



دانشگاه آزاد واحد ساری

سیستم های چندپردازنده ای

سیستم های عامل پیشرفته

مدرس:

دکتر میر سعید حسینی شیروانی

فهرست مطالب

مقدمه: نگاه کلی به سیستم عامل

سیستمهای چندپردازنده ای

فصل اول: مقدمه ای بر سیستمهای توزیع شده

فصل دوم: معماری ها

فصل سوم: فرایندها

فصل چهارم: ارتباطات

فصل پنجم: نامگذاری

انواع سیستم‌های چندپردازنده ای

از دیدگاه سخت افزاری

همگن (متجانس)

نا همگن (نا متجانس)

از دیدگاه نرم افزاری

متقارن: در این سیستم ها، تمامی پردازنده ها دارای نسخه یکسانی از سیستم عامل هستند و در هنگام نیاز با هم ارتباط برقرار می کنند.

نا متقارن: در این سیستم ها، یا هر پردازنده سیستم عامل خاص خودش را دارد (با هم متفاوتند) و یا اینکه یک سیستم عامل به بخش‌هایی شکسته شده و هر بخش روی یک پردازنده اجرا می شود.

مزایای یک سیستم چندپردازنده ای

افزایش سرعت (بعلت وجود اجرای موازی برنامه ها)

افزایش قابلیت اطمینان (در صورت خرابی یک پردازنده، دیگر پردازنده

ها می توانند این کار را انجام دهند اما با کارایی پایین تر)

کاهش نسبت هزینه به کارایی (به جای تهیه چند کامپیوتر کامل می توان

از چند پردازنده و تنها یک **case** و یک **power** استفاده کرد.)

مقایسه سیستم های چند کامپیوتری با چند پردازنده ای

هر دو سیستم به روش چند داده - چند دستور (MIMD) عمل می کنند.

ارتباط میان فرآیندها در سیستم های چند پردازنده ای با استفاده از حافظه مشترک (Shared Memory) و در سیستم های چند کامپیوتری از طریق تبادل پیغام (Message Passing) صورت می گیرد.

سهولت قابلیت گسترش (Scalability): در سیستمهای چند کامپیوتری راحت تر از چند پردازنده ای ها انجام می شود.

روی سیستم های چند پردازنده ای یک نوع سیستم عامل وجود دارد اما در سیستم های چند کامپیوتری می تواند بر روی هر کامپیوتر سیستم عاملهای متفاوتی وجود داشته باشد.

مقایسه سیستم های چند کامپیوتری با چند پردازنده ای

معیار کارایی/هزینه

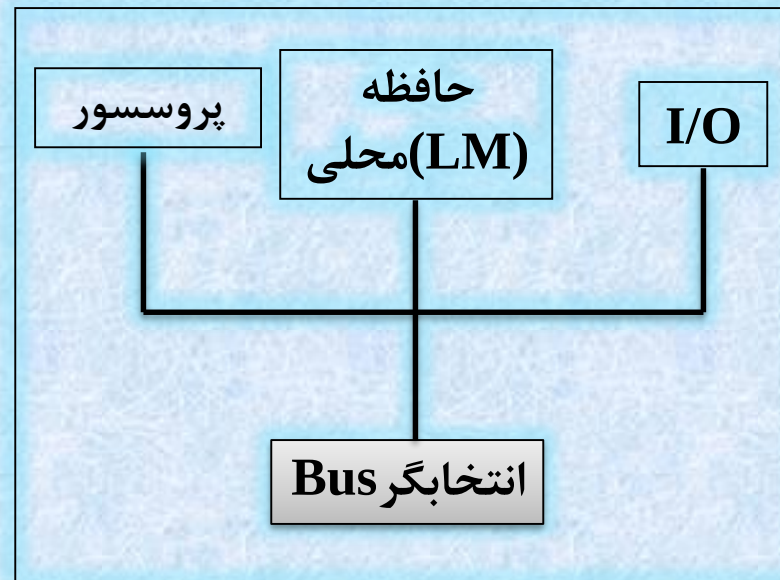
سرعت درسیستم های چندپردازنده ای نسبت به سیستم های چند کامپیوتری بیشتر است.

چند کامپیوتری ها از نظر سرعت مانند چند پردازنده ای ها عمل می کنند به شرطی که شبکه ارتباطی آنها سریع باشد.

معماری یک سیستم چند کامپیوتری



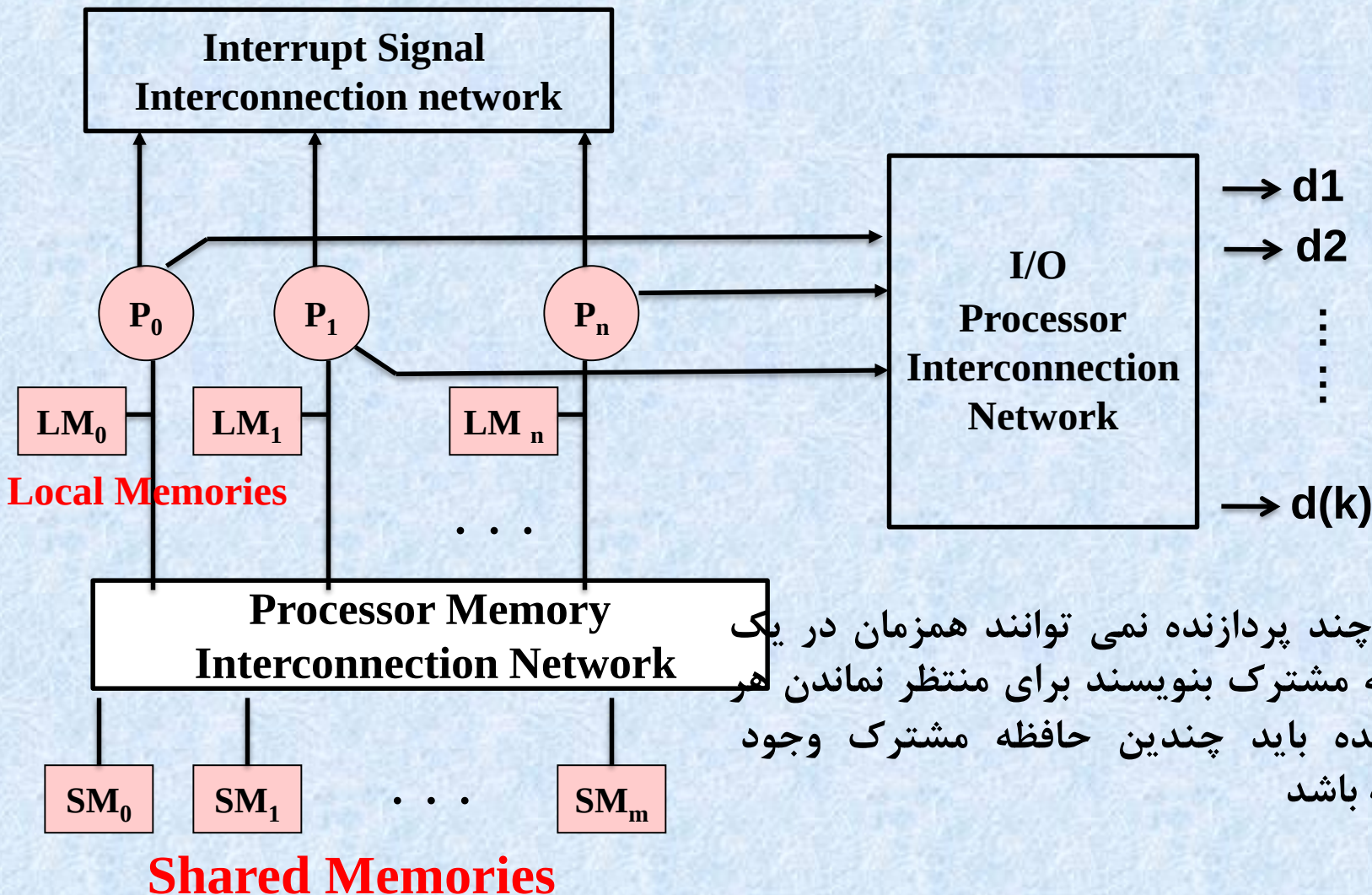
شبکه



نمایی از یک

سیستمی از یک ماژول عامل پیشرفته

معماری یک سیستم چندپردازنده ای



چون چند پردازنده نمی توانند همزمان در یک حافظه مشترک بنویسند برای منتظر نماندن هر پردازنده باید چندین حافظه مشترک وجود داشته باشد

ملاحظات طراحی سیستم عامل چند پردازنده ای

هم زمانی فرآیند ها یا نخ ها: توسط هر یک از پردازنده ها انجام شود.

زمانبندی: توسط هر یک از پردازنده ها انجام شود.

همگام سازی: ابزاری است برای تنظیم انحصار متقابل و تنظیم حادثه ها.

مدیریت حافظه: باید سخت افزار مناسب را بهره جوید.

قابلیت اطمینان و تحمل خرابی: در برخورد با خرابی باید اقدام کند.

سازماندهی سیستم عامل های چند پردازنده ای

ازجمله وظایف سیستم عامل سرویس دهی است، در واقع سرویس های سیستم عامل چند پردازنده ای بین چند پردازنده توزیع می شوند و یا هر پردازنده یک یا چندین سرویس دارد.

۱- روش Master/Slave

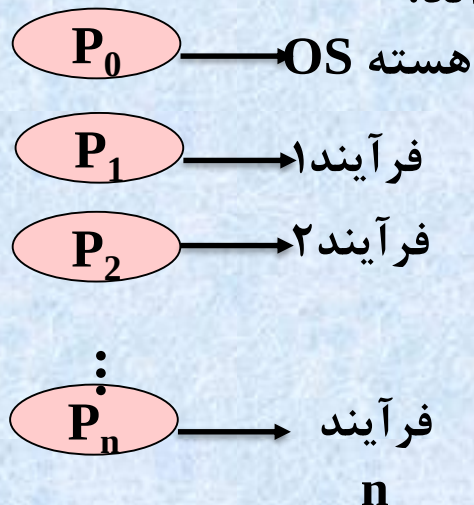
۲- روش Separate Supervisor (سرپرست مجزا)

۳- روش Floating Supervisor (روش سرپرست شناور)

روش Master/Slave

در این روش هسته بر روی یک پردازنده قرار می‌گیرد و این پردازنده فرآیندهای موجود را بر روی پردازنده‌های دیگر توزیع می‌کند. فرآیندها بصورت موازی اجرا می‌شود در نتیجه سرعت اجرا بالا رفته و زمان اجرا کاهش می‌یابد.

سیستم‌هایی مانند Windows و Unix از این دسته‌اند.



ویژگیهای روش Master/Slave

پیاده سازی و مدیریت آن بسیار ساده است.

قابلیت اطمینان پایین: با از کارافتادن یک پردازنده (پردازنده مربوط به هسته) فرآیندهای اجرایی دیگر از کار می افتند.

وجود گلوگاه (Bottleneck): پدیده ای است که در آن چند پردازنده برای اجرای وظایف خود می خواهند از یک منبع (پردازنده) استفاده کنند.

کاهش سرعت: برای فرایندهایی که نیاز به System Call دارند.

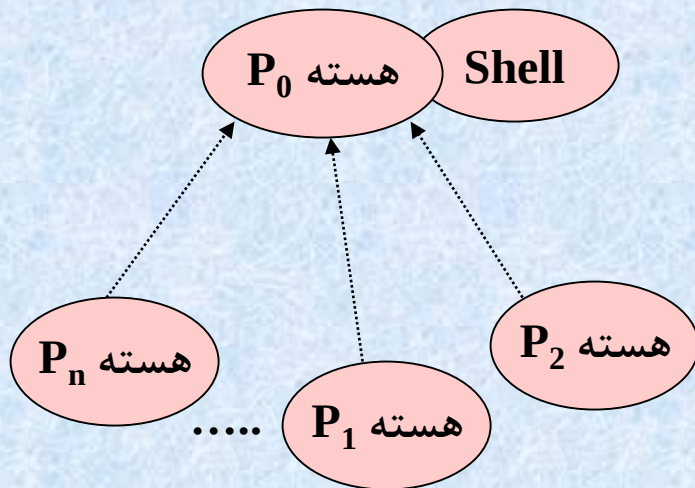
هسته همیشه در حال اجرا است.

روش سرپرست مجزا

در این حالت هسته سیستم عامل بر روی تمام پردازنده ها به صورت جداگانه قرار می گیرد اما تنها یک Shell (پوسته ارتباط با کاربر) وجود دارد که فرآیندهای دیگر هم برای ارتباط با کاربر از همین پوسته استفاده می کنند.

هسته روی تمام پردازنده ها قرار می گیرد و روی هر پردازنده چندین فرایند می تواند

اجرا شود.



ویژگی های روش سرپرست مجزا

این روش مشکلات روش Master/Slave را ندارد.

تکرار هسته سیستم عامل فضای زیادی از حافظه را اشغال می کند.
مدیریت پیچیده تری دارد.

اگر پوسته یا پردازنده دارای پوسته از کار بیفتد امکان ارتباط فرایندها با کاربر قطع می شود هرچند که پردازنده ها می توانند به کار پردازشی خود ادامه دهند.

افزایش سرعت: برای فرایندهایی که نیاز به System Call دارند هر زمانی که نیاز باشد همزمانی ایجاد می شود زیرا کدهای سیستم عامل بر روی هر پردازنده قرار می گیرد.

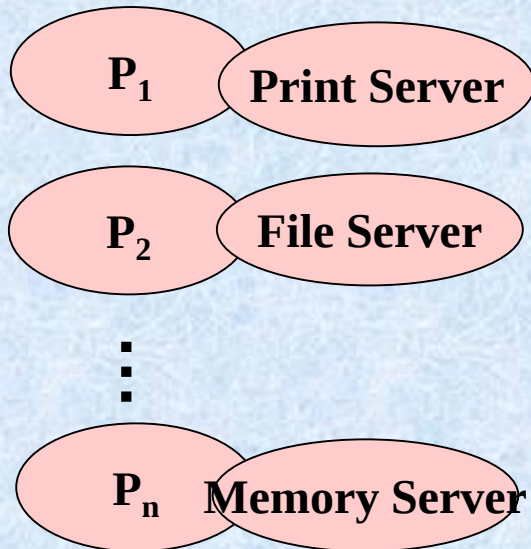
روش سرپرست شناور

این روش ترکیبی از دو روش قبل است.

ریزهسته سیستم عامل بر روی تمام پردازنده ها قرار می گیرد.

هر یک از سرویسهای سیستم عامل بر روی یکی از پردازنده ها قرار

می گیرد.



ویژگی های روش سرپرست شناور

نسبت به روش Master/Slave جای بیشتری را اشغال می کند چون علاوه بر تمام سرویسهای سیستم عامل هر پردازنده ریز هسته دارد.

نسبت به روش سرپرست مجزا حجم کمتری اشغال می کند چون سرویسها توزیع شده است.

تا حدودی از اجرای همزمان و موازی فرایندها استفاده می شود.

وظایف سیستم عامل در سیستم چندپردازنده‌ای

۱- مدیریت فرایند (زمانبندی، بن بست،)

تفاوت چندپردازنده‌ای با تک‌پردازنده‌ای در تعداد منابع پردازشی است و در نتیجه زمانبندی‌ها نیز متفاوت است چون در سیستم‌های چندپردازنده‌ای علاوه بر چگونگی تخصیص پردازنده‌ها به فرایندها، تخصیص کدامین پردازنده به کدامین فرایند نیز مورد نظر است.

۲- مدیریت I/O (چاپگر)

۳- مدیریت حافظه اولیه (RAM)

۴- مدیریت حافظه ثانویه (فایل)

۵- شبکه‌بندی

بسیار شبیه سیستم عامل تک‌پردازنده‌ای

زمان بندی (Scheduling)

هدف از زمان بندی، تخصیص پردازنده ها به فرآیندها در طول زمان است به گونه ای که هدف های سیستم برآورده شود.

الگوریتم زمان بندی

تصمیم گیری برای چگونگی اختصاص فرآیندها به پردازنده ها است نه اجرای فرآیندها

انواع زمان بندی

۱- انحصاری، غیر قابل پس گرفتن (Non-preemptive)

هنگامی که پردازنده در اختیار یک فرآیند قرار گرفت، آنقدر در اختیارش باقی می ماند تا فرآیند به اتمام برسد یا عمل I/O رخ دهد.

۲- غیر انحصاری، قابل پس گرفتن (Preemptive)

هنگامی که زمانبند سیستم عامل پردازنده را به یک فرآیند واگذار کرد، می تواند بر طبق الگوریتم زمان بندی بر خلاف میل فرآیند، پردازنده را از فرآیند باز پس گیرد.

انواع زمان بندی ۲

۱- زمان بندی قطعی **Deterministic**

هر بار که الگوریتم زمانبندی برای تعداد فرایند یا تعداد کار ثابت اجرا می شود یک جواب حاصل می شود. مانند SJF

۲- زمان بندی غیر قطعی **Non deterministic**

در هر بار اجرای الگوریتم زمانبندی ممکن است جواب های متفاوتی بدست آید.
مانند الگوریتم های ژنتیک

انواع زمان بندی ۳

۱- زمان بندی ایستا (Static)

به گونه ای که زمان ورود کارها به سیستم و مدت زمان انجام آنها قبل از شروع برای الگوریتم زمانبندی مشخص است و اگر کار جدیدی وارد سیستم شود تا پایان اجرای کلیه کارهای قبلی در زمانبندی دخالت داده نمی شود.

۲- زمان بندی پویا (Dynamic)

ممکن است مشخصات کلیدی کارها از قبیل مدت زمان انجام آنها در زمان اجرای برنامه تغییر یابد. اگر کار جدیدی وارد سیستم شود در هنگام اجرای کارهای قبلی ممکن است در زمانبندی دخالت داده شود.

الگوریتمهای زمانبندی پویا در سیستم چند پردازنده ای

۱- اشتراک بار Load Sharing

۲- زمانبندی گروهی

۳- اختصاص پردازنده به فرآیند

۴- جدا سازی سیاست از مکانیزم زمانبندی

الگوریتم زمانبندی اشتراک بار

فرآیندها وارد صف مشترک مابین پردازنده‌ها (اشتراک بار) می‌شوند.

پردازنده‌های بیکار آنها را از صف برمی‌دارند و اجرا می‌کنند.

پس از اتمام زمان در نظر گرفته شده برای هر یک از فرآیندها (زمانبندی قابل پس

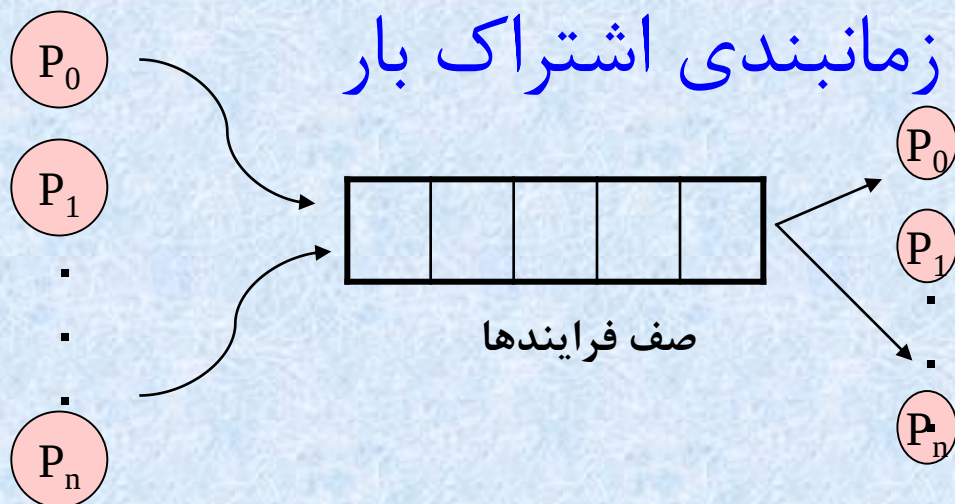
گرفتن) یا اتمام فرایند یا انجام عمل I/O

(زمانبندی غیر قابل پس گرفتن) فرآیند دوباره به انتهای صف می‌رود.

پردازنده‌ای که از انجام آن فارغ شده فرآیند بعدی موجود در صف را برمی‌دارد و

انجام می‌دهد و به همین ترتیب...

ویژگی های الگوریتم زمانبندی اشتراک بار



همیشه پردازنده ها بطور متوازن در حال کارند و معمولا پردازنده بیکار وجود ندارد.

صف مشترک را هرگونه می توان پیاده سازی نمود و به روشهای مختلف زمان بندی کرد.

مدیریت صف مشترک ساده است.

cache در نظر گرفته شده برای هر یک از پردازنده ها جهت ذخیره اطلاعات یا کدهای فرایندها

قبلی اجرا شده روی پردازنده ممکن است ارزشی نداشته باشد.

صف مشترک بخش بحرانی می شود و سبب ایجاد گلوگاه می شود.

الگوریتم زمانبندی گروهی

گروهی از فرایندها که با هم ارتباط بیشتری دارند یا یک فرایند به همراه نخهای آن با یکدیگر زمانبندی شده و همزمان به گروهی از پردازنده ها تخصیص داده شده یا از گروه گرفته می شود.

چون به صورت موازی اجرا می شود باعث افزایش سرعت می شود و کمتر مسدود می شود بخاطر ارتباطاتی که با یکدیگر دارند.

الگوریتم زمانبندی اختصاص پردازنده به فرآیند

گروهی از فرایندها که با هم ارتباط بیشتری دارند یا یک فرایند به همراه نخهای آن همزمان به گروهی از پردازنده ها تخصیص داده می شوند. هیچگاه فرآیند از پردازنده پس گرفته نمی شود مگر اینکه فرآیند تمام شود.

سرعت اجرا بالا می رود چون تعویض فرایند وجود ندارد. عیب بزرگ این روش بی کار بودن پردازنده و کاهش بهره وری پردازنده است.

در یک سیستم چند پردازنده ای پردازنده ها بی اهمیت هستند.

جداسازی سیاست از مکانیزم زمانبندی

یک ایده کلی است و ربطی به زمانبندی ندارد.

باید میان سیاست زمان بندی و مکانیزم آن تفاوت قائل شد.

سیاست نسبت به مکانیزم سطح بالاتر است.

مکانیزم طریقه پیاده سازی سیاست است.

مثلا سیاست زمان بندی مشخص می کند که زمان بندی به صورت پویا انجام شود یا ایستا، اما مکانیزم آن مشخص می کند که برای پیاده سازی سیاست در نظر گرفته شده از چه روشی استفاده نمود.

الگوریتم های زمان بندی ایستا

معیارها در الگوریتم های زمان بندی ایستا

معیار های زمانبندی برای سیستمهای تک پردازنده ای با سیستمهای چند پردازنده ای فرق می کند.

- زمان اجرا
- متوسط زمان پاسخ
- متوسط زمان انتظار
- بهره وری
- توان عملیاتی
- ...

معیار مهم زمانبندی در سیستمهای چند پردازنده ای

Total finish time یکی از معیارهای زمانبندی ایستا در سیستمهای چند پردازنده ای است.

حداکثر زمانی که برای اجرای کل کارها (Taskها) طول می کشد را **Total finish time** می گویند.

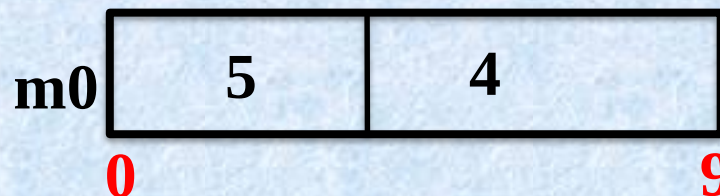
در ادبیات مقاله هایی که در بحث زمانبندی ارائه می شوند از واژه **makespan** برای نشان دادن حداکثر طول زمان اجرا استفاده می شود.

مثال

T1	T2	T3
5	4	7

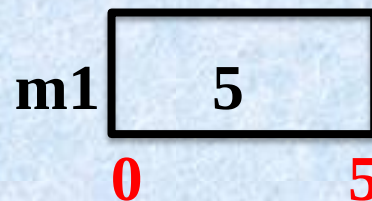
سیستم دو پردازنده ای

روش اول



TFT = 9

روش دوم



TFT = 11

سیستم های عامل پیشرفته

معیارهای زمان بندی

$$\text{میزان موازی سازی} = \frac{\text{جمع کل زمانهای اجرا}}{\text{بزرگترین زمان در TFT}}$$

$$\text{بهره وری CPU} = \frac{\text{زمان پردازش}}{\text{زمان کل}}$$

$$\text{توان عملیاتی} = \frac{\text{تعداد کار تکمیل شده در واحد زمان}}{\text{زمان کل}}$$

الگوریتمهای زمانبندی ایستا در چند پردازنده ای

- الگوریتم زمانبندی ایستا قابل پس گرفتن
Bin Packing

- الگوریتم زمانبندی ایستا غیر قابل پس گرفتن

SPT (Shortest Processing Time)

LPT (Largest Processing Time)

RLPT (Reverse Largest Processing Time)

LSPT (Largest Shortest Processing Time)

الگوریتم های اکتشافی

الگوریتم زمانبندی ایستا قابل پس گرفتن Bin Packing

یک الگوریتم قابل پس گرفتن است. در این روش به هر پردازنده حداکثر زمانی به

اندازه FT_{opt} داده می شود که عبارت است از:

$$FT_{opt} = \text{Max} \left\{ \frac{\sum T_i}{m}, \text{Max}(T_i) \right\}$$

طول بزرگترین کار موجود

مجموع طول کارها تقسیم بر تعداد پردازنده ها

سپس کارها به هر ترتیبی (مثلاً ترتیب بزرگی به کوچکی) به پردازنده ها نسبت

داده می شوند.

مثال الگوریتم Bin Packing

زمان کارهای موجود در سیستم عبارتند از:

$$T_i = \{15, 11, 8, 7, 4, 2, 1, 1\}$$

مجموعه کارها

$$M = 2$$

تعداد پردازنده ها

$$Ft_{opt} = \text{Max} \{ 15, (15+11+8+7+4+2+1+1)/2 \}$$

$$= \text{Max} \{ 15, 24.5 \} = 24.5$$

M1	<table><tr><td>15</td><td>9.5/11</td></tr></table>	15	9.5/11
15	9.5/11		

24.5

M2	1.5/11	8	7	4	2	1	1
----	--------	---	---	---	---	---	---

24.5

الگوریتمهای زمانبندی ایستا غیرقابل پس گرفتن

SPT (Shortest Processing Time)

LPT (Largest Processing Time)

RLPT (Reverse Largest Processing Time)

LSPT (Largest Shortest Processing Time)

الگوریتم زمانبندی SPT

Shortest Process Time یک الگوریتم غیرقابل پس گرفتن است.

۱- کارها (Taskها) بر اساس زمان اجرایشان به صورت صعودی مرتب می شوند.

۲- کوچکترین کار در اولین پردازنده آزاد قرار می گیرد و کار بعدی در پردازنده آزاد بعدی الی آخر تا تمام کارها زمانبندی شوند.

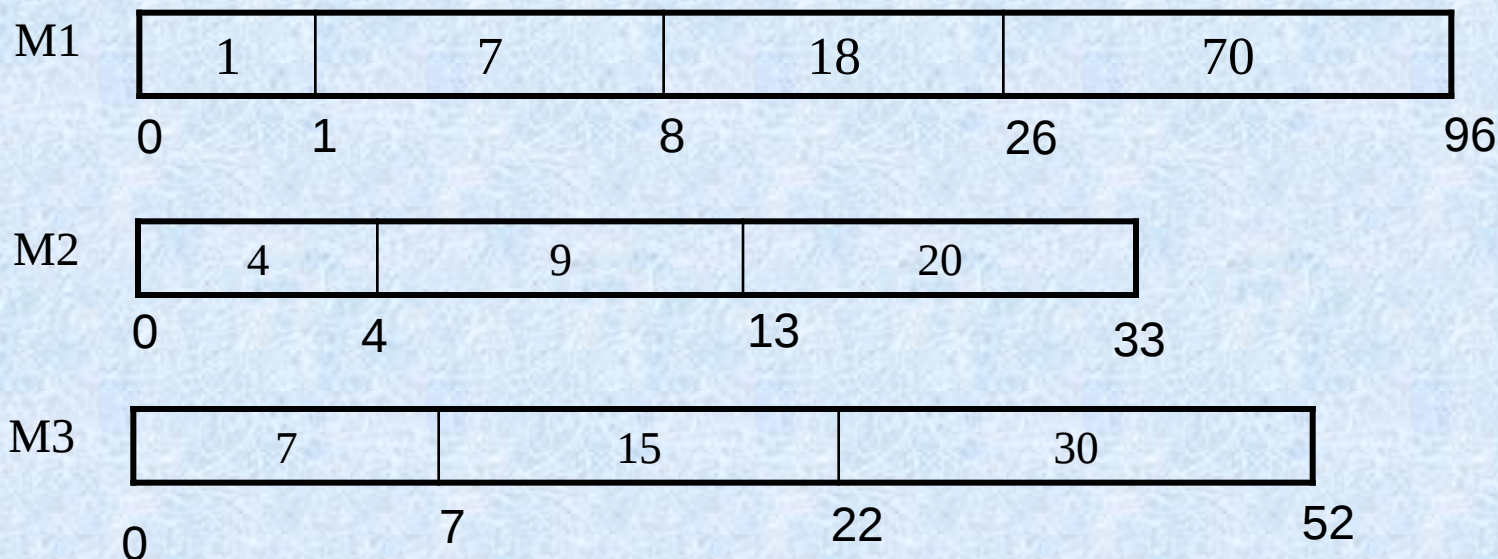
مزیت؟ عیب؟

مثال الگوریتم زمانبندی SPT

۱۰ کار موجود در یک سیستم با ۳ پردازنده عبارتند از:

$$T_i = \{1, 4, 7, 7, 9, 15, 18, 20, 30, 70\}$$

$$M = 3$$



$$\text{TFT (SPT)} = 96$$

الگوریتم زمانبندی LPT

Longest Processing Time یک الگوریتم غیرقابل پس گرفتن است.

۱- کارها بر اساس زمان اجرایشان به صورت نزولی مرتب می شوند.

۲- بزرگترین کاردر اولین پردازنده آزاد قرار می گیرد و کار بعدی در پردازنده آزاد بعدی الی آخر تا تمام کارها زمانبندی شوند.

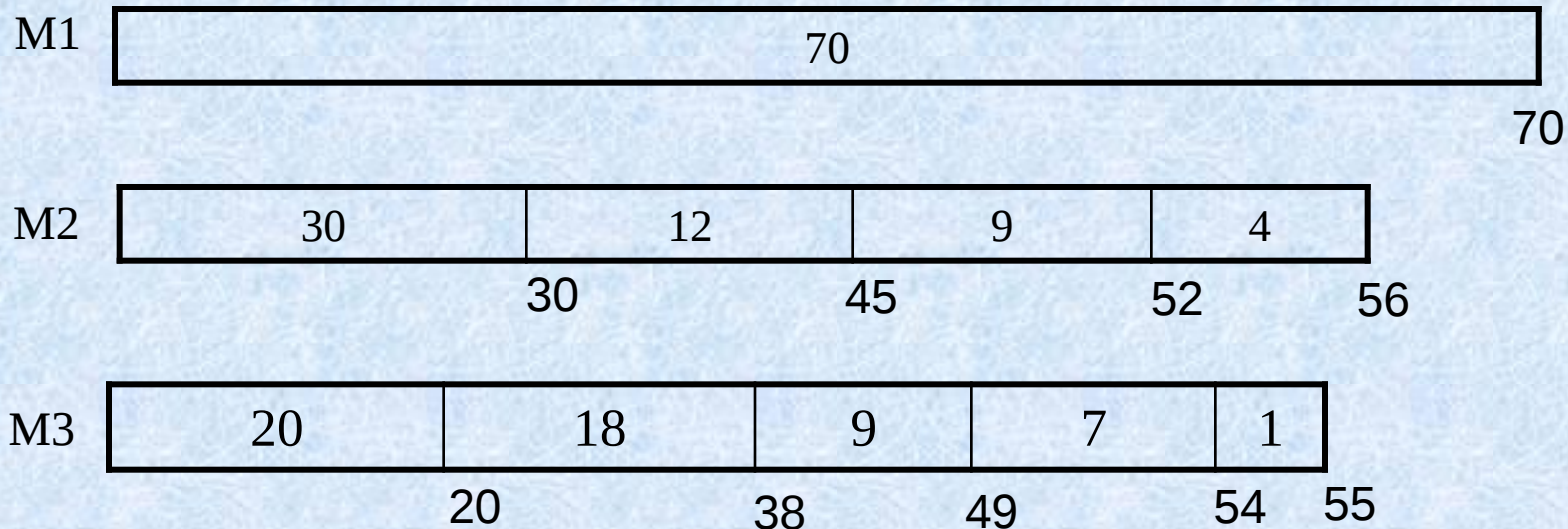
LPT چه مزیتی نسبت SPT به دارد؟

مثال الگوریتم زمانبندی LPT

۱۰ کار موجود در یک سیستم با ۳ پردازنده عبارتند از:

$$T_i = \{1, 4, 7, 7, 9, 15, 18, 20, 30, 70\}$$

$$M = 3$$



$$\text{TFT (LPT)} = 70$$

الگوریتم زمانبندی RLPT

Reverse Longest Processing Time یک الگوریتم غیرقابل پس گرفتن است.

- ۱- کارها بر اساس زمان اجرایشان به صورت نزولی مرتب می شوند.
- ۲- بزرگترین کار در اولین پردازنده آزاد قرار می گیرد و کار بعدی در پردازنده آزاد بعدی الی آخر تا تمام کارها زمانبندی شوند.
- ۳- برای هر پردازنده لیست کارهای اجرایی معکوس می شود.

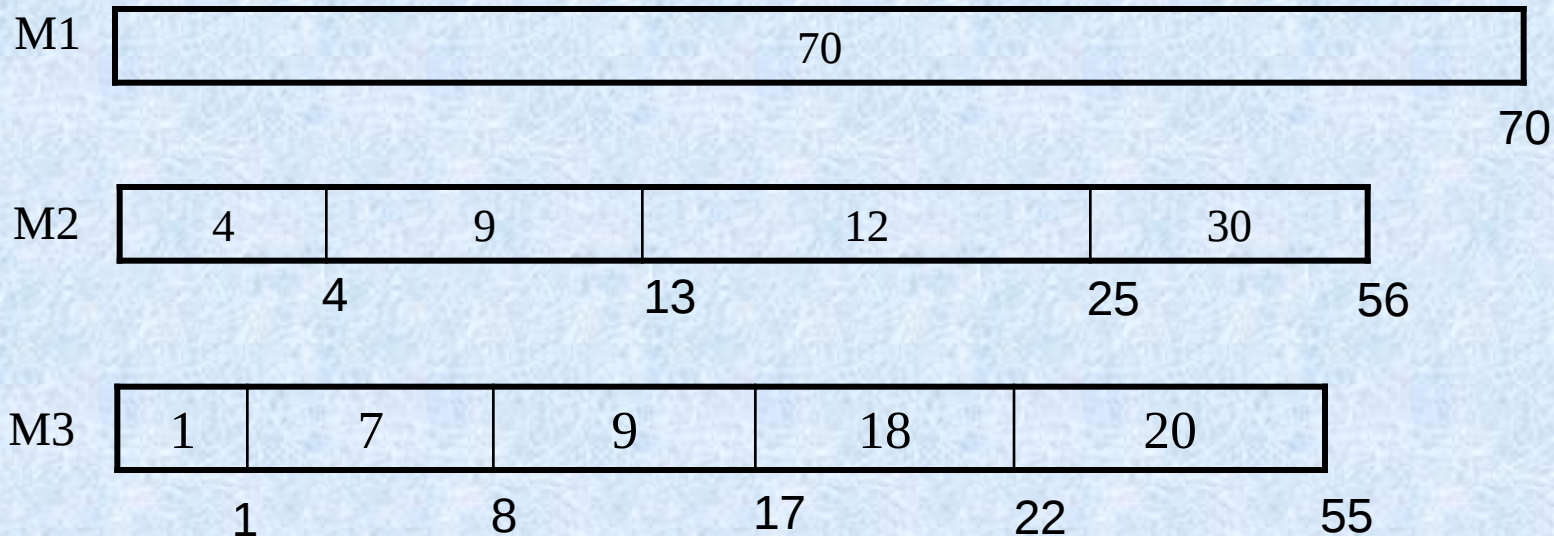
RLPT چه مزیتی نسبت LPT به دارد؟

مثال الگوریتم زمانبندی RLPT

۱۰ کار موجود در یک سیستم با ۳ پردازنده عبارتند از:

$$T_i = \{1, 4, 7, 7, 9, 15, 18, 20, 30, 70\}$$

$$M = 3$$



$$\text{TFT (RLPT)} = 70$$

الگوریتم زمانبندی LSPT

Longest Shortest Processing Time یک الگوریتم غیرقابل پس گرفتن است.

این الگوریتم تلفیقی از الگوریتم های SPT و LPT است.

۱- کارها بر اساس زمان اجرایشان به صورت صعودی مرتب می شوند.

۲- سپس به ترتیب کارها به دسته های M تایی تقسیم می شوند (M تعداد پردازنده ها است).

۳- به ترتیب M کار بصورت LPT زمانبندی می شوند. یعنی بزرگترین کار در اولین پردازنده آزاد قرار می گیرد و کار بعدی در پردازنده آزاد بعدی الی آخر تا تمام M کارها زمانبندی شوند.

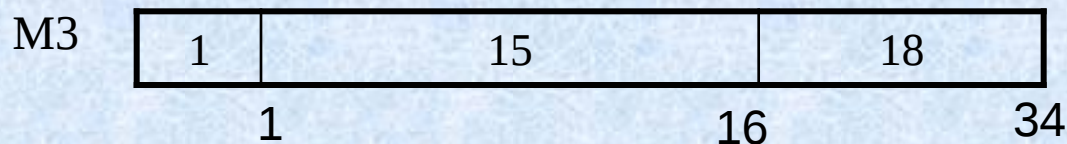
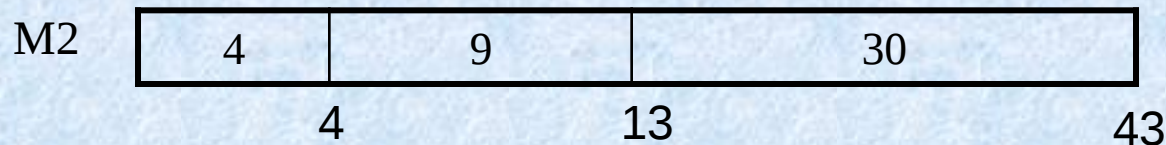
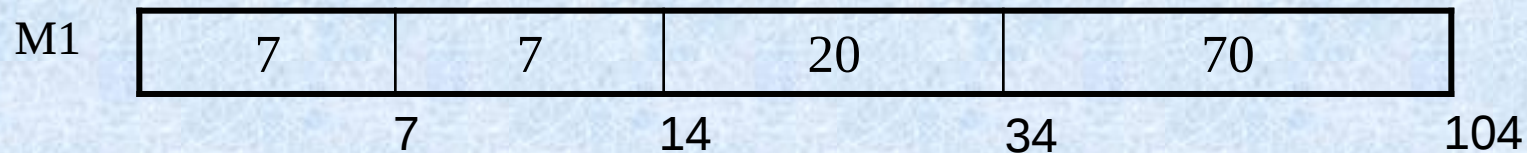
اگر تعداد کارها مضربی از تعداد پردازنده ها باشد؟
سیستم های عامل پیشرفته

مثال الگوریتم زمانبندی LSPT

۱۰ کار موجود در یک سیستم با ۳ پردازنده عبارتند از:

$$T_i = \{1, 4, 7, 7, 9, 15, 18, 20, 30, 70\}$$

$$M = 3$$



$$\text{TFT (LSPT)} = 104$$

الگوریتم های زمانبندی اکتشافی

روشهای اکتشافی برای کاربرد های خاص ایجاد می شوند:

MAX-MIN و OLB و MET و ...

تمرین

روش های مبتنی بر روش های هوشمند (اکثراً غیرقطعی هستند)

الگوریتم ژنتیک و PSO و کلونی مورچه ها و ...

الگوریتم های مبتنی بر روش های هوشمند یک روش جستجو هستند که از روش های

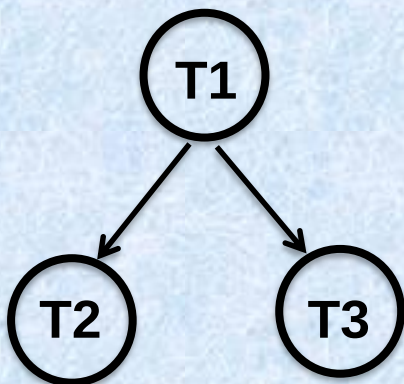
کلاسیک و قطعی بهتر هستند ولی تضمین یافتن زمانبندی بهینه را نمی دهند.

گراف جهت دار غیر حلقه بدون دور

بعضی از مواقع ترتیب و تقدم اجرای کارها مهم است که

Directed Acyclic Graph (DAG) ترتیب اجرا را نشان می دهد و

برای سیستمهای چند پردازنده ای و توزیع شده نیز قابل استفاده است.

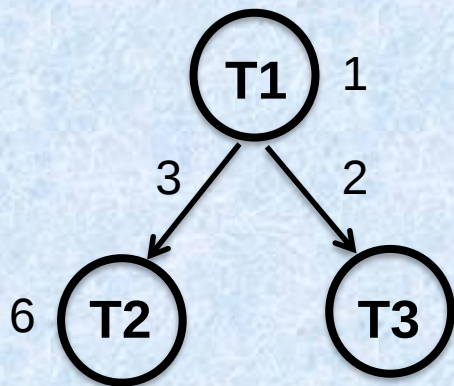


T2, T3 در صورتی اجرا می شود که T1 تمام شده باشد

گراف وزندار جهت دار غیر حلقه بدون دور

بعضی از مواقع علاوه بر ترتیب و تقدم اجرای کارها هزینه ارتباط بین کارها نیز مهم است که

Weighted Directed Acyclic Graph (WDAG) ترتیب اجرا و هزینه ارتباط بین کارها را نشان می دهد.



زمان اجرای T1, T2, T3 به ترتیب ۱، ۶ و ۵ است.

T3 حداقل ۲ واحد زمان بعد از اتمام T1 بایستی آغاز به اجرا نماید.