

Linux

Apple

Windows



تعاریف سیستم عامل

- ❑ برنامه‌ای است که اجرای برنامه‌های کاربردی (Application Programs) را کنترل می‌کند.
❖ کنترل اجرا مرتبا بین برنامه‌ها و سیستم عامل منتقل می‌شود.
- ❑ رابطی (Interface) بین کاربر و سخت افزار
- ❑ یک تخصیص دهنده منابع و یا مدیر منابع

ویژگیهای سیستم عامل

☐ سهولت (Convenience)

❖ استفاده از منابع سیستم را ساده تر می کند.

☐ کارآمدی (Efficiency)

❖ موجب استفاده کارآمد از منابع سیستم می شود.

☐ قابلیت تکامل (Ability to Evolve)

❖ امکان بررسی، یافتن اشکالات و تغییر ساختار سیستم را فراهم می کند.

❖ سخت افزارهای سیستم می تواند به سادگی به آن افزوده شوند.

هسته سیستم عامل (kernel)

- ❑ بخش اصلی سیستم عامل که در حافظه اصلی بار می شود که شامل پرکاربردترین توابع سیستم عامل است.
- ❑ بیشترین دفعات استفاده را دارد.
- ❑ در مورد زمان استفاده از I/O تصمیم می گیرد.
- ❑ میزان تخصیص پردازنده به یک کار را مشخص می کند.

سرویس‌های ارائه شده توسط سیستم عامل

← ایجاد برنامه

← اجرای برنامه

← کنترل دسترسی به فایلها

← کشف و پاسخ به خطا

← جمع آوری اطلاعات آماری سیستم

← دسترسی به دستگاه های ورودی و خروجی

← دسترسی به سیستم

← حسابداری

← صورت حساب کاربران

- ۱- سیستم‌های پردازش ترتیبی
(Serial Processing)
- ۲- سیستم‌های دسته‌ای ساده
(Simple Batch Systems)
- ۳- سیستم‌های چند برنامه‌ای دسته‌ای
(Multiprogramming)
- ۴- سیستم‌های اشتراک زمانی
(Time Sharing)

سیستم‌های پردازش ترتیبی

- ✓ به علت دسترسی کاربران به کامپیوتر به صورت نوبتی، این نام را دارند.
- ✓ در سال ۱۹۵۰-۱۹۴۰ به وجود آمد.
- ✓ کاربر با سخت افزار مستقیماً در ارتباط بود و عملاً سیستم عاملی وجود نداشت.
- ✓ این ماشینها از طریق یک میز فرمان (شامل چراغها و کلیدها و ...) اجرا می شدند.
- ✓ برنامه‌ها به زبان ماشین و بوسیله دستگاه ورودی (کارت خوان) بارگذاری می شد.
- ✓ اگر خطایی پیش می آمد، شرایط خطا توسط چراغها بیان می شد و برنامه ساز می توانست محتویات ثبات‌ها و حافظه اصلی را بررسی کند.
- ✓ خروجی در چاپگر ظاهر می شد.

مشکلات اساسی سیستم‌های پردازش ترتیبی

زمانبندی :

❖ توسط برگه‌های نوبت‌گیر، برای ماشین وقت می‌گرفتند و باید در زمان پیش‌بینی شده کارش تمام می‌شد.

زمان آماده‌سازی :

❖ برای هر کار (Job) یکسری کارتها و نوارها باید نصب می‌شد.

❖ اگر در زمان آماده‌سازی برای هر یک از کارها مشکلی ایجاد می‌شد باید از اول کارش را شروع کند.

- ❖ در این سیستم ها کارها (Job) بصورت یک لیست و بصورت خودکار توسط ماشین خوانده شده و به ترتیب اجرا می‌شود.
- ❖ یک برنامه بنام ناظر (Monitor) نقش سیستم عامل را بعهده دارد که نتایج اجرای برنامه ها را برای عرضه به کاربر حفظ می‌کند.
- ❖ با تکمیل هر برنامه، پردازنده به ناظر باز می‌گردد و ناظر به‌طور خودکار برنامه بعدی را بار (Load) کرده و سپس پردازنده را در اختیار برنامه بارشده قرار می‌دهد.

سیستم های اشتراک زمانی Time Sharing

- این سیستم ها نیز از چند برنامه گشتیبانی می کنند.
- برای کاربردهای محاوره ای (Interactive) لازم است که میزان معطلی اجرای برنامه ها کم شود پس زمان اجرا بین آنها پخش می شود.
- معمولاً روند بدین صورت که یک واحد زمانی خاص تعریف می شود و به هر کاربر به اندازه این واحد، زمان اختصاص می یابد.
- کاربران متعدد همزمان از طریق ترمینالها می توانند به سیستم دسترسی داشته باشند.

اهداف اصلی	چند برنامه ای دسته ای	اشتراک زمانی
منبع دستورات به سیستم عامل	زبان کنترل کار	کارها با دستور مستقیم کاربران از طریق ترمینال
حداکثر شدن کارایی پردازنده	حداقل زمان پاسخ	

زبان کنترل کار (Job Control Language)

□ نوعی زبان برنامه‌نویسی برای فرمان دادن به ناظر (Monitor) برای تعیین نوع کامپایلر

یا برای تعیین نوع داده ورودی می باشد که با علامت \$ مشخص می‌شدند.

\$JOB

\$FTN

•
•
•

FORTRAN instructions

\$LOAD

\$RUN

•
•
•

Data

\$END

نکته: این دستورالعمل‌ها ممتاز (الویت دار) می باشند.
(توسط ناظر اجرا می‌شوند و پردازنده سریعاً به ناظر منتقل می‌گردد)

□ می دانیم بعضی سیستم عاملها به صورت دسته ای و به ترتیب برنامه ها را خوانده، پردازش و

خارج می کردند. با روش **spooling** که همزمان بود با استفاده از دیسک سخت؛

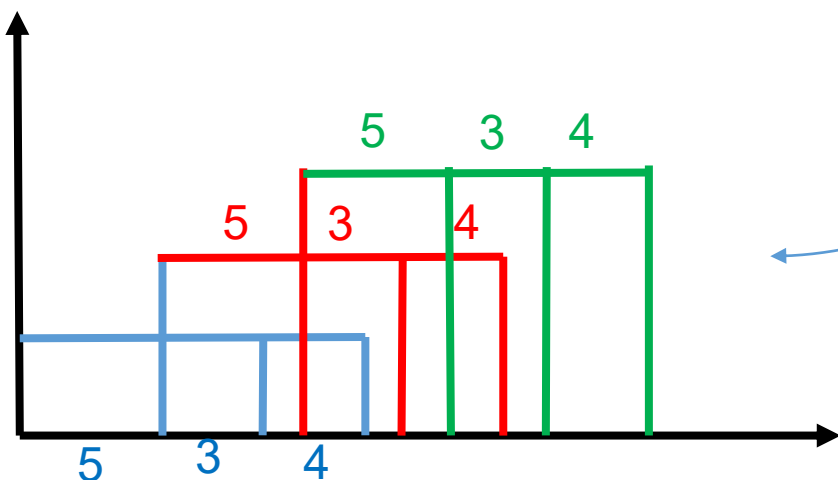
زمان بیکاری cpu کمتر شد. چون همزمانی (concurrent) عملیات ورودی و خروجی و پردازش امکان پذیر شد.

❖ روش spooling یعنی: همزمان که cpu پردازش می کند، داده های برنامه بعدی روی حافظه جانبی خوانده می شود. همچنین می توان خروجی برنامه ای دیگر را ذخیره نمود. چون سیستم عامل این همزمانی کارها را کنترل و امکان پذیر می سازد.

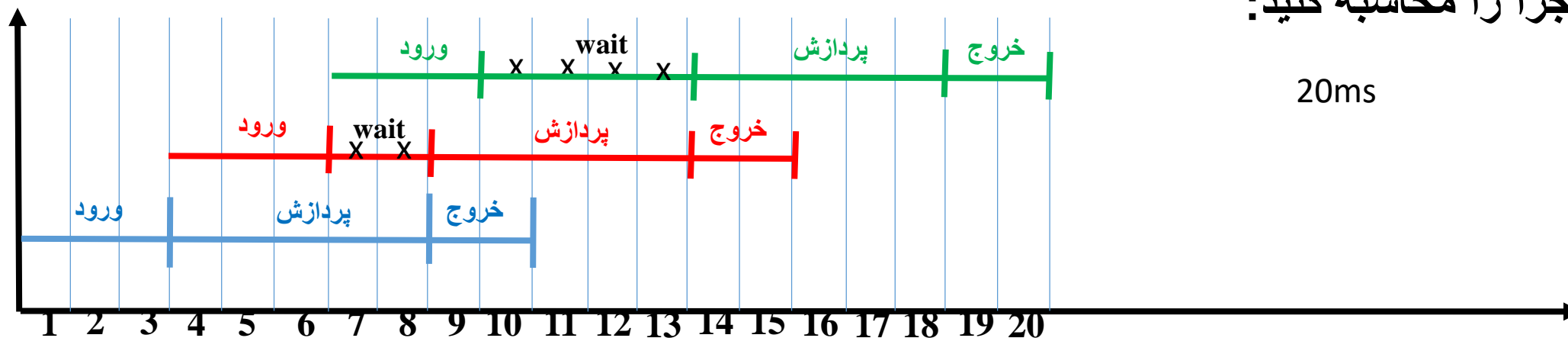
مثال: فرض کنید سه برنامه برای اجرا داریم که زمانهای ورود و پردازش و خروج آنها (5 و 3 و 4) میلی ثانیه باشد. با روش اجرای Batch و روش spooling مدت زمان اجرای آنها را بنویسید.

حل: در روش دسته ای خواهیم داشت: $(5 + 3 + 4) + (5 + 3 + 4) + (5 + 3 + 4) = 36 \text{ ms}$

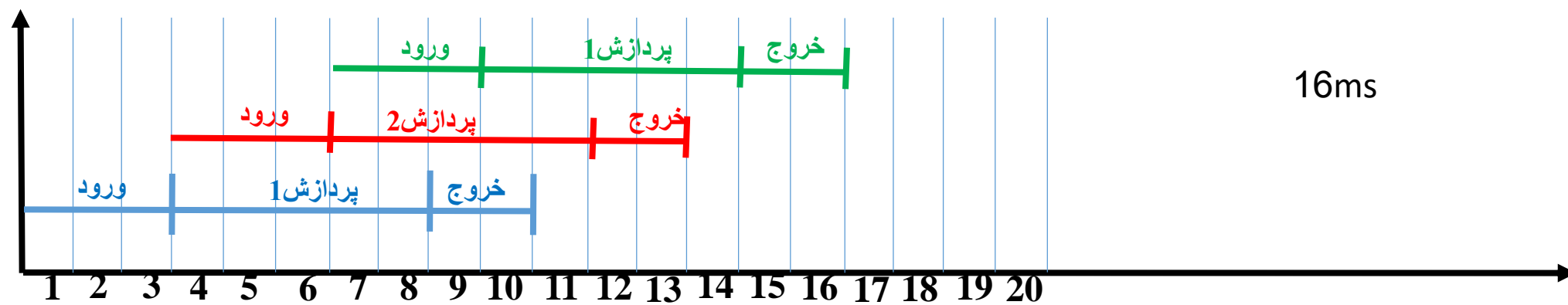
اما با روش **spooling** خواهیم داشت: 22ms



مثال: در یک سیستم **spooling** سه کار با زمانهای ورودی 3 و پردازش 5 و خروجی 2 میلی ثانیه باشد. کل زمان اجرا را محاسبه کنید:



در مثال فوق یک پردازنده وجود داشت در صورتیکه دو پردازنده باشد. زمان اتمام کارها چقدر می شود؟



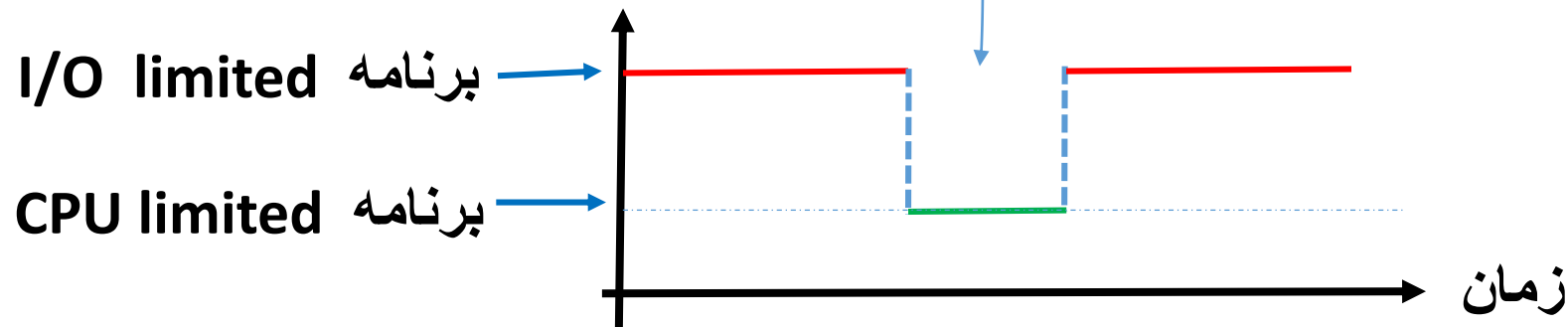
CPU limited & I/O limited JOB

□ در برنامه هایی که ورودی و خروجی زیادی دارند. این وسایل تعیین کننده سرعت کامپیوتر خواهند بود.

□ در برنامه هایی که پردازش زیادی دارند. پردازشگر تعیین کننده سرعت کامپیوتر خواهند بود.

تعیین اینکه برنامه در ابتدا کدام نوع خواهد بود مشکل است
اگر فقط یک برنامه پردازشی و یک برنامه ورودی و خروجی
با هم اجرا شود

در این زمان وسیله I/O
منتظر آماده سازی داده
می باشد



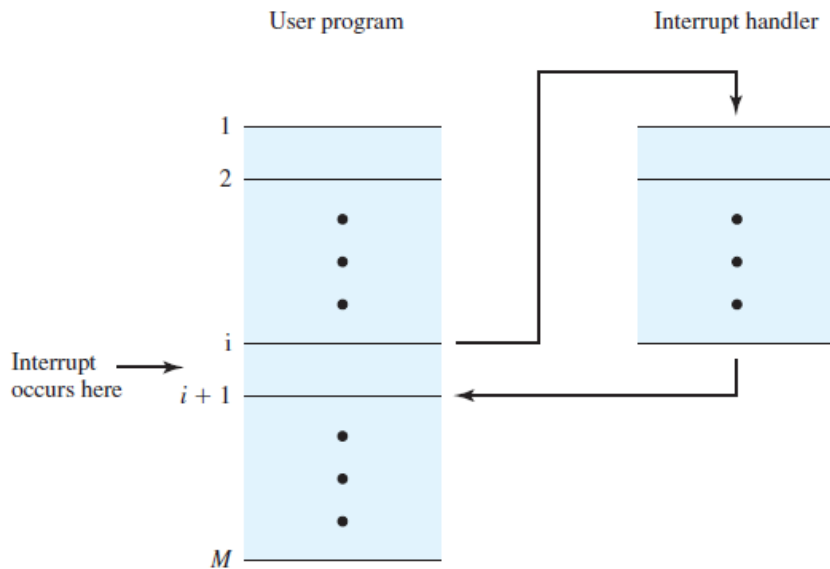
ایرادات:

اینکه معلوم نیست کدام کار پردازش می خواهد یا ورودی خروجی
اینکه کارهای پردازشی هر چند کم اما به ورودی خروجی نیاز دارند. (چند وسیله ورودی خروجی باید باشد)
(در هر لحظه هر دستگاه جانبی peripheral فقط در اختیار یک برنامه می باشد)

❑ چه وقت عملیات ورودی و خروجی تمام می شود تا پردازش شروع شود؟

CPU می تواند مرتبا وضعیت وسیله I/O را چک کند تا وقتی که مشغول است کار خودش را انجام دهد. (Pooling)
نیاز است CPU در یک حلقه را پردازش کند تا مرتبا وضعیت وسیله را چک کند که به آن حلقه busy waiting
گویند. (وقت گیر می باشد)

CPU می تواند پردازشها را انجام دهد و وسیله ورودی خروجی هر گاه آماده بود یا کارش تمام شده
سیگنالی بفرستد. (Interrupt)



Transfer of Control via Interrupts

❑ جهت اجرای برنامه موافق با هر سیگنال وقفه ابتدا پردازش موجود
متوقف می شود. داده ها و آدرسها و کلیه اطلاعات پردازشی (status)
آن در محلی ذخیره می گردد (dump). برنامه موافق سیگنال وقفه
(interrupt service routine) اجرا می گردد.
سپس اجرای برنامه قبلی از جاییکه متوقف شده بود ادامه می یابد.

Interrupt

□ تقاضای وقفه از منابع مختلفی صادر می گردد:

الف) وقفه های داخلی (trap)(تله): دستور وقفه در داخل دستورات برنامه کاربر باشد

ب) وقفه خارجی: از وسایلی مانند I/O یا DMA یا تایمر یا صفحه کلید یا خطاهای سخت افزاری می باشد.

ج) وقفه نرم افزاری: فراخوانی توابع سیستمی توسط برنامه کاربر.

□ جهت پاسخ به وقفه باید پردازش جاری متوقف شده و وضعیت آن حفظ شود (psw program status word)

سپس کار تعویض متن (context switching) انجام شده و برنامه وقفه (Interrupt Service Routine ISR)

اجرا گردد.

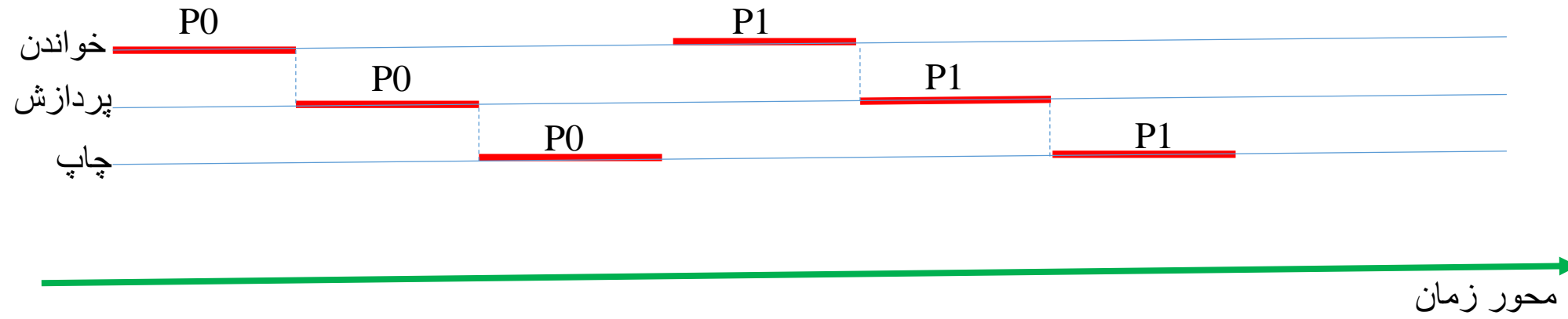
□ پاسخ به وقفه ها یا به صورت polling یا با الویت داده می

شوند

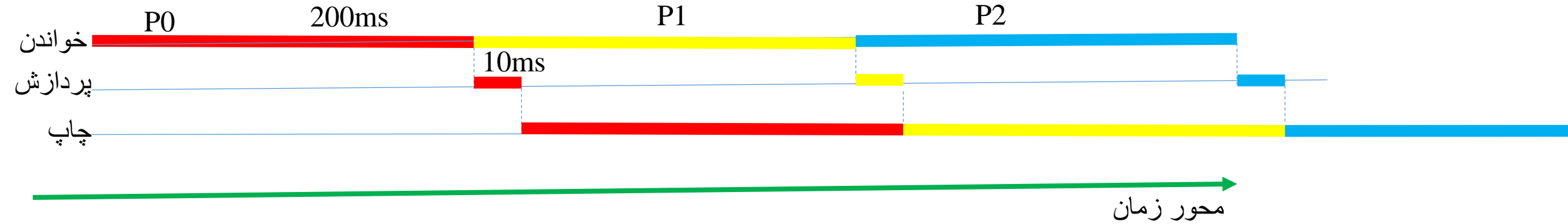
□ جهت انتقال داده ها از حافظه اصلی به مکانی دیگر یا بر عکس، بدون دخالت cpu ،

از سخت افزار DMA(Direct Memory Access) استفاده می شود

انجام عملیات در ماشین ترتیبی



انجام عملیات در ماشین چند برنامه ای

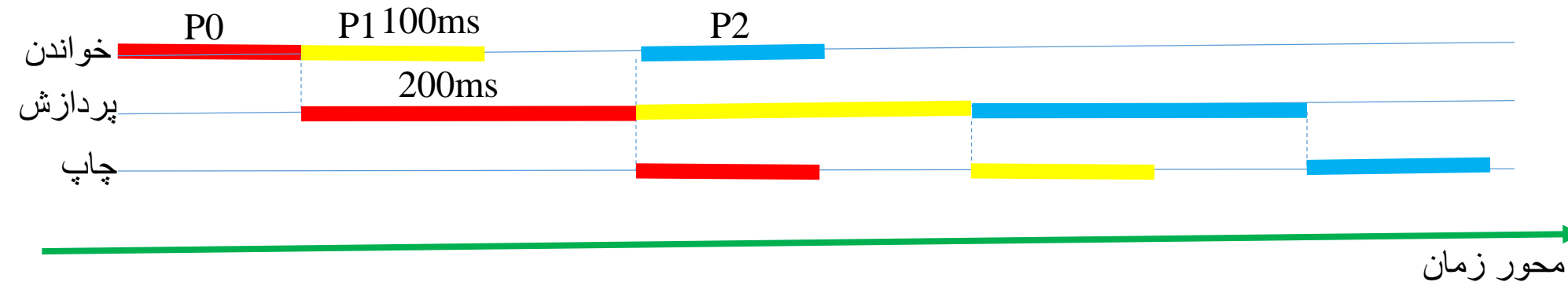


برنامه های I/O Limited

□ داده های ورودی یا خروجی در حافظه ای بافر می شوند (با ظرفیت سه برنامه)

برنامه های CPU Limited

□ داده های ورودی یا خروجی در حافظه ای بافر می شوند (با ظرفیت دو برنامه)



$$\text{CPU کارایی} = \frac{\text{زمان استفاده از CPU}}{\text{کل زمان}} \times 100$$