

فصل چهارم

آشنایی با مفاهیم سیستم فایل

- روش رایج نگهداری اطلاعات ذخیره بر روی رسانه خارجی در واحدهایی به نام فایل
- ایجاد و مدیریت فایلها با سیستم فایل
- سیستم فایل نرم افزاری چند لایه
- جزئیات درونی از دید کاربر پنهان، تنها جنبه‌های بیرونی مورد توجه کاربر
- مفاهیم مقدماتی: فیلد، رکورد، فایل

فیلد - رکورد

• فیلد

- مکان ذخیره‌سازی یک واحد معنادار یا یک اطلاع
- اطلاع : مجموع یک صفت و مقدار آن
 - نام خانوادگی = صفت
 - احمدی = مقدار
- تجزیه‌ناپذیر

• رکورد

- مجموعه‌ای از فیلدها تشکیل رکورد می‌دهند
 - نام دانشجو، شماره دانشجویی، رشته
- واحد پردازش فایل از دید پردازشگر فایل

انواع ساختار فیلد

- فیلدهای با طول مشخص
- فیلدهای محتوی مشخص طولی
- فیلدهای محتوی کاراکتر پایانی یا جدا کننده
- فیلدهای مشخص شده بوسیله کلمه کلیدی

انواع ساختار فیلد

- فیلدهای با طول مشخص
 - دسترسی به فیلدها راحت و سریع
 - مقداری فضا رزرو شده بیهوده مصرف می‌شود
 - برای فیلدهای دارای تنوع، نامناسب (آدرس)

Ames	Mary	911 High St.	Austin	TX78701
Kammermeister	Alan	102 Main St.	Buda	TX78642

انواع ساختار فیلد

- فیلدهای محتوی شناختی طولی
 - فضای اضافی رزرو نشده و بیهوده به هدر نمی‌رود.
 - امکان پرش به فیلدهای بعدی برامتی میسر است .
 - لااقل یک بایت برای هر فیلد اضافه می‌شود

04Ames04Mary12911 High St.06Austin02TX0578701

13Kammermeister04Alan12102 Main St.04Buda02TX0578642

انواع ساختار فیلد

- فیلدهای محتوی کاراکتر پایانی
 - فضای اضافی رزرو نشده و بیهوده به هدر نمی‌رود.
 - محدودیت روی طول فیلد وجود ندارد.
 - یک بایت برای هر فیلد اضافه می‌شود.
 - کاراکتر جداکننده نبایستی در خود فیلد استفاده شود.
 - امکان پرش سریع به فیلدهای بعدی وجود ندارد.
 - کاراکترهای هر فیلد بایستی یک به یک چک شوند.

Ames|Mary|911 High St.|Austin|TX|78701

Kammermeister|Alan|102 Main St.|Buda|TX|78642

انواع ساختار فیلد

- فیلدهای مشخص شده بوسیله کلمه کلیدی
 - هر فیلد بصورت `Keyword=value` مشخص می‌شود.
 - مانند حالت سوم نیاز به کاراکتر پایانی (یا جدا کننده) می‌باشد.
 - ترتیب فیلد ها اهمیتی ندارد.
 - در صورت عدم امتیاج به یک فیلد، مستقیما فیلد بعدی در رکورد ثبت می‌گردد.

```
Last=Ames | First=Mary | Address=911 High St. | City=Austin | State=TX |  
Zip=78701
```

انواع ساختار رکورد

- رکوردهای با طول مشخص.
- رکوردهای تعریف شده بر مسب تعداد فیلد.
- رکوردهای محتوی شاخص طولی.
- رکوردهای محتوی کاراکتر پایانی یا جدا کننده.
- رکوردهای مشخص شده با کمک ایندکس

انواع ساختار رکورد

- رکوردهای با طول مشخص:

- معمولا با فیلدهای بطول مشخص هستند.
- ولی انواع دیگر فیلدها نیز می توانند در چنین رکوردی تعریف شوند.

Ames	Mary	911 High St.	Austin	TX78701
Kammermeister	Alan	102 Main St.	Buda	TX78642

انواع ساختار رکورد

- رکوردهای تعریف شده بر حسب تعداد فیلد:
 - این رکوردها با طول متغیر هستند.
 - معمولا با فیلدهای بطول متغیر تعریف می‌شوند.

```
04Ames04Mary12911 High St.06Austin02TX0578701  
13Kammermeister04Alan12102 Main St.04Buda02TX0578642
```

```
Ames|Mary|911 High St.|Austin|TX|78701  
Kammermeister|Alan|102 Main St.|Buda|TX|78642
```

```
Last=Ames | First=Mary | Address=911 High St. | City=Austin | State=TX |  
Zip=78701
```

انواع ساختار رکورد

- رکوردهای همراه با شاخص طول:
 - در آغاز رکورد حداقل یک یا دو کاراکتر برای شاخص طول لازم است.
 - معمولاً با فیلدهای بطول متغیر تعریف می‌شوند.

4504Ames04Mary12911 High St.06Austin02TX0578701

5213Kammermeister04Alan12102 Main St.04Buda02TX0578642

38Ames|Mary|911 High St.|Austin|TX|78701

45Kammermeister|Alan|102 Main St.|Buda|TX|78642

72Last=Ames|First=Mary|Address=911 High St.|City=Austin|State=TX|
Zip=78701

انواع ساختار رکورد

- رکوردهای محتوی کاراکتر پایانی:
 - کد ASCII پایان رکورد (End-of-Record)
 - یا یک کاراکتر دیگر مثل 'n'
 - روش بسیار متداول برای فایل‌های متنی ساده.

```
04Ames04Mary12911 High St.06Austin02TX0578701 \n
```

```
13Kammermeister04Alan12102 Main St.04Buda02TX0578642 \n
```

```
Ames | Mary | 911 High St. | Austin | TX | 78701 \n
```

```
Kammermeister | Alan | 102 Main St. | Buda | TX | 78642 \n
```

```
Last=Ames | First=Mary | Address=911 High St. | City=Austin | State=TX |  
Zip=78701 \n
```

انواع ساختار رکورد

- رکوردهای تعریف شده به کمک ایندکس:
 - یک فایل جداگانه بنام Index file لازم می باشد.
 - که مای آدرس اولین بایت هر رکورد می باشد.
 - معمولا با فیلدهای بطول متغیر تعریف می شوند.

Index file:

00 39 85 ...

Data file:

Ames|Mary|911 High St.|Austin|TX|78701#Kammermeister|Alan|102 Main
St.|Buda|TX|78642#...

مزایا و معایب انواع ساختار رکورد

- رکوردهای با طول ثابت:

- از نظر دسترسی سریع به هر رکورد بهتر هستند.
- ولی مقدار فضای رزرو شده ممکن است بیهوده بماند.

- رکوردهای با طول متغیر:

- از به هدر رفتن فضای اضافی پیشگیری می کنند.
- ولی دسترسی سریع به هر رکورد مشکل خواهد داشت.
- استفاده از ایندکس امکان دسترسی سریع را می سازد.

رکورد از سه دیدگاه

- رکورد در سطح انتزاعی
 - مفهوم رکورد مستقل از جنبه‌های نمایشی آن
- رکورد در سطح منطقی
 - توسط برنامه‌نویس در برنامه تعریف می‌شود
- رکورد در سطح فیزیکی
 - رکورد به صورتی که در محیط ذخیره‌سازی مانند دیسک قرار می‌گیرد

رکورد فیزیکی

- وقتی رکورد در سطح حافظه جانبی ذخیره می‌شود اطلاعات دیگری به آن اضافه می‌شود
- رکورد ذخیره شده دارای دو بخش داده‌ای و غیرداده‌ای
- بخش غیرداده‌ای توسط سیستم فایل استفاده می‌شود از دید برنامه‌نویس مخفی است
- در بخش غیر داده‌ای اطلاعات زیر ذخیره می‌شود:
 ۱. طول رکورد
 ۲. نوع رکورد
 ۳. اشاره‌گرها
 ۴. پرچم‌ها
 ۵. اطلاعات ویژه بعضی ساختارها

رکورد فیزیکی

۱. طول رکورد

- در صورتی که طول رکورد متخیر باشد، طول آن ذخیره می شود
- رکوردهایی با طول ثابت به این اطلاعات نیازی ندارند

۲. نوع رکورد

- در یک فایل دو یا چند نوع رکورد ذخیره شود
- در این فایل نوع رکورد باید در ابتدای آن مشخص شود

رکورد فیزیکی

۳. اشاره‌گرها

- ارتباط بین رکوردها از دید برنامه‌نویس ساختار منطقی و نحوه ذخیره شدن آن ساختار فیزیکی
- مثلا از دید برنامه‌نویس رکوردها به ترتیب حروف الفبا پردازش شود ولی این رکوردها منطقا مجاور یکدیگرند و هنگام ذخیره شدن بر روی دیسک الزاما به همان ترتیب نخواهند بود
- با استفاده از اشاره‌گرها ارتباط منطقی بین رکوردها پیاده‌سازی می‌شود

ali

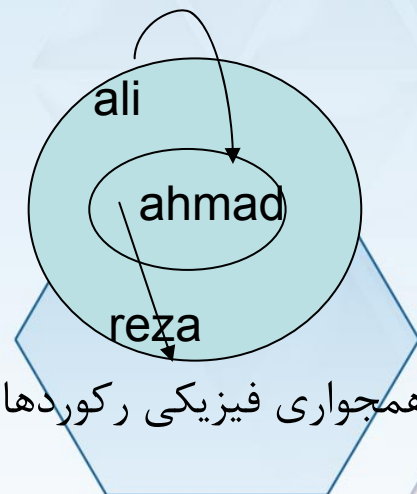
ahmad

reza

همجواری منطقی رکوردها



همجواری فیزیکی رکوردها



ناهمجواری فیزیکی رکوردها

رکورد فیزیکی

۴. پرچم‌ها

- برای نشان دادن عملیاتی که قرار است روی رکوردها انجام شود
- مثلا مذف در دو مرحله فیزیکی و منطقی صورت می‌گیرد
- مثلا در محیط‌های اشتراکی نمونه دسترسی افراد به رکوردها

۵. اطلاعات ویژه بعضی ساختارها

- یک فایل ممکن است از دنباله‌ای از بایتها که بر روی حافظه جانبی ذخیره شده است
- اما در این درس مجموعه‌ای از چند رکورد با ساختار مشخص
- سیستم فایل دو لایه اساسی دارد
 - سیستم فایل منطقی
 - سیستم فایل فیزیکی
- از نظر سیستم فایل منطقی: فایل، مجموعه‌ای از رکوردهای ذخیره‌شده
- از نظر سیستم فایل فیزیکی: فایل، مجموعه‌ای از بلاک‌ها

بلاک و بلاک‌بندی

- بلاک قالبی است با ساختار مشخص و شامل تعدادی رکورد
- با دادن تعدادی بلاک در این قالب بزرگتر را بلاک‌بندی گوئیم
- بلاک واحد رد و بدل اطلاعات بین حافظه جانبی و حافظه اصلی توسط سیستم فایل است
- تعداد رکوردهای درون بلاک را ضریب بلاک‌بندی گویند با B_f نشان داده می‌شود (Blocking Factor)



بلاک و بلاک بندی

زاویه های مهم برای بدست آوردن B_f و تعداد بلاک

$$B_f = \left\lfloor \frac{B}{R} \right\rfloor \quad b = \left\lceil \frac{nR}{B} \right\rceil \quad b = \left\lceil \frac{n}{B_f} \right\rceil$$

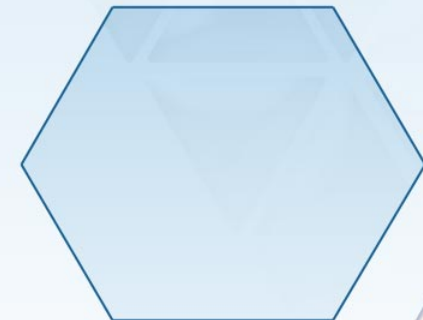
n تعداد رکورد

R طول رکورد

B طول بلاک

b تعداد بلاک

B_f فاکتور بلاک بندی



بلاک و بلاک بندی

- مثال: فرض می‌کنیم در فایلی ۱۰۰۰۰۰۰ رکورد با طول هر یک ۲۵۰ بایت موجود باشد. اگر طول هر بلاک ۲۰۰۰ بایت باشد، فاکتور بلاک بندی و تعداد بلاک را بدست آورید؟

$$B_f = \left\lfloor \frac{B}{R} \right\rfloor = \left\lfloor \frac{2000}{250} \right\rfloor = 8$$

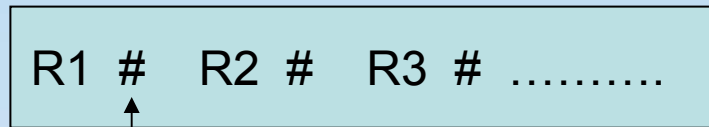
$$b = \left\lceil \frac{n}{B_f} \right\rceil = \left\lceil \frac{1000000}{8} \right\rceil = 125000$$

تکنیک‌های تعیین محدوده رکورد در بلاک

- رکوردها با طول ثابت : یکبار طول رکورد در سیستم فایل ذخیره می‌شود
- رکوردها با طول متغیر: باید محدوده رکورد در بلاک مشخص شود که از روشهای زیر استفاده می‌شود
 - درج نشانگر پایان رکورد
 - درج طول آن در بخش غیرداده‌ای
 - ایجاد جدول مکان‌نما

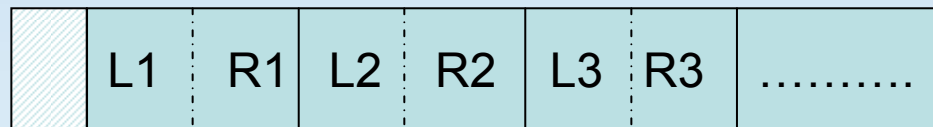
تکنیک‌های تعیین محدوده رکورد در بلاک

- درج علامت در پایان رکورد

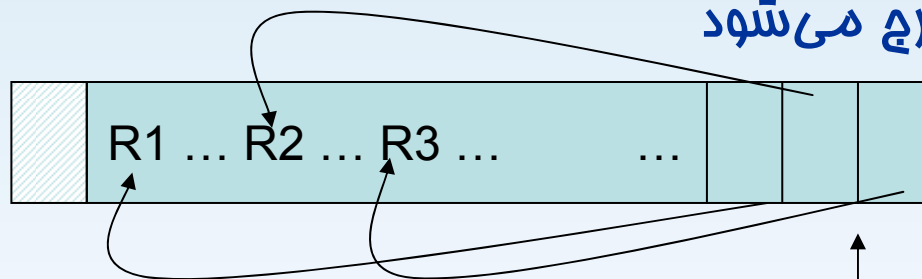


نشانه‌گر پایان رکورد

- درج طول رکورد در ابتدای آن



- ایجاد جدول مکان‌نما: آدرس نسبی پایان رکورد در بلاک نسبت به آغاز بلاک، در جدول درج می‌شود



جدول مکان‌نما

مزایا و معایب بلاک بندی

- مزایای بلاک بندی

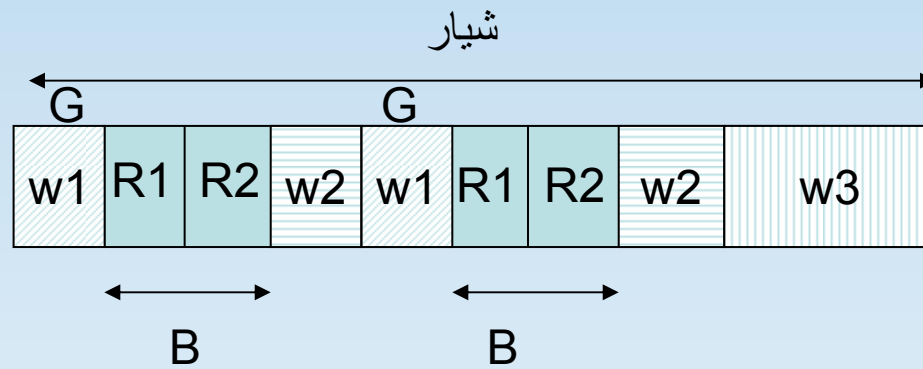
- کاهش دفعات ورودی/خروجی
- صرفه جویی در مصرف رسانه ذخیره سازی از طریق کاهش گپها (زیرا اگر بلاک بندی نکنیم گپهای بین رکوردی داریم که اندازه فایل را افزایش می دهد)

- معایب بلاک بندی

- کار نرم افزاری بیشتر برای بلاک بندی و بلاک گشایی (عکس عمل بلاک بندی که رکوردهای منطقی در اختیار کاربر قرار می گیرد)
- مصرف بیشتر حافظه اصلی به علت لزوم بافرینگ
- بالا رفتن احتمال اشتباه در مبادله اطلاعات به خاطر افزایش دادهای که منتقل می شود

روشهای مختلف بلاک بندی

- بلاک بندی با طول ثابت و یکپارچه



$$w_B = w_1 + w_2 + w_3$$

$$w_B = G + \frac{R}{2} + \frac{w_3}{T_F}$$

w_1 حافظه هرز ناشی از گپ

w_2 حافظه هرز ناشی از ننگجیدن رکورد آخر در بلاک

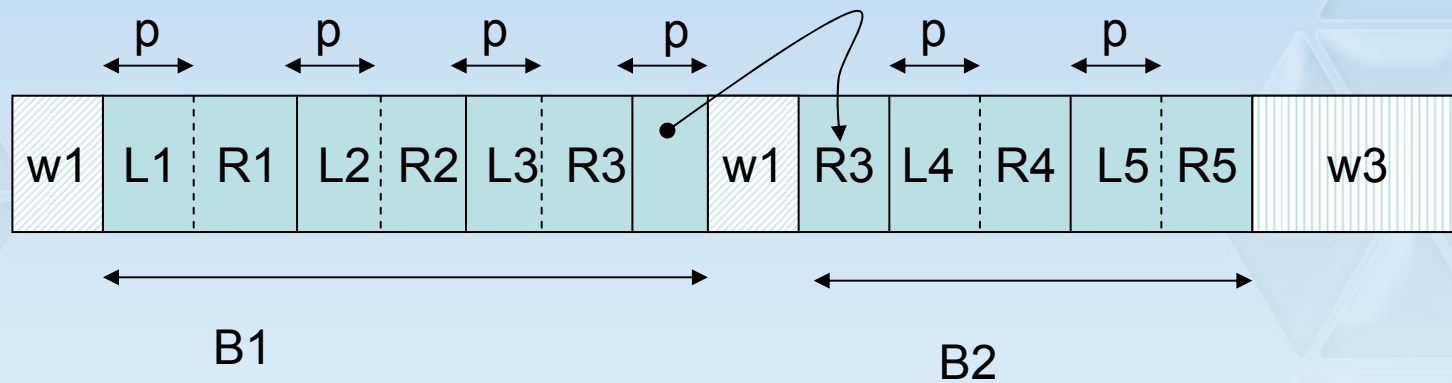
w_3 حافظه هرز ناشی از ننگجیدن بلاک آخر در شمار

w_B میزان حافظه هرز بازای هر بلاک

T_f تعداد بلاکها در هر شمار

روشهای مختلف بلاک بندی

- بلاک بندی با طول متغیر و دوپاره

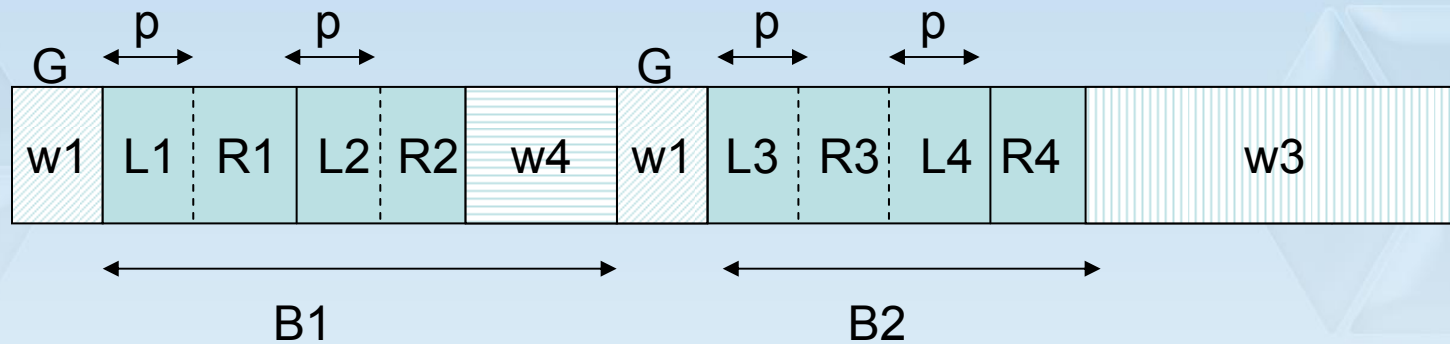


$$wB = w1 + w2 + w3$$

$$wB = G + B_f * p + p + \frac{w3}{T_f}$$

روشهای مختلف بلاک بندی

- بلاک بندی رکوردها با طول متغیر و یکپارچه



$$wB = w1 + w2 + w3 + w4$$

$$wB = G + B_f * p + \frac{R}{2} + \frac{W3}{T_f}$$

لوکالیتی رکوردها

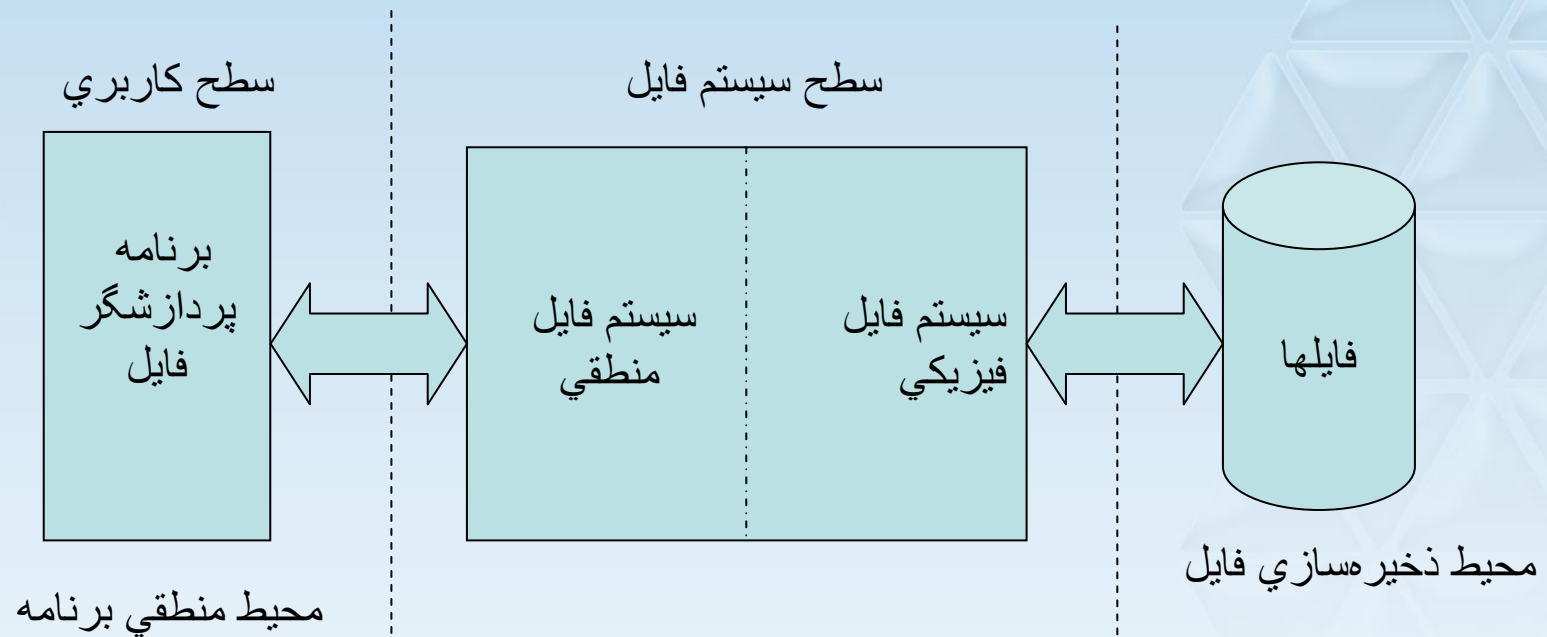
- اگر دو رکورد $R1$ و $R2$ منطقا همجوار باشند و $R1$ واكشی شود در اینصورت
 - $R1$ رکورد فعلی
 - $R2$ رکورد منطقا بعدی
- لوکالیتی : میزان همسایگی رکورد منطقا بعدی نسبت به رکورد فعلی روی حافظه جانبی

لوکالیتی رکوردها

- درجات لوکالیتی از قوی به ضعیف

- رکورد بعدی در همان بلاکی که رکورد جاری است و بلاک در بافر قرار دارد، عمل I/O نداریم و کمترین زمان برای بدست آوردن رکورد بعدی مصرف می‌شود.
- رکورد بعدی در بلاک بلافاصله بعدی، بلاک حاوی رکورد فعلی از همان استوانه باشد. عمل I/O داریم، اما $r=0, s=0$
- رکورد بعدی در همان استوانه رکورد فعلی $s=0$ و $r>0$
- رکورد بعدی در استوانه همجوار $r>0$ و $s>0$
- رکورد بعدی در استوانه دیگر (آدرس آن از رکورد فعلی قابل محاسبه است)

سطوح آدرس دهی



سطوح آدرس دهی

- سه سطح برخورد با فایل
 - سطح برنامه کاربر
 - سطح سیستم فایل منطقی
 - سطح سیستم فایل فیزیکی
- بنابراین سه سطح آدرس دهی
 - آدرس دهی در سطح برنامه فایل پرداز
 - آدرس دهی در سطح سیستم فایل منطقی
 - آدرس دهی در سطح سیستم فایل فیزیکی

سطوح آدرس دهی - آدرس دهی در سطح برنامه کاربر

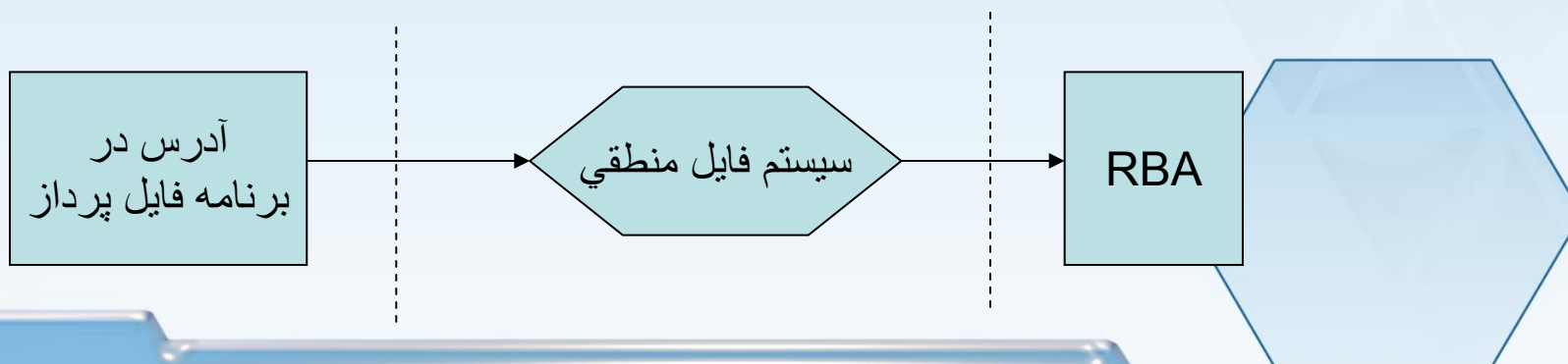
- آدرس دهی محتوایی یا مقداری: کاربر مقدار یک یا چند صفت خاصه (فیلد) را برای جستجو می دهد. این صفت خاصه ممکن است کلید باشد یا نباشد.
 - مثال: رکوردهایی که نام آنها علی است و در مقطع کاردانی تمصیل می کنند را بیابید
- آدرس دهی نسبی: کاربر آدرس نسبی رکورد را می دهد. کاربر فایل را به صورت یک ساختار خطی می بیند که هر رکورد یک شماره یکتا دارد.
 - مشخصات رکورد شماره ۱۰ را نمایش بده
- آدرس دهی نمادین: در این حالت رکورد دلخواه توسط یک نام مشخص و آدرس دهی می شود
 - مشخصات رکورد مدیر گروه را بده

سطوح آدرس دهی - آدرس دهی در سطح سیستم فایل

- بخشی از سیستم عامل که با فایلها سروکار دارد، سیستم فایل نامیده می شود که شامل دو بخش زیر می باشد:
 - بخش منطقی: درخواستهای برنامه کاربر را دریافت و انجام می دهد. این درخواستها شامل: بارکردن، خواندن، نوشتن و بسته شدن فایل
 - بخش فیزیکی: به طور مستقیم با فایل های موجود در حافظه جانبی دسترسی دارد. دستورات وارد شده از قسمت منطقی سیستم فایل را به دستوراتی جهت صدور به رسانه ذخیره سازی تبدیل می کند. سه عمل اصلی این بخش: مکان یابی، خواندن و نوشتن

سطوح آدرس دهی - آدرس دهی در سطح منطقی سیستم فایل

- سیستم فایل منطقی کل فضای ذخیره سازی را به صورت آرایه ای از بلاکها می بیند
- هر بلاک یک شماره یکتا دارد که به آن آدرس نسبی بلاک یا RBA Relative Block Address گفته می شود
- RBA از صفر شروع می شود
- سیستم فایل منطقی آدرس تولید شده در سطح برنامه را دریافت کرده و آن را به آدرس RBA تبدیل می کند، سپس این RBA را به سمت بخش فیزیکی می فرستد.



سطوح آدرس دهی - آدرس دهی در سطح منطقی سیستم فایل

- سیستم فایل منطقی، با داشتن تعداد، نوع و ظرفیت هر یک از رسانه‌های محیط فیزیکی، طیف مقادیر RBA را برای هر رسانه و نیز در کل فضای ذخیره‌سازی مشخص می‌کند.
- فرض کنید دو دیسک D1 و D2 با مشخصات زیر داریم :

شماره دیسک	تعداد استوانه	تعداد شیار در استوانه	تعداد بلاک در شیار
D1	c1	t1	b1
D2	c2	t2	b2

ظرفیت d1 به بلاک

$$S1 = C1 * t1 * b1$$

$$0 \leq RBA_{D1} \leq S1 - 1$$

ظرفیت d2 به بلاک

$$S2 = C2 * t2 * b2$$

$$S1 \leq RBA_{D2} \leq S1 + S2 - 1$$

طیف مقادیر RBA در فضای ذخیره‌سازی

$$0 \leq RBA \leq S1 + S2 - 1$$

سطوح آدرس دهی - آدرس دهی در سطح منطقی سیستم فایل

- اگر آدرس داده شده در برنامه آدرس نسبی رکورد باشد، بخش منطقی باید آن را به RBA تبدیل کند. برای این منظور به اطلاعات زیر نیازمندیم:

- شماره رکورد
- اندازه رکورد
- اندازه بلاک
- آدرس شروع فایل

$$\text{BYTEOFFSET}_{\text{REC}} = (i-1) * R$$

افست رکورد i ام نسبت به اول فایل

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

آدرس نسبی بلاک حاوی رکورد مورد نظر کاربر نسبت به آغاز فایل

$$\text{RBA}_{\text{REC}} = \text{RBA}_{\text{BOF}} + rba_{\text{REC}}$$

RBA رکورد مورد نظر نسبت به ابتدای دیسک

سطوح آدرس دهی - آدرس دهی در سطح منطقی سیستم فایل

- مثال: کاربری در یک برنامه به زبان پاسکال نوشته شده است دستور زیر را دارد، RBA این رکورد نسبت به آغاز دیسک با توجه به فرضیات زیر بدست آورید؟

```
SEEK(MYFILE,8);
```

$B=1000, R=500, RBA_{BOF}=10$

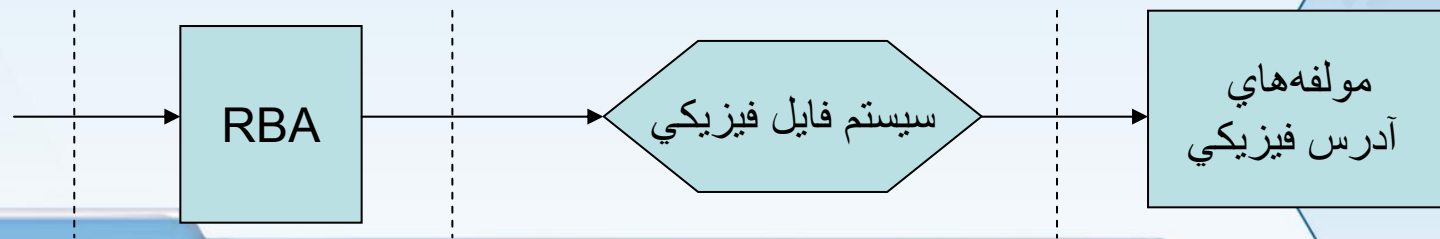
```
READ(MYFILE,MYAREA);
```

$$rba = \left\lfloor \frac{7 * 500}{1000} \right\rfloor = 3$$

$$RBA_{REC} = 10 + 3 = 13$$

سطوح آدرس دهی - آدرس دهی در سطح فیزیکی سیستم فایل

- در این سطح ، باید آدرس فیزیکی داده شود، یعنی آدرسی که مکان داده مورد نظر را در محیط فیزیکی ذخیره سازی را مشخص کند.
- اگر رسانه دیسک باشد اجزای زیر را داریم
 - شماره درایو
 - شماره استوانه
 - شماره شیار از استوانه
 - شماره بلاک از شیار
- مکان یابی و دستیابی فیزیکی به "داده مورد نظر" در حافظه خارجی باید با استفاده از همین آدرس فیزیکی صورت پذیرد، بنابراین سیستم فایل فیزیکی باید RBA را به آدرس فیزیکی تبدیل کند.



سطوح آدرس دهی - آدرس دهی در سطح فیزیکی سیستم فایل

- برای تبدیل RBA به آدرس فیزیکی به صورت زیر عمل می‌کنیم
 - مناسبه شماره درایو : این شماره با مقایسه بلاک حاوی رکورد با کران بالا مقادیر RBA هر رسانه بدست می‌آید

If $0 < RBA \leq S1-1$ then $DR\#=1$

If $s1 < RBA \leq s1+s2-1$ then $DR\#=2$

⋮

- مناسبه شماره استوانه

$$cly\# = \left\lfloor \frac{RBA_{REC} - RBA_{Beginofdevice}}{t_i * b_i} \right\rfloor$$

b_i تعداد بلاک در هر شیار

t_i تعداد شیار در هر سیلندر

$t_i * b_i$ ظرفیت استوانه دیسک D_i

سطوح آدرس دهی - آدرس دهی در سطح فیزیکی سیستم فایل

• محاسبه شماره شیار

$$trk \# = \left\lfloor \frac{(RBA_{REC} - RBA_{BeginofDevice}) \bmod (t_i * b_i)}{b_i} \right\rfloor$$

• محاسبه شماره بلاک

$$blk \# (RBA_{REC} - RBA_{BeginofDevice}) \bmod b_i$$

سطوح آدرس دهی - آدرس دهی در سطح فیزیکی سیستم فایل

- مثال: دیسکی داریم که هر شیار آن شامل ۴ بلاک و هر سیلندر آن ۲ شیار دارد. آدرس فیزیکی بلاکی با $RBA=17$ را بدست آورید؟

$$cyl\# = \left\lfloor \frac{RBA}{t * b} \right\rfloor = \left\lfloor \frac{17}{2 * 4} \right\rfloor = 2$$

$$trk\# = \left\lfloor \frac{RBA \bmod (t * b)}{b} \right\rfloor = \left\lfloor \frac{17 \bmod (2 * 4)}{4} \right\rfloor = 0$$

$$blk\# = RBA \bmod b = 17 \bmod 4 = 1$$

سطوح آدرس دهی - مثال

• مثال: الف- بر اساس جدول زیر RBA در چه محدوده‌ای قرار دارد؟

نوع رسانه	تعداد	تعداد استوانه در واحد	تعداد شیار در استوانه	تعداد بلاک در شیار
طبله	۲	۱	۱۲۸	۱۶
دیسک دائم	۲,۵	۴۰۰	۲۰	۸
دیسک قابل حمل	۳	۲۰۰	۲۰	۱۴

سطوح آدرس دهی - مثال

- ب- بر اساس جدول، $RBA=80000$ در کدام درایو قرار دارد؟
- ج- بر اساس جدول، $RBA=80000$ در کدام استوانه قرار دارد؟
- د- بر اساس جدول، $RBA=80000$ در کدام شیار استوانه قرار دارد؟
- ه- بر اساس جدول، $RBA=80000$ در کدام بلاک شیار قرار دارد؟

سطوح آدرس دهی - مثال

• الف -

$$s1 = 2 * 1 * 128 * 16 = 4096$$

$$s2 = 2.5 * 400 * 20 * 8 = 160000$$

$$s3 = 3 * 200 * 20 * 4 = 48000$$

$$0 \leq RBA < s1 + s2 + s3 - 1$$

$$0 \leq RBA < 4096 + 160000 + 48000 - 1$$

$$0 \leq RBA \leq 212096 - 1$$

سطوح آدرس دهی - مثال

ب-

0—2047	2048----4095	4096--- 68095	68096----132095	132096 -----
طبقه اول	طبقه دوم	دیسک دائم اول	دیسک دائم دوم	دیسک قابل حمل اول



RBA=80000

سطوح آدرس دهی - مثال

۵-

$$cly\# = \left\lfloor \frac{RBA_{REC} - RBA_{Beginofdevice}}{t_i * b_i} \right\rfloor =$$
$$\left\lfloor \frac{80000 - 68096}{20 * 8} \right\rfloor = \left\lfloor \frac{11904}{160} \right\rfloor = 74$$

سطوح آدرس دهی - مثال

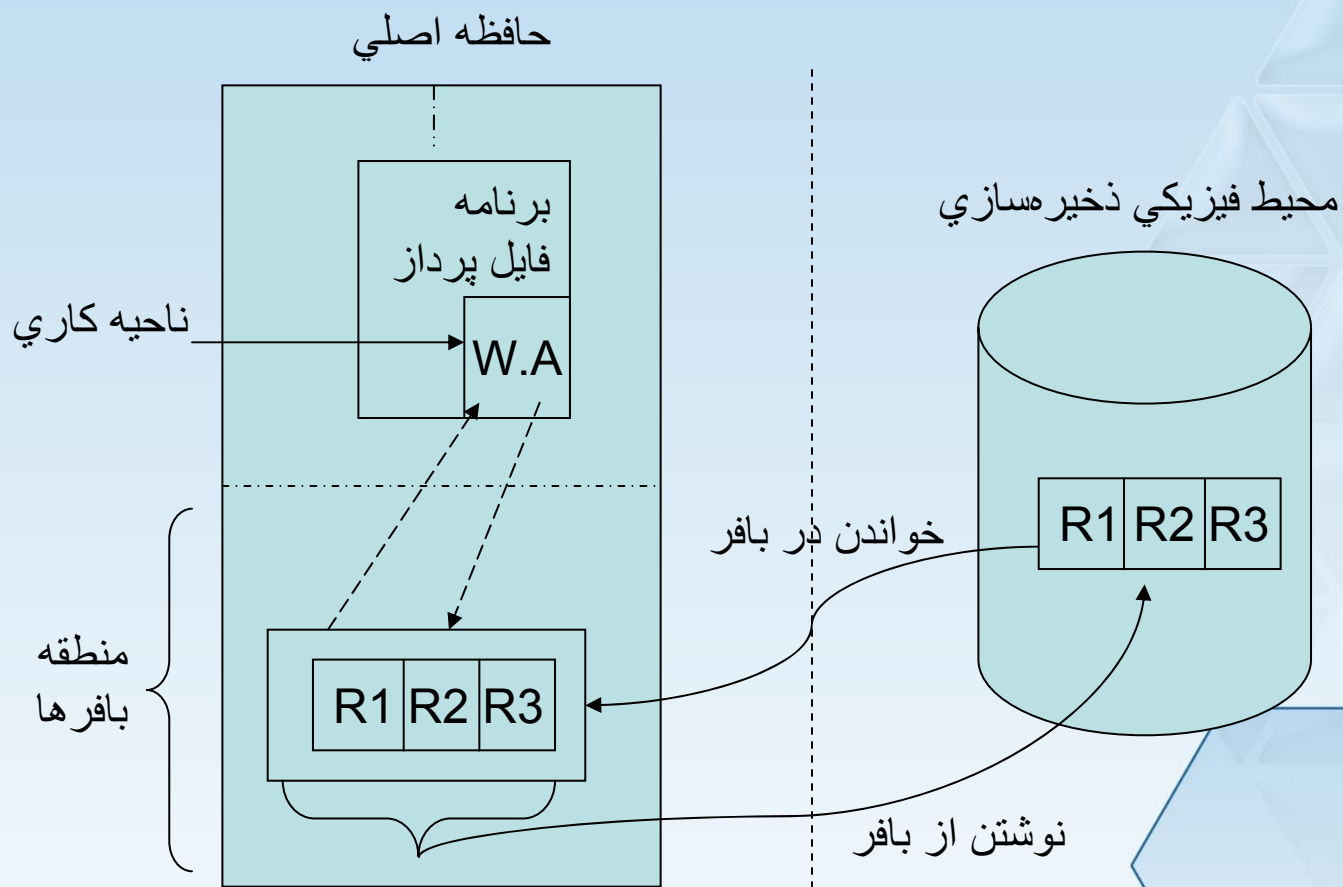
$$trk \# = \left\lfloor \frac{(RBA_{REC} - RBA_{BeginofDevice}) \bmod (t_i * b_i)}{b_i} \right\rfloor =$$

$$\left\lfloor \frac{11904 \bmod 160}{8} \right\rfloor = \left\lfloor \frac{64}{8} \right\rfloor = 8$$

$$blk \# (RBA_{REC} - RBA_{BeginofDevice}) \bmod b_i$$

$$blk \# = 80000 - 68096 \bmod 8 = 0$$

آشنایی با بافر و بافرینگ



آشنایی با بافر و بافرینگ

- بافر نامیه‌ای است واسط بین عملیات ورودی و خروجی
- در این نامیه حداقل یک رکورد (در حالت فایل بلاک‌بندی نشده) و یا حداقل یک بلاک در حالت فایل بلاک‌بندی شده
- برای ایجاد هماهنگی بین عملیات پردازنده ورودی/خروجی و واحد پردازش مرکزی
- در سیستمه فایل، بافر معمولاً از منطقه‌ای از حافظه اصلی به برنامه فایل پردازش تخصیص داده می‌شود که به آن منطقه بافرها می‌گویند

بافر از نظر محل ایجاد

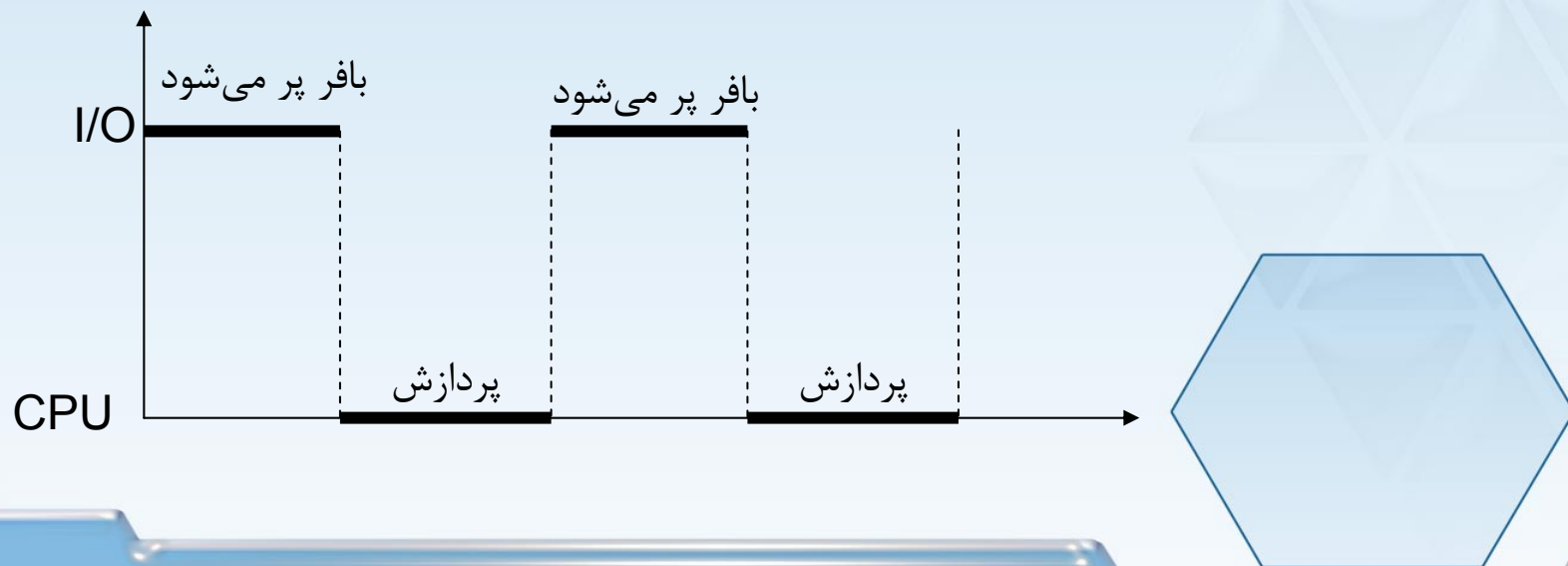
- دو نوع بافر وجود دارد
 - بافر سخت افزاری
 - بافر نرم افزاری
- بافر سخت افزاری:
 - بافری است که در بعضی دستگاههای ذخیره سازی بویژه دستگاههای کندتر مانند چاپگر مورد استفاده قرار می گیرد.
 - این گونه بافرها معمولاً تعداد کمی کاراکتر را در خود جای می دهند و پس از آنکه در مبادله داده ها پر شود، کانال به آن اختصاص داده می شود تا از محیط برون ماشینی به محیط درون ماشینی منتقل شود
- بافر نرم افزاری:
 - نامیه ای است در حافظه اصلی که توسط سیستم عامل ، طبق الگوریتم های تخصیص بافر در مدیریت حافظه، در اختیار برنامه های فایل پرداز قرار می گیرد

انواع بافرینگ

- از نظر تعداد بافرهایی که به عملیات ورودی/خروجی برنامه فایل پرداز تخصیص می‌یابد انواع زیر وجود دارد:
 - بافرینگ ساده
 - بافرینگ مضاعف
 - بافرینگ چندگانه

بافرینگ ساده

- یک بافر در اختیار برنامه فایل پرداز قرار داده می‌شود.
- زمان انتظار واحد پردازش مرکزی افزایش می‌یابد و در نتیجه زمان اجرای برنامه افزایش خواهد یافت
- در هنگامی که بافر در حال پر شدن است CPU نمی‌تواند از محتویات بافر استفاده کند و باید منتظر و بیکار باقی بماند



بافرینگ مضاعف

- می‌توان هنگام خواندن یک بلاک و انتقال آن به بافر، محتوی بافر دیگر را پردازش کرد
- به عبارت دیگر به هنگام انتقال یک بلاک به بافر، بافر دیگری را پر و پردازش کرد
- باید زمانی که برای پردازش محتوی یک بلاک مصرف می‌شود کمتر از زمان انتقال یک بلاک به بافر دیگر باشد
- به عبارت دیگر برای کارایی بافر مضاعف باید رابطه زیر برقرار باشد

$$C_B \leq b_{tt}$$

زمان انتقال یک بلاک $b_{tt} = \frac{B}{t}$

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

CB زمان پردازش یک بلاک

CR زمان پردازش یک رکورد

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

t نرخ انتقال رسانه

بافرینگ چندگانه

- با استفاده از بافرینگ چندگانه هنگام پردازش پی در پی و انبوه فایل می‌توان بلاکهای فایل را از پیش خواند و در بافر ذخیره کرد
- بنابراین رکورد بعدی همیشه در بافر وجود دارد
- به این تکنیک صف‌بندی می‌گویند زیرا سیستم با پیش‌دستی در عملیات I/O، بافرها را از قبل برای CPU آماده می‌سازد
- این نوع عملیات اغلب با صف چرخشی پیاده‌سازی می‌شود به همین دلیل به آن بافر چرخشی نیز می‌گویند