

## ذخیره و بازیابی داده ها

### مدیریت بافر – Buffer Management

احمد خادم زاده  
[khademzaeh@iust.ac.ir](mailto:khademzaeh@iust.ac.ir)  
 دانشگاه آزاد اسلامی واحد مشهد – زمستان ۱۳۸۲

## نوشتن یک بایت روی دیسک

- فرض کنید که در برنامه خود از دستور `<< outfile` استفاده کرده ایم
- این دستور مدیر فایل که قسمتی از سیستم عامل می باشد را برای انجام یک عمل ورودی خروجی فراخوانی می کند.
- مدیر فایل ، عمل نوشتن کاراکتر را انجام می دهد.
- قسمتهای سخت افزاری و نرم افزار که برای انجام یک عمل ورودی/خروجی درگیر می شوند عبارتند از :
  - برنامه کاربردی
  - قسمت مدیر فایل سیستم عامل
  - پردازنده اعمال ورودی خروجی
  - کنترلر دیسک

[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

## نوشتن یک بایت روی دیسک

- برنامه کاربردی :
- درخواست عمل ورودی خروجی
- مدیر فایل سیستم عامل :
- وضعیت تمام فایلهای باز را می داند.
- سکتور مناسب که بایت درخواست شده در آن قرار دارد را به بافر می آورد.
- بایت را در بافر می نویسد
- به پردازنده ورودی خروجی دستور می دهد تا داده ها را در مکان مناسب آنها در دیسک بنویسد
- بافر یک کپی از قسمتی از اطلاعات یک دیسک می باشد.

[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

## نوشتن یک بایت روی دیسک

- پردازنده ورودی خروجی (I/O Processor)
- یک تراشه مستقل از CPU جهت انجام و پردازش اعمال ورودی و خروجی می باشد.
- زمان مناسبی را که دیسک آماده دریافت داده می باشد ، انتخاب می کند و داده ها را با فرمت و شکل مناسب برای دیسک قرار می دهد.
- داده ها را به کنترلر دیسک ارسال می کند.

[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

## نوشتن یک بایت روی دیسک

- کنترلر دیسک
- تراشه مستقل روی برد دیسک می باشد.
- دستور حرکت هد خواندن و نوشتن به مکان مناسب را می دهد.
- در زمانی که سکتور مناسب زیر هد خواندن و نوشتن قرار گرفت ، داده ها را می نویسد.
- شکل ۲۱-۳ صفحه ۸۴ کتاب.

[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

## بافرینگ

- برنامه کاربر می خواهد که در هر لحظه یک رکورد را بخواند و یا بنویسد.
- اما در هر دسترسی به حافظه جانبی یک بلاک از داده ها خوانده و یا نوشته می شود.
- راه حل چیست ؟
- کارکردن با قسمت بزرگی از داده ها در حافظه اصلی ، به منظور کاهش تعداد دسترسی به حافظه جانبی را بافرینگ می گویند.

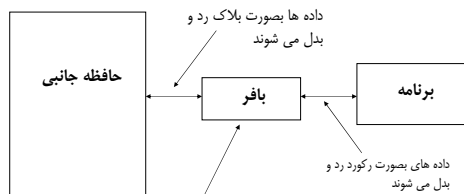
[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

## بافر ورودی/خروجی سیستم

- سیستم عامل از بافر به عنوان یک واسطه بین حافظه جانبی و برنامه کاربر استفاده می کند.
- نیاز به ترجمه و تبدیل از رکورد به بلاک و بالعکس نیز می باشد. این عمل نیز توسط وی انجام می گیرد.
- کل پروسه ای که توسط سیستم عامل جهت واسطه قرار دادن بافر ، صورت می گیرد ، از دید کاربر مخفی است و کاربر فکر می کند که یک رکورد را از حافظه جانبی می خواند و یا یک رکورد را در آن می نویسد.

khademzadeh@iust.ac.ir

## بافر ورودی/خروجی سیستم



حافظه موقتی در Ram برای یک بلاک از داده ها

khademzadeh@iust.ac.ir

## تنگنای بافر

- برنامه زیر را که کاراکترهای یک فایل را خوانده و در فایل دیگری می نویسد را در نظر بگیرید.

```
while (1) {
    infile >> ch;
    if (infile.fail()) break;
    outfile << ch;
}
```

- اگر سیستم عامل فقط از یک بافر استفاده کند چه اتفاقی می افتد؟
- اگر سیستم عامل برای این منظور از دو بافر استفاده کند چطور؟
- اگر اندازه سکتور ها ۲۵۶ بایت باشد ، تفاوت بین حالت تک بافره و دو بافره ، چند عمل اضافی انتقال بین بافر و حافظه جانبی خواهد بود؟

khademzadeh@iust.ac.ir

## مدیریت بافر

- آنچه که ما در مورد آن صحبت می کنیم ، بافرهای ورودی و خروجی سیستم می باشد و کنترل این بافرها در اختیار برنامه کاربردی نمی باشد. این بافرها توسط سیستم عامل مدیریت می شوند.
- ممکن است برنامه کاربردی ما بافری را برای خود پیاده سازی کرده و از آن استفاده می کند.
- بعضی مثال یک متغیر یا یک رکورد که در حافظه اصلی قرار دارند و در زمان لازم در دیسک نوشته می شوند.

khademzadeh@iust.ac.ir

## استراتژی های بافرینگ – بافرینگ چندگانه

### بافر دهی چند گانه

- بجای استفاده از دو بافر از تعداد دلخواهی بافر جهت کمک به همپوشانی اعمال I/O و پردازش استفاده می شود.
- استخر بافر (Buffer Pooling)**
- استخری از بافرها موجود می باشد.
- در زمان درخواست استفاده از یک سکتور خاص ، سیستم عامل ابتدا به دنبال سکتور مورد نظر در یکی از بافرها می گردد.
- اگر سکتور مورد نظر در بافرها نبود ، این سکتور توسط سیستم عامل به یکی از بافرهای خالی منتقل می شود.
- اگر بافر خالی موجود نبود ، یکی از بافرهای پر انتخاب شده ، و سکتور مورد نظر به آن بافر منتقل می شود. (معمولاً از الگوریتم LRU برای این منظور استفاده می شود)

khademzadeh@iust.ac.ir

## استراتژی های بافرینگ – بافرینگ چندگانه

### بافر دهی دوگانه

- می توان برای همپوشانی اعمال پردازش توسط CPU و اعمال I/O از دو بافر استفاده نمود.
- فرض کنیم که برنامه فقط روی دیسک می نویسد.
- CPU می خواهد در حالی که عمل ورودی خروجی در حال انجام است در بافر چیزی بنویسد.
- اگر از دو بافر استفاده شود این عمل قابل انجام بوده و CPU یکی از بافرها را پر می کند و بافر دیگر در حال انتقال به دیسک خواهد بود.
- در زمانی که هر دو عمل فوق تمام شد، نقش دو بافر عوض می شود.
- مدیریت اصلی انجام عمل فوق توسط سیستم عامل صورت می گیرد.

khademzadeh@iust.ac.ir

## استراتژی های بافرینگ – حالت حرکت و حالت تعیین محل در بافرینگ

- حالت حرکت
  - حالتی که داده ها باید همیشه از بافر سیستم به بافر برنامه منتقل شود (و بالعکس)
  - اگر تعداد این انتقال ها زیاد باشد ، زمان لازم برای انجام این کار هم قابل ملاحظه خواهد بود.
- حالت تعیین محل – برای رفع مشکل فوق سیستم عامل می تواند به دو صورت عمل کند :
  - مدیر فایل از بافرهای سیستم جهت انجام اعمال ورودی-خروجی استفاده کند. اما مکان بافرهای سیستم را با استفاده از اشاره گرها در اختیار برنامه قرار دهد.
  - مدیر فایل اعمال ورودی-خروجی را بین دیسک و ناحیه داده های برنامه انجام دهد.

[khademzadeh@iust.ac.ir](mailto:khademzadeh@iust.ac.ir)

سوال ؟