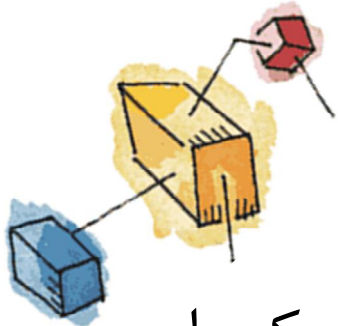


به نام خدا

فصل یازدهم

مدیریت ورودی/خروجی
و زمان بندی دیسک

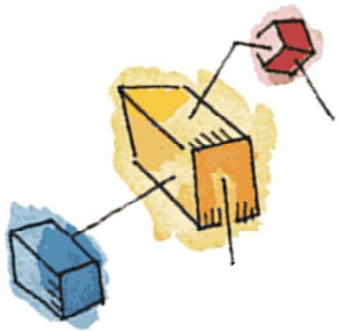
I/O Management and Disk Scheduling



دستگاه های I/O

- دستگاه های I/O و کاربردهای آنها بسیار متنوع هستند و ارائه یک راه حل عام برای همه آنها مشکل است.
- با بررسی روند تکاملی سیستم های عامل، ملاحظه می شود که تلاش به این سمت بوده که اعمال I/O بدون دخالت پردازنده انجام شود.
- هرچه که پردازنده سیستم، از عملیات I/O آزاد شود، کارایی سیستم بیشتر می شود.
- در سیستم های امروزی، I/O دیسک مهم ترین I/O و کلید کارایی سیستم است.
- در طی سال های گذشته، افزایش سرعت پردازنده ها و حافظه اصلی بسیار بیشتر از افزایش سرعت دسترسی به دیسک بوده است.



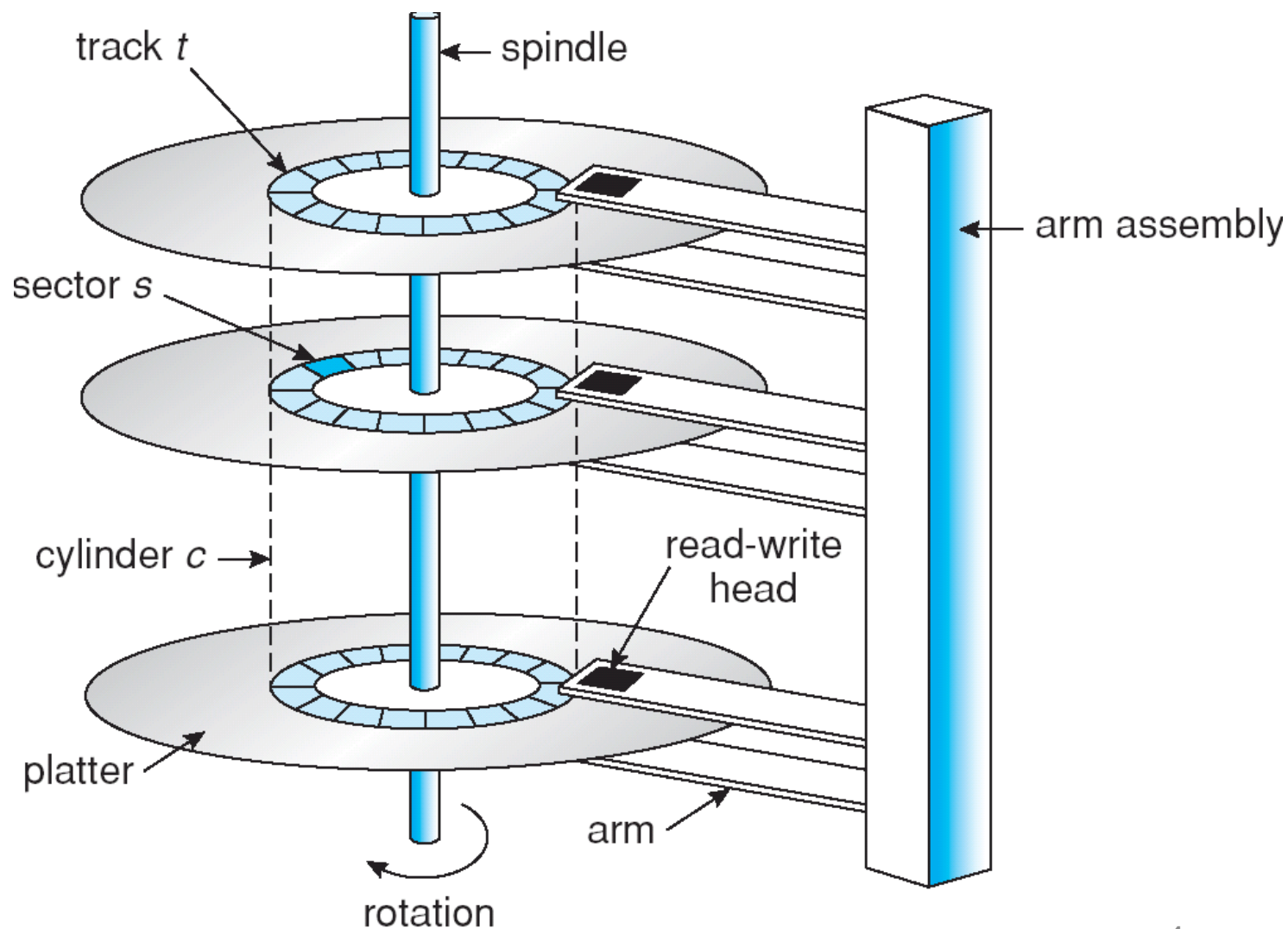


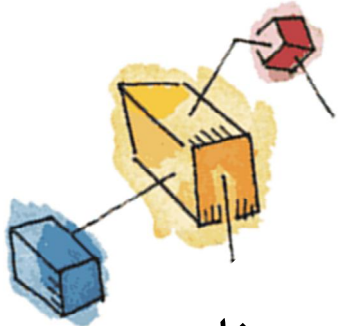
ساختار دیسک مغناطیسی

- یک head خواندن-نوشتن در بالای هر سطح هر صفحه قرار دارد.
- هدها به بازوی دیسک متصل هستند که هدها را به حرکت در می آورد.
- سطح صفحه به شیارهای حلقوی (track) تقسیم می شود.
- هر شیار به چندین قطاع (sector) تقسیم می شود.
- مجموعه ای از شیارها که در یک مکان از بازو قرار دارند، سیلندر یا استوانه نامیده می شوند.
- نرخ انتقال، به معنی نرخ انتقال داده ها بین درایور دیسک و کامپیوتر است.



دیسک مغناطیسی و head خواندن و نوشتن

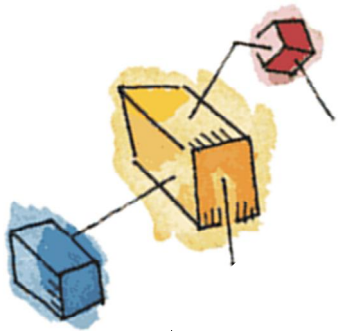




پارامترهای کارایی دیسک

- هنگامی که گرداننده دیسک کار می کند، دیسک با سرعت ثابتی می چرخد.
- برای خواندن و نوشتن، باید هد دیسک بر روی شیار موردنظر و در ابتدای قطاع مورد نظر از آن شیار قرار گیرد.
- برای انتخاب شیار:
 - در دیسک های با هد متحرک، نیاز است که هد به سمت شیار حرکت کند. زمان لازم برای این کار را زمان پیگرد (seek time) می گویند.
 - در دیسک های با هد ثابت، هد مربوط به شیار موردنظر به صورت الکترونیکی انتخاب می شود.





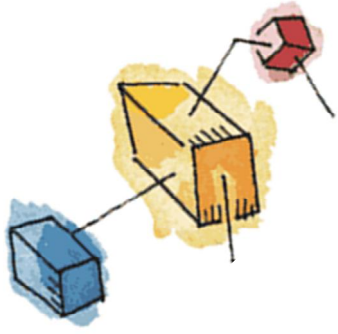
پارامترهای کارایی دیسک

- سپس با چرخیدن دیسک، قطاع مورد نظر به زیر هد می رسد.
- زمان لازم برای رسیدن قطاع موردنظر به هد را تاخیر چرخشی می گویند.

- زمان دسترسی: مجموع زمان پیگرد و زمان تاخیر چرخشی

- پس از اینکه هد در موقعیت مناسب قرار گرفت، انتقال داده ها صورت می گیرد و خواندن یا نوشتن انجام می شود.
- زمان لازم برای این کار، زمان انتقال نامیده می شود.





زمان انتقال

- زمان انتقال بستگی به سرعت چرخش دیسک دارد.

$$T = \frac{b}{rN}$$

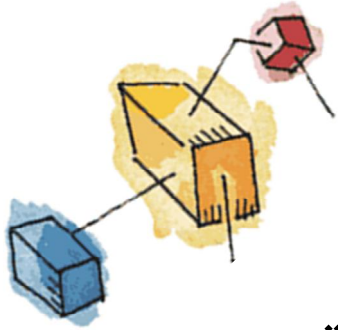
T = transfer time

b = number of bytes to be transferred

N = number of bytes on a track

r = rotation speed, in revolutions per second





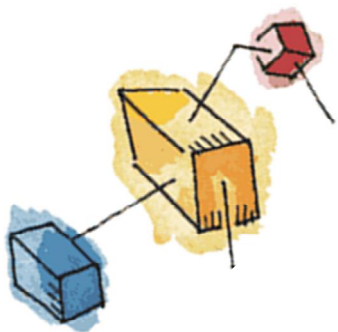
متوسط کل زمان دسترسی

- بنابراین، متوسط کل زمان دسترسی به صورت زیر است:

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

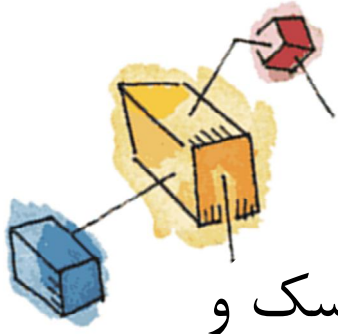
- T_s : متوسط زمان پیگرد (وابسته به قطر دیسک).
- $1/2r$: متوسط زمان تاخیر دورانی (به طور متوسط، دیسک نیم دور می چرخد تا به قطاع موردنظر برسیم).





سیاست های زمان بندی دیسک





زمان بندی دیسک

- هرگاه فرآیندی نیاز به I/O دیسک داشته باشد، اگر گرداننده دیسک و کنترلر آماده باشند، درخواست فوراً پاسخ داده می شود.
- اگر گرداننده دیسک و کنترلر مشغول باشند، درخواست جدید در صف درخواست ها قرار می گیرد.
- در سیستم چند برنامه ای، صف درخواست دیسک ممکن است همیشه حاوی چند درخواست معوق باشد.
- الگوریتم زمان بندی دیسک مشخص می کند که سیستم عامل از بین درخواست های موجود در صف کدام یک را انتخاب کند.





زمان بندی دیسک

- الگوریتم های مختلفی برای زمان بندی دیسک وجود دارند که در ادامه معرفی خواهند شد.
- برای بررسی عملکرد این الگوریتم ها، فرض می کنیم درخواست ها برای شیارهای زیر در صف قرار دارند:

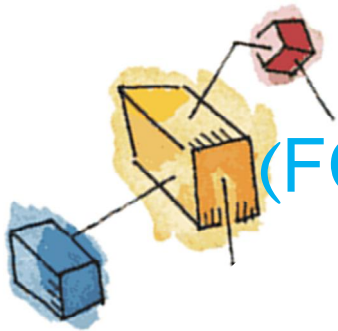
98, 183, 37, 122, 14, 124, 65, 67

- فرض می شود که در شروع کار، هد دیسک بر روی شیار ۵۳ قرار دارد.
- فرض می کنیم که دیسک دارای ۲۰۰ شیار است.

– خارجی ترین شیار: شیار ۱۹۹

– داخلی ترین شیار: شیار صفر

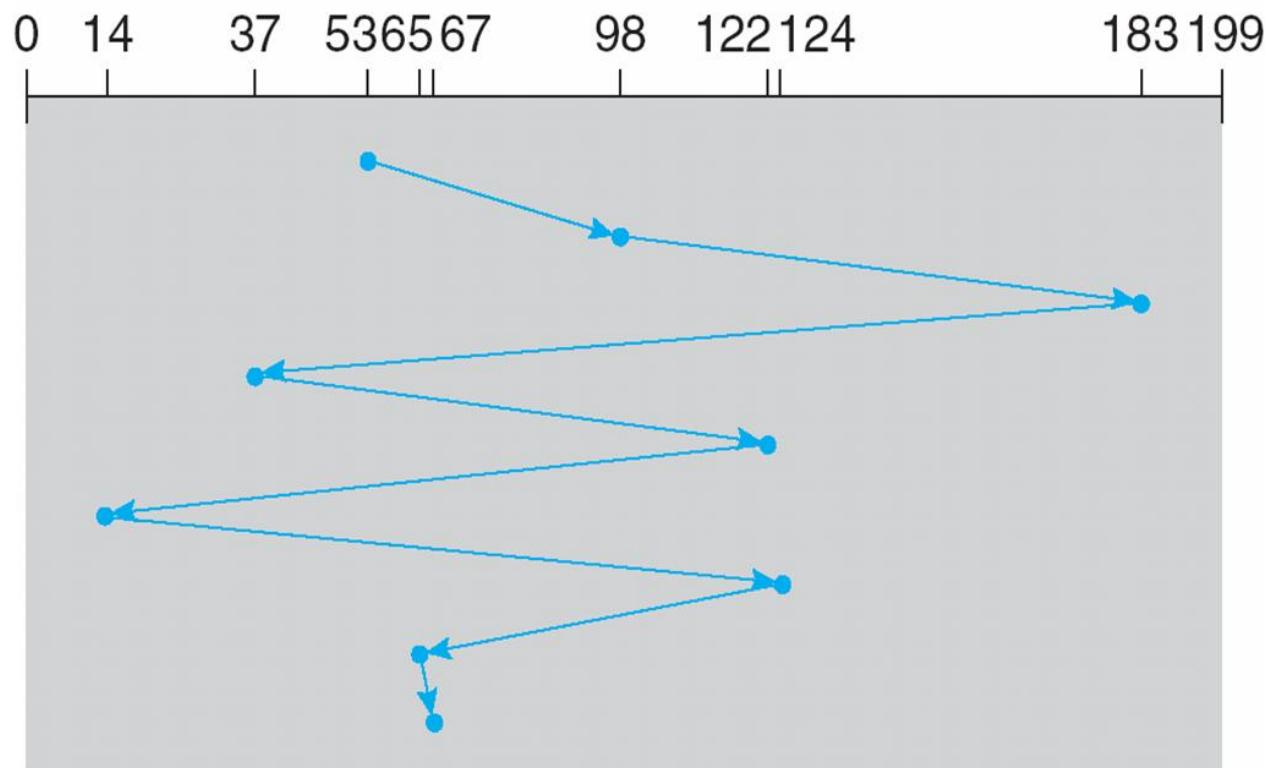




زمان بندی سرویس دهی به ترتیب ورود (FCFS)

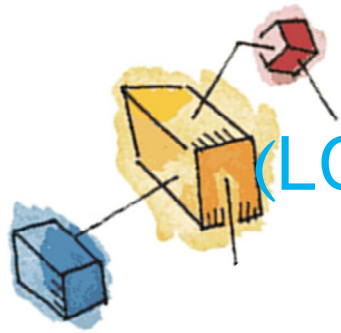
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



در مجموع، هد دیسک به اندازه ۶۴۰ سیلندر حرکت می کند.



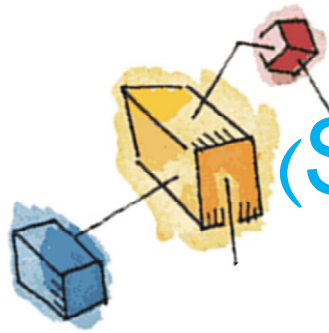


زمان بندی سرویس دهی به ترتیب عکس ورود (LCFS)

- این روش دارای امتیازاتی است.
- در بعضی سیستم ها منجر به حرکت کمتر یا عدم حرکت بازوی دیسک می شود.
- به علت اصل محلی بودن مراجعات
 - در خواستی که دیرتر از همه مطرح می شود، ممکن است مربوط به همان فرآیندی باشد که اخیرا درخواستش را پاسخ داده ایم. زیرا آن فرآیند اخیرا به حالت آماده برگشته و ممکن است پس از دریافت پردازنده، مجددا نیاز به دیسک پیدا کرده باشد.

ولی این روش منصفانه نیست و ممکن است باعث گرسنگی شود.

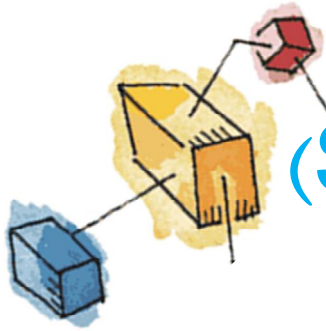




الگوریتم کوتاه ترین زمان پیگرد (SSTF)

- این الگوریتم درخواستی را انتخاب می کند که نسبت به موقعیت فعلی هد، کمترین زمان پی گرد (seek time) را دارد.
- یعنی درخواستی که به موقعیت فعلی هد نزدیک تر است.
- این الگوریتم نوعی از الگوریتم های SJF است.

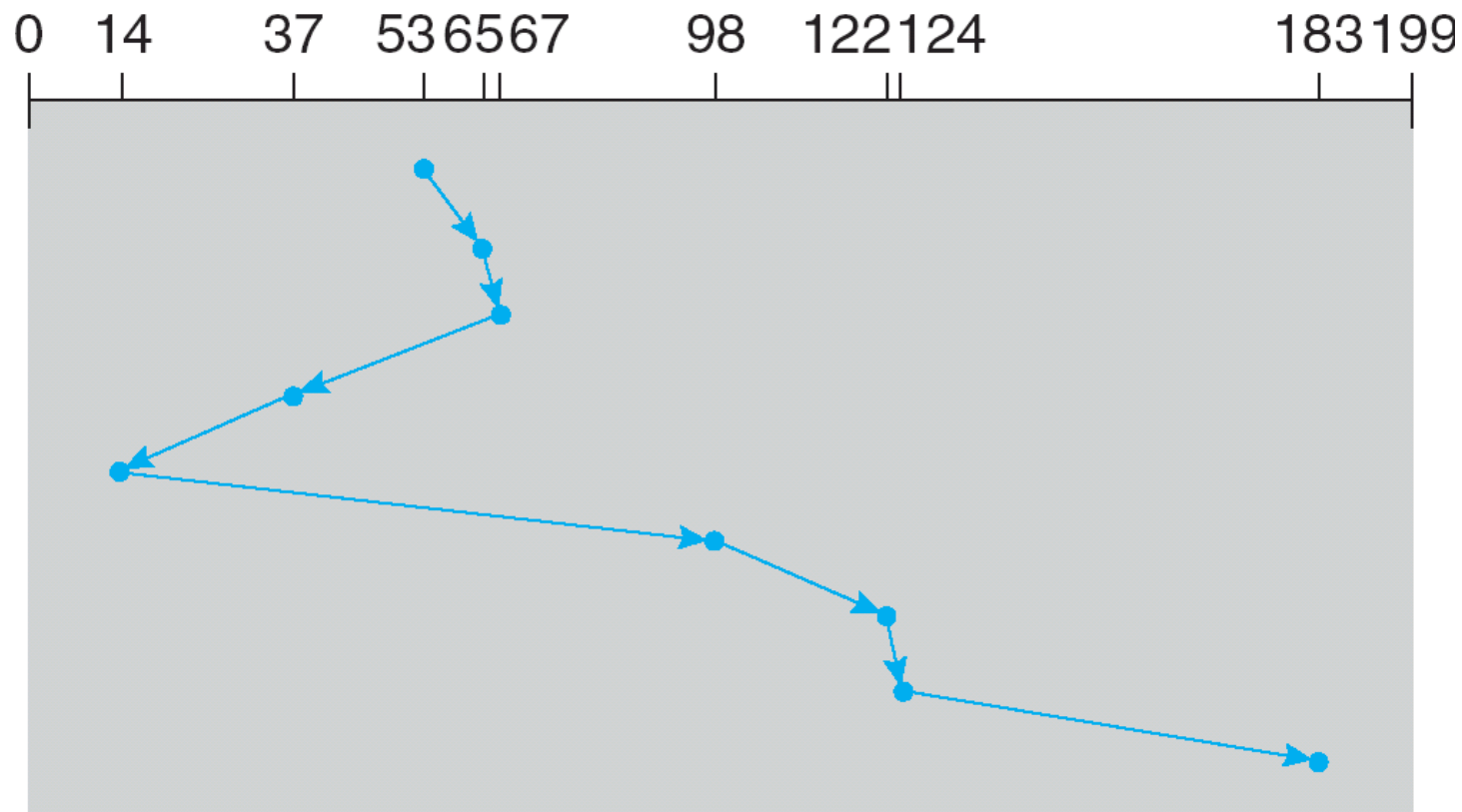




الگوریتم کوتاه ترین زمان پیگرد (SSTF)

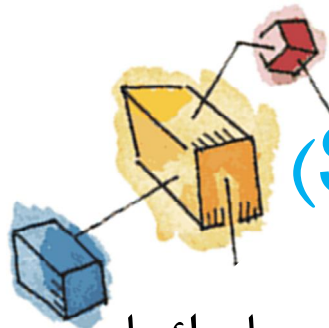
queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



در مجموع، هد دیسک به اندازه ۲۳۶ سیلندر حرکت می کند.





الگوریتم کوتاه ترین زمان پیگرد (SSTF)

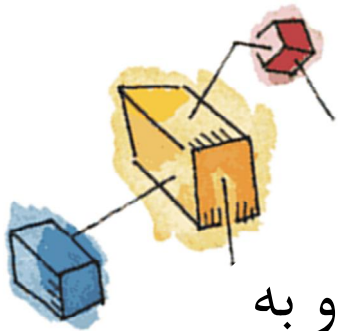
- ممکن است الگوریتم SSTF دچار گرسنگی شود و برخی درخواست ها دائما به تعویق بیافتند.

- گرچه الگوریتم SSTF بهتر از FIFO است ولی بهینه نیست.

- اگر درخواست ها در این مثال به ترتیب زیر پاسخ داده شوند، تعداد حرکت هد به ۲۰۸ خواهد رسید.

53, 37, 14, 65, 67, 98, 122, 124, 183

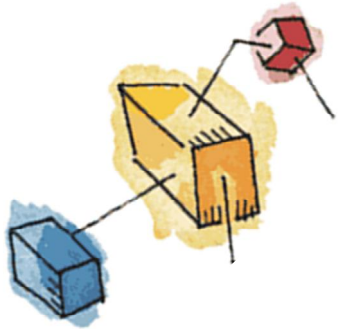




الگوریتم پیمایش (SCAN)

- در این الگوریتم، هد از یک طرف دیسک شروع به حرکت می کند و به سمت دیگر می رود و در طول مسیر، به درخواست های موجود در مسیر پاسخ می دهد.
- در انتهای دیسک، مسیر حرکت هد، برعکس می شود و مسیری که پیموده است را بر می گردد.
- این الگوریتم، الگوریتم آسانسور نیز نامیده می شود زیرا مانند آسانسور یک ساختمان ابتدا درخواست های رو به بالا و سپس درخواست های رو به پایین را پاسخ می دهد.
- برای بررسی مثال، باید جهت حرکت هد در شروع کار را بدانیم.

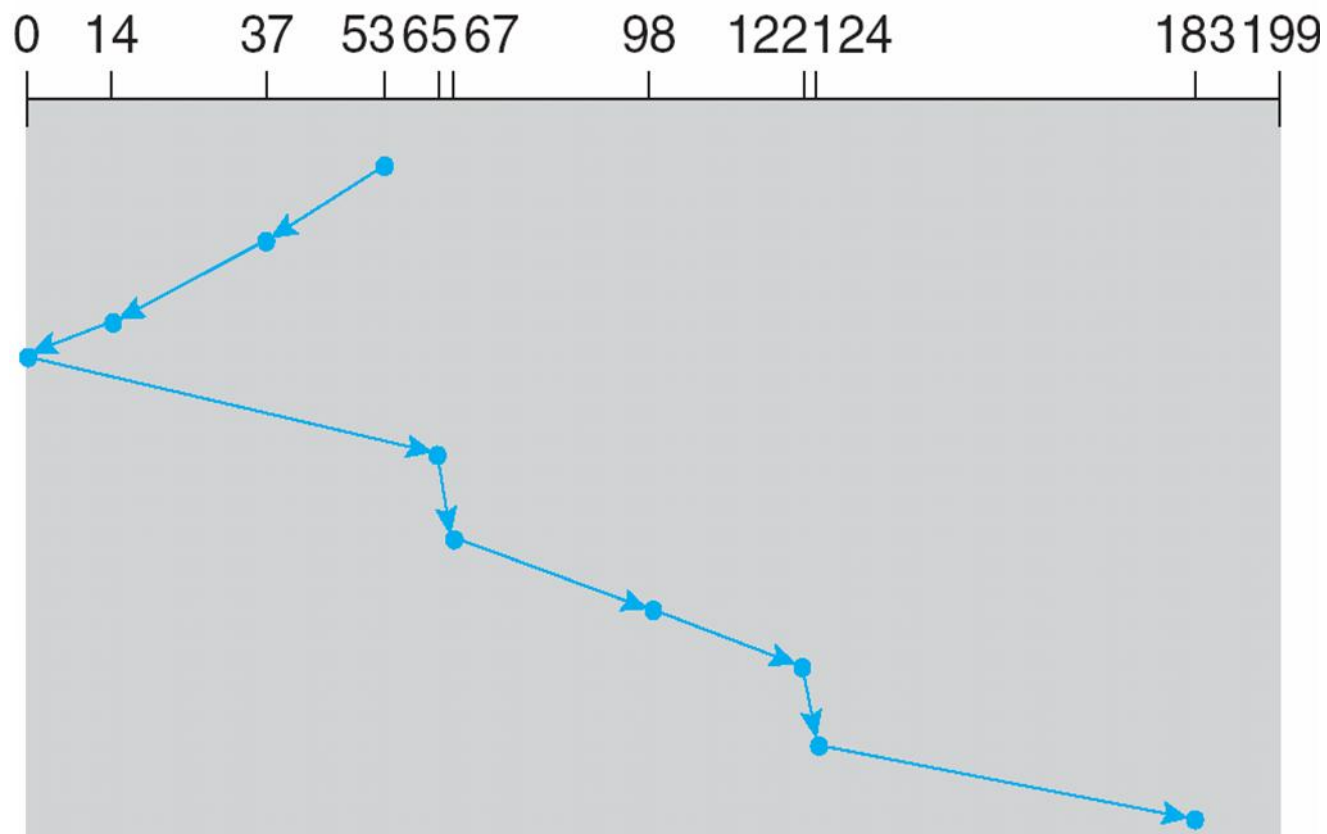


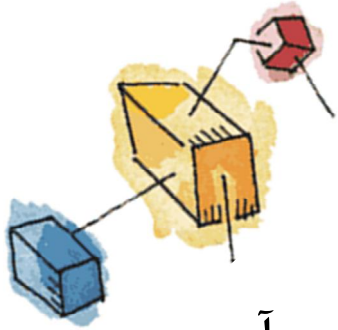


الگوریتم پیمایش (SCAN)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



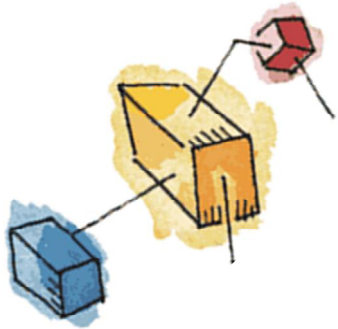


الگوریتم پیمایش حلقوی (C-SCAN)

- در الگوریتم SCAN، هنگامی که هد، تغییر جهت می دهد و مسیر آمده را برمی گردد، درخواست های کمی در جلوی هد قرار خواهند داشت. – زیرا هد اخیرا از روی این سیلندرها عبور کرده است و این سیلندرها اخیرا خدمات گرفته اند.

- الگوریتم C-SCAN شبیه به الگوریتم SCAN است با این تفاوت که هنگامی که هد به انتهای دیسک رسید، این بار از طرف دیگر دیسک شروع می کند.

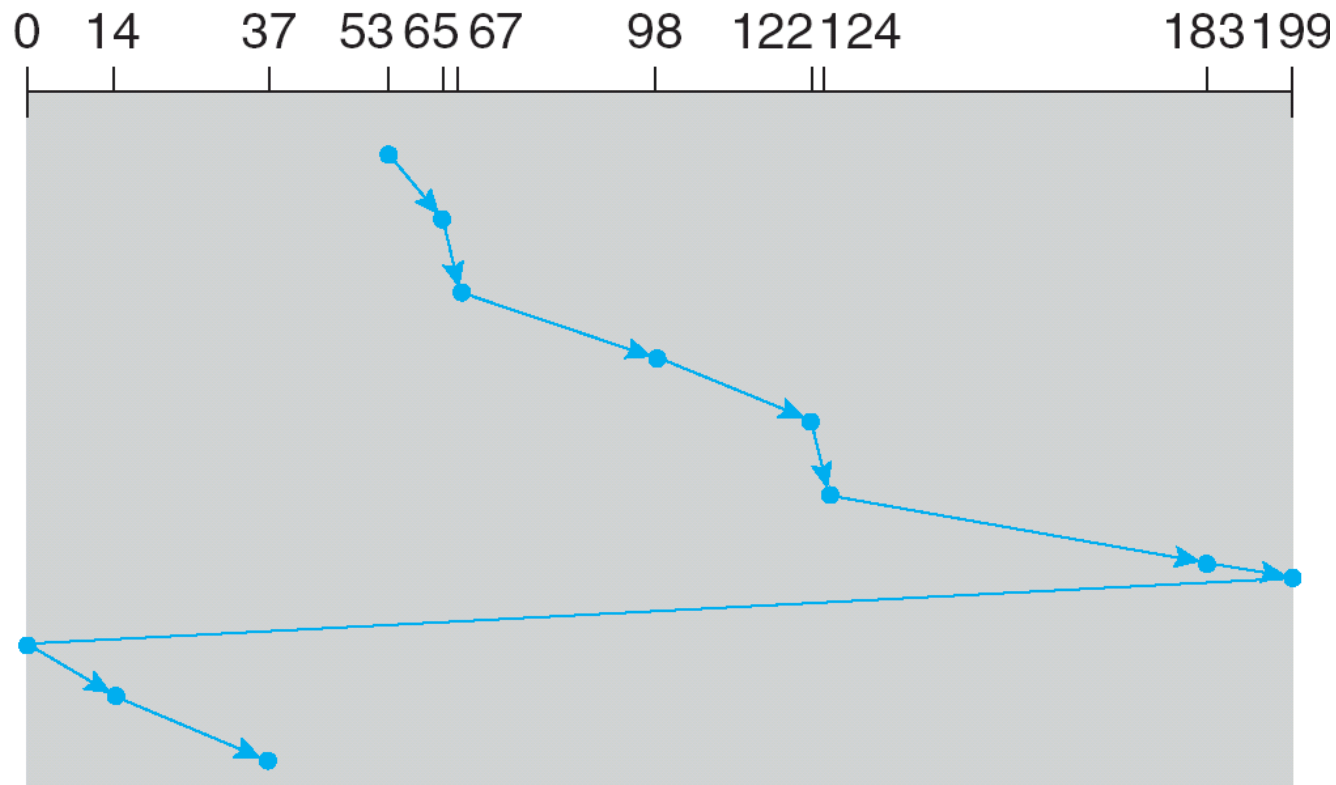


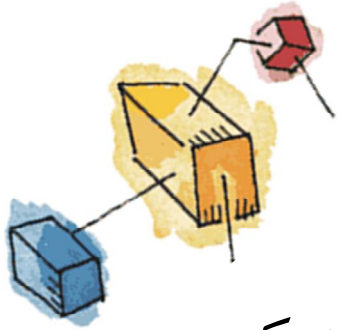


الگوریتم پیمایش حلقوی (C-SCAN)

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53

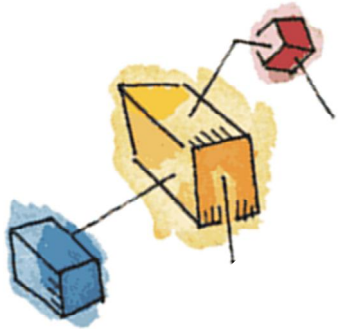




الگوریتم C-LOOK

- این الگوریتم نسخه ای از الگوریتم C-SCAN است با این تفاوت که در آن، هد تا انتهای دیسک پیش نمی رود.
- بلکه به جلو نگاه می کند و تا آخرین درخواست موجود در هر طرف دیسک پیش می رود بدون اینکه به انتهای دیسک برسد.
- به طور مشابه، الگوریتم LOOK، نسخه ای از الگوریتم SCAN است که هد در آن تا آخرین درخواست پیش می رود.

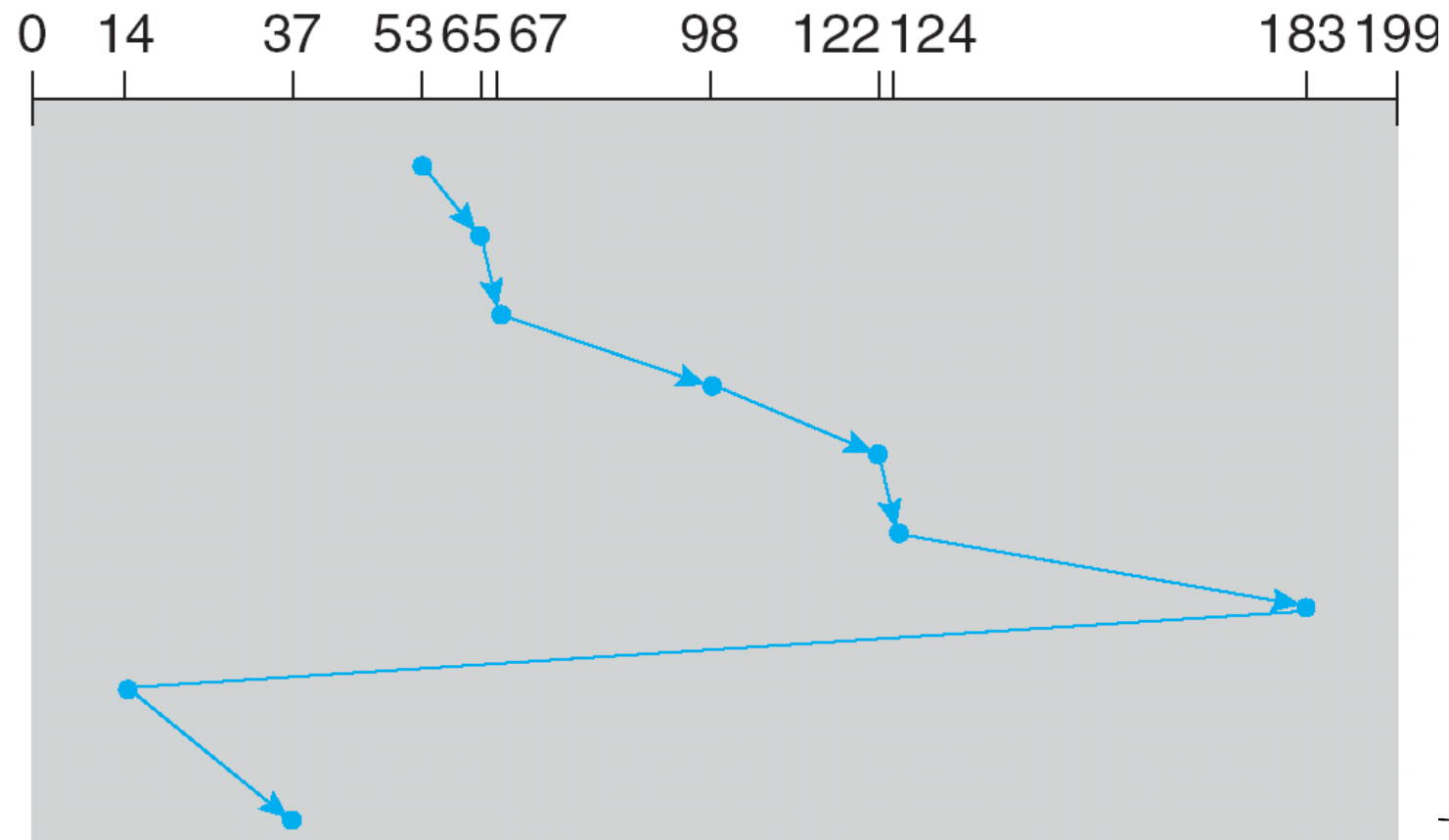


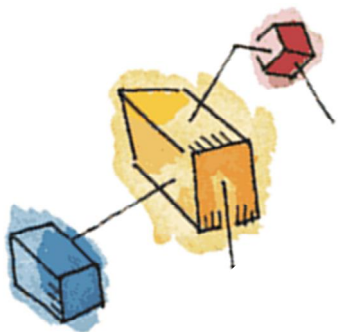


C-LOOK الگوریتم

queue 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53





• پایان فصل یازدهم

