بایتی حاوی رکوردهای 100 بایت است. در صورتی که RBA شروع فایل 3 باشد و رکورد موردنظر رکود 31ام باشد، شماره بلوک موردنظر (#BLK) را بدست آورید.

$$Bfr = \frac{B}{R} = 4 \Rightarrow RBA_i = \left\lfloor \frac{(i-1)}{Bfr} \right\rfloor = \left\lfloor \frac{31-1}{4} \right\rfloor = 7$$

$$BLK \neq= (RBA_i + RBA_{BOF}) \bmod b_i = (۷+۳) \bmod ۵ = ۰$$

از روی شیار نیز چنین می‌شود:



همچنان که دیده می‌شود B_7 در شیار دو (بلوک شماره صفر) واقع است.

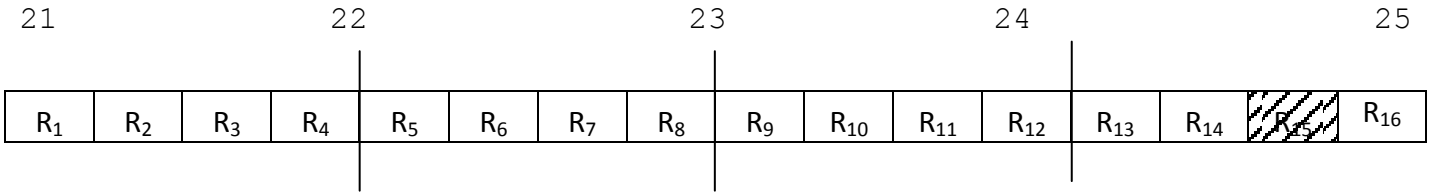
مثال: فایلی دارای رکوردهای منطقی 200 و بلاک‌های 800 بایتی موجود است و RBA شروع فایل 21 می‌باشد. کاربر درخواست خواندن رکورد پانزدهم فایل را نموده است. RBA حاوی این رکورد کدام است؟

1) 24 2) 23 3) 22 4) 25

☑ پاسخ: گزینه «1»

$Bfr\left[\frac{B}{R}\right]=4 \Rightarrow$ نسبت به ابتدای فایل $RBA_i=\left[\frac{i-1}{Bfr}\right]=\left[\frac{15-1}{4}\right]=\left[\frac{14}{4}\right]=3$

RBA_{rec}



مثال)کاربری در یک برنامه به زبان پاسکال نوشته است :

```
Seek (myfile , 8 )
Read (myfile , my  area)
```

اگر بایت B=1000و بایت R=500 و $RBA_{BOF}=10$ باشد RBA_{REC} عبارت است از :

$RBA_I=$

===> $RBA_i=\left\lfloor\frac{(8-1)*500}{1000}\right\rfloor=3$

===> $RBA_{REC}=RBA_{BOF}+RBA_i=10+3=13$

مثال) در یک فایل با مشخصات زیر مطلوب است RBAرکورد چهارم ؟

500= طول رکورد

1000= طول بلاک

RBA (BOF) =8

$RBA=\left[\frac{(I-1)\ R}{B}\right]=\left[\frac{(4-1)*500}{1000}\right]=1$

RBA (REC)=8+1=9

مثال) فایلی با طول منطقی 200 و طول هر بلاک 800 بایت موجود است کاربر در خواست خواندن رکورد سیزدهم فایل را کرده است RBAحاوی این رکورد چه خواهد بود ؟ RBA شروع فایل 21 می باشد .

$$RBA=\left[\frac{(I-1)\ R}{B}\right]=\left[\frac{(13-1)*200}{800}\right]=3$$

RBAنسبی + RBAشروع فایل
21+3=24

مثال) اگر RBA (BOF) =10و R=500 و B=1000 باشد . RBA (REC) دستور Seek (fp1 ,12) کدام است ؟
دستور Seek (fp1 .12) یعنی رکورد دوازدهم (i=12)

$$RBA(rec)=\left[\frac{(I-1)\ R}{B}\right]=\left[\frac{(12-1)*500}{1000}\right]=5$$

RBA_{rec}

مثال) فرض کنید فایلی داریم که روی دیسکی ثابت با مشخصات زیر ذخیره شده است ؟

تعداد سیلندر	تعداد شیار در سیلندر	تعداد بلاک در شیار
300	40	5

اگر ادرس نسبی بلاکی برابر 120111 باشد معین کنید در ادرس فیزیکی بلاک مورد نظر شماره شیار کدام است ؟
(شماره سیلندر و شماره شیار از صفر شروع می گردد)

$$t\#=\left[\frac{RBA_i \bmod(t*b)}{b}\right]=\left[\frac{120111 \bmod(40*5)}{5}\right]=22$$