تمرین تئوری سری دوم سیستم عامل

سارا سادات یونسی / 98533053

Multi-thread/Single-thread1

(آ) سناریویی مثال برنید که در آن یک راه حل multithread که ازتعدادی kernel thread استفاده می کند ,عملکرد بهتری را در مقایسه با راه حل در یک سیستم تک پردازنده ای ارائه میکند. شرایط لازم برای ایجاد این سناریو را توضیح دهید

(ب) دو مثال برنامه نویسی ارائه کنید که در آن ها استفاده از multithreading عملکرد بهتری در مقایسه با single threaded ارئه نمی کند .

پاسخ

(الف) برای مثال برای محاسبه ضرب دو ماتریس در یکدیگر زیرا ضرب دو سط و ستون در هم به یکدیگر ربطی ندارند و روی multithread بهتر از single thread جواب میدهد.

The process of executing multiple threads simultaneously is known as multithreading. Some examples where multi threading improves performance include:

1--Matrix multiplication — Individual rows and columns of the matrices can be multiplied in separate threads, reducing the wait time of the processor for addition.

2-- UI updates — We can render some UI elements such as a view or images on a background thread so that the main thread does not block the view, causing performance issues and noticeable lags.

3-- A Web server that services each request in a separate thread.

4-- A parallelized
application such as matrix multiplication where different parts of the matrix may be
worked on in parallel.

5-- An interactive GUI program such as a debugger where a
thread is used to monitor user input, another thread represents the running application,
and a third thread monitors performance.

(ب) تخصیص مموری کار سریعی است و اجرای آن با multithread با توجه به overhead ایجاد شده به مراتب کندتر خواهد شد.

یا وقتی که همزمان چند تا thread روی یک دستگاه io کار کند.

Any kind of sequential program is not a good candidate to be threaded. An example of this is a program that calculates an individual tax return. (2) Another example is a "shell" program such as the C-shell or Korn shell. Such a program must closely monitor its own working space such as open files, environment variables, and current working directory.

If we are running a trivial program (constant time complexity) in a separate thread, the overhead of creating the threads exceeds the tasks performed by them, thus decreasing performance when compared to a single threaded alternative.

Some examples include:

Trivial operations on a list of numbers — Multi threading won't speed up the operations since the time taken by the operations is constant, and other elements of the list may or may not wait for the previous to finish.

Allocating memory to a set of data variables — Allocating memory is a very fast task, and the overhead of creating multiple threads to process separate blocks of variables exceeds the performance gained by multi threading.

Thread interference2

(آ) تابع نوشته شده چه کاری انجام میدهد؟

(ب) آیا اجرای همزمان thread های A , B مشکلی در محاسبات تابع به وجود می آورد؟ توضیح دهید.

(ج) برخی از کامپایلر ها مانند gcc که کد های نوشته شده را optimize میکنند، تابع را به شکل زیر در می آورند

آیا در این حالت اجرای همزمان thread های A و B مشکلی برای محاسبات تابع به وجود می آورد؟ توضیح دهید

پاسخ

آ) تعداد اعداد مثبت داخل لیست ورودی را می شمارد و درglobal_vare ذخیره میکند.

(ب) در یک لحظه خاص ممکن است ولی در پایان اجرا خیر، با توجه به اینکه در thread A برای محاسبه مقدار خروجی به قطعه کد داخل thread B نیاز داریم و این فرآیندها لزوما به صورت پشت سر هم اجرا نمیشوند، تغییرات ممکن است دیرتر اعمال شوند، پس در بعضی لحضات ممکن است که خروجی درست نباشند، ولی در پایان که همه فرایندها انجام شدند، خروجی هم درست خواهد بود

(ج) خیر چون به یکدیگر در طول برنامه نیاز ندارند.

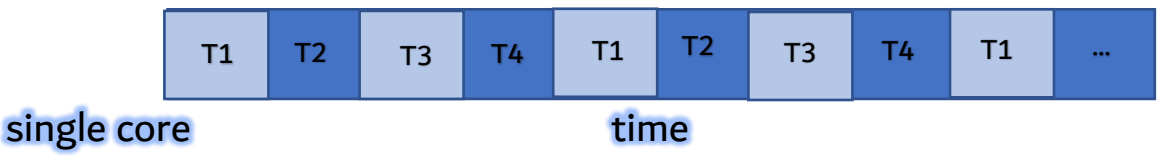
Concurrency Vs. Parallelism3

پردازش موازی(Parallel) و همروند(Concurrent) را توضیح دهید و با رسم نمودار روند اجرای هر کدام را با یکدیگر مقایسه کنید.

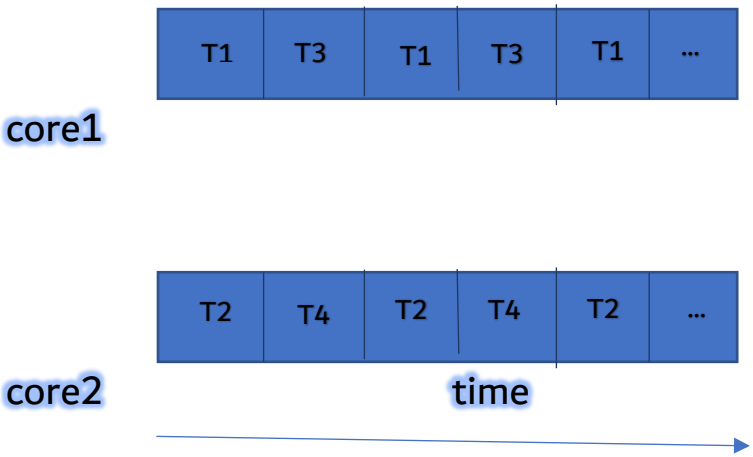
پاسخ

در پردازش موازی(Parallel) در هر لحظه می توانند چند تسک مختلف بطور همزمان درحال اجرا باشند؛ بطوری که تسک های موجود روی core ها تقسیم می شون د و هر تسک روی یک core جداگانه اجرا می شود. به همین علت، در سیستم های core single پردازش موازی معنایی ندارد .در پردازش همروند(Concurrent) برخالف پردازش موازی، روی یک core میتواند بیشتر از یک تسک اجرا شود اما در هر لحظه فقط یک تسک باید در حال اجرا باشد. می توانیم با استفاده از sharing timeو switch های متوالی بین تسک ها روی یک core چندین تسک را در حال اجرا داشته باشیم

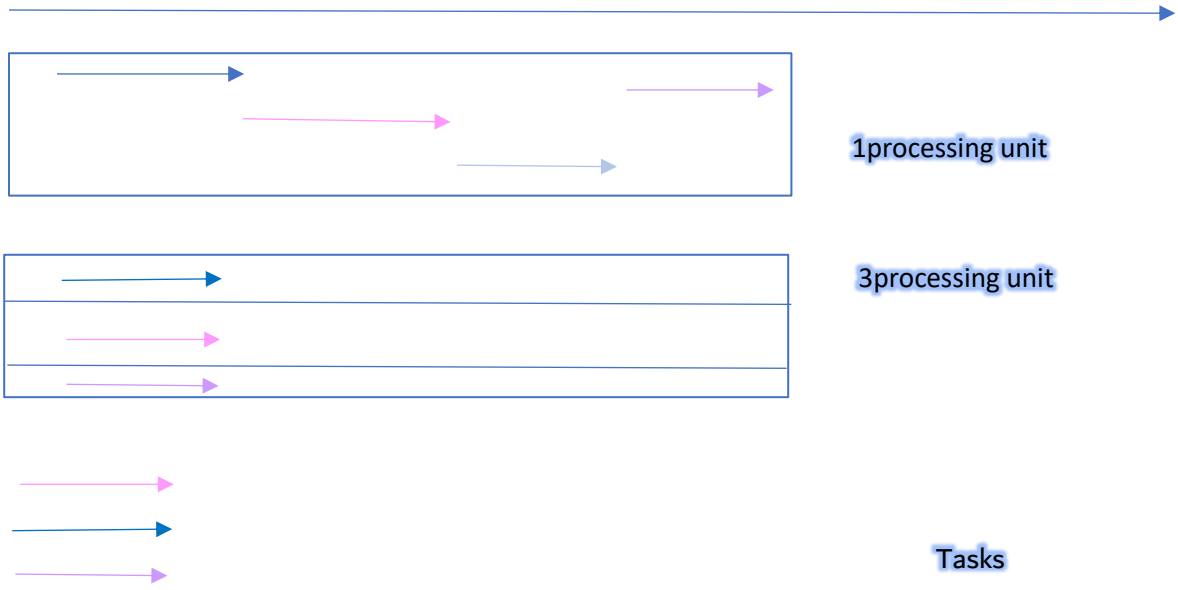
.نمودارهای زیر می توانند درک بهتری از تعاریف باال را ارائه دهند:



شکل 1 : پردازش همروند



شکل 2 : پردازش موازی



شکل 3 :مقایسه ی پردازش همروند (بالا) و پردازش موازی (پایین)

Data Parallelism Vs. Task Parallelism 4

Parallelism Data وParallelism Task را توضیح داده و حداقل سه مورد از تفاوت های آن ها را بیان کنید.

پاسخ

Data parallelism

اجرای همزمان (concurrent) یک تسک روی چندهسته (cpu core) است. که به منظور محاسبه سریع تر انجام میشود .برای مثال محاسبه مجموع اعداد داخل یک لیست که با داشتن دو هسته نیمه اول را یکی و نیمه دوم را دیگری انجام میدهد.

موازی سازی داده به معنای اجرای همزمان یک کار روی هر هسته محاسباتی چندگانه است.

یابید مثالی بزنیم، محتویات یک آرایه به اندازه N را جمع کنیم. برای یک سیستم تک هسته ای، یک رشته به سادگی عناصر [0] را جمع می کند. . . [N - 1].با این حال، برای یک سیستم دو هسته ای، رشته A که روی هسته 0 اجرا می شود، می تواند عناصر [0] را جمع کند... [N/2 – 1] و در حالی که thread B که روی هسته 1 اجرا می شود می تواند عناصر [N/2] راجمع کنذ ....[N-1]بنابراین دو رشته به صورت موازی روی هسته‌های محاسباتی جداگانه اجرا می‌شوند.

Task parallelism

به معنای اجرای همزمان (concurrent) تسک های مخلف روی چند هسته (cpu core) است.

موازی کاری به معنای اجرای همزمان وظایف مختلف در چندین هسته محاسباتی است.

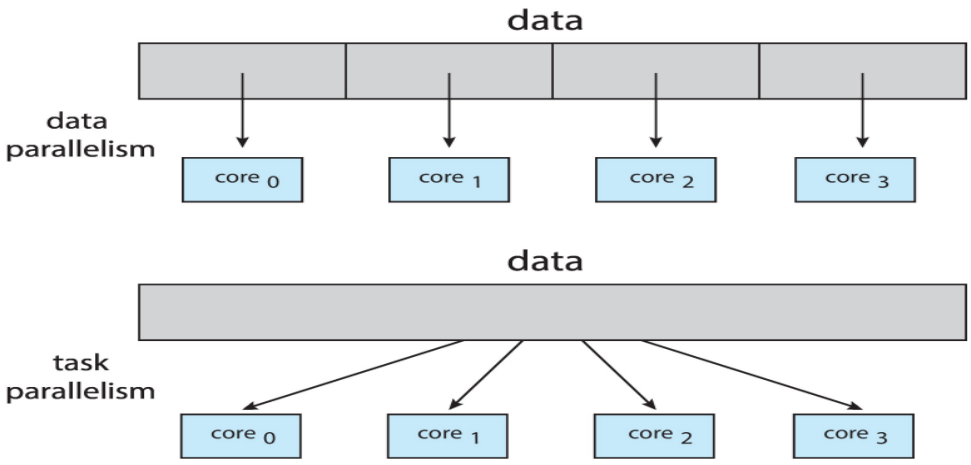
مثال بالا را دوباره در نظر بگیرید، یک مثال از موازی کاری ممکن است شامل دو رشته باشد که هر یک عملیات آماری ها به طور موازی روی هسته های محاسباتی جداگانه threadمنحصر به فرد را روی آرایه عناصر انجام می دهد. مجدداً کار می کنند، اما هر کدام یک عملیات منحصر به فرد را انجام می دهند.

تفاوت ها

 Data parallelism – distributes subsets of the same data across multiple cores, same operation on each

 Task parallelism – distributing threads across cores, each thread performing unique operation

Data Parallelisms	Task Parallelisms
1. Same task are performed on different subsets of same data.	1. Different task are performed on the same or differen data.
2. Synchronous computation is performed.	2. Asynchronous computation is performed.
3. As there is only one execution thread operating on all sets of data, so the speedup is more.	3. As each processor will execute a different thread or process on the same or different set of data, so speedı is less.
4. Amount of parallelization is proportional to the input size.	4. Amount of parallelization is proportional to the numb of independent tasks is performed.
5. It is designed for optimum load balance on multiprocessor system.	5. Here, load balancing depends upon on the e availability of the hardware and scheduling algorithms like static and dynamic scheduling.



Multithreading Models 5

یک سیستم multicore و یک برنامه multithreaded را که با استفاده از مدل many-to-many threading نوشته شده است، درنظر بگیرید. فرض کنید تعداد thread level-user ها در برنامه بیشتر از تعداد هسته های محاسباتی در سیستم باشد. در هریک از حالت های زیر Performance را مورد بررسی قرار دهید

(آ) تعداد thread kernel های اختصاص داده شده به برنامه از تعداد هسته های پردازشی کمتر باشد .

(ب) تعداد thread kernel های اختصاص داده شده به برنامه با تعداد هسته های پردازشی برابر باشد .

(ج) تعداد thread kernel های اختصاص داده شده به برنامه از تعداد هسته های پردازشی بیشتر باشد اما از تعداد thread level-userها کمتر باشد .

پاسخ

(آ)در این حالت بعضی از پردازنده ها بیکار می مانند، ز یرا scheduler فقط thread level-kernel ها را به پردازنده نگاشت میکند، نه user-level thread ها را

(ب) در این حالت این امکان وجود دارد که همه ی پردازنده ها همزمان به کار گرفته شوند. اما اگر در یکی از پردازنده ها، kernel threadبه هر دلیلی (fault page یا درحین انجام یک (system call مختل شود، آن پردازنده بیکار می ماند

(ج)در این حالت همواره همه ی پردازنده ها به کار گرفته می شوند. چون حتی اگر یکی از kernel thread ها هم مختل شو ند، یک kernel thread د یگر جایگز ین آن می شود که این مورد، استفاده ی بهتری از سیستمهای چند پردازنده ای را فراهم میکند