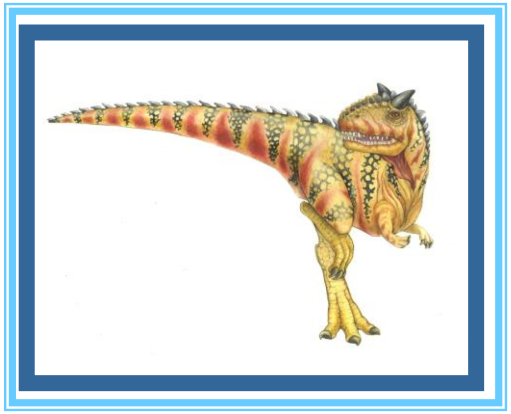
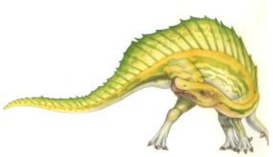
Chapter 4: Threads

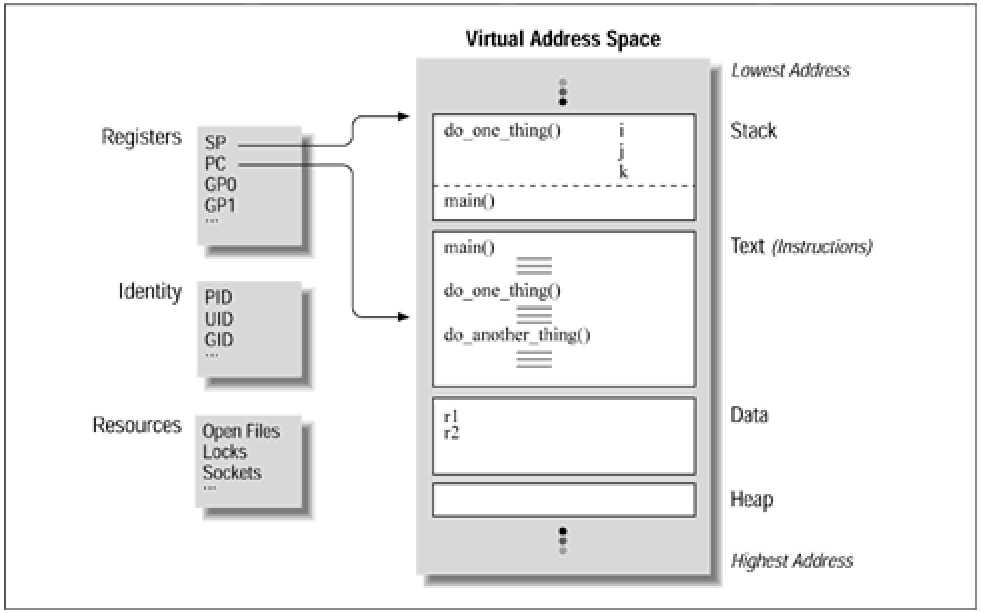


Objectives

- To introduce the notion of a thread — a fundamental unit of CPU utilization that forms the basis of multithreaded computer systems
- To discuss the APIs for the Pthreads, Win32, and Java thread libraries
- To examine issues related to multithreaded programming

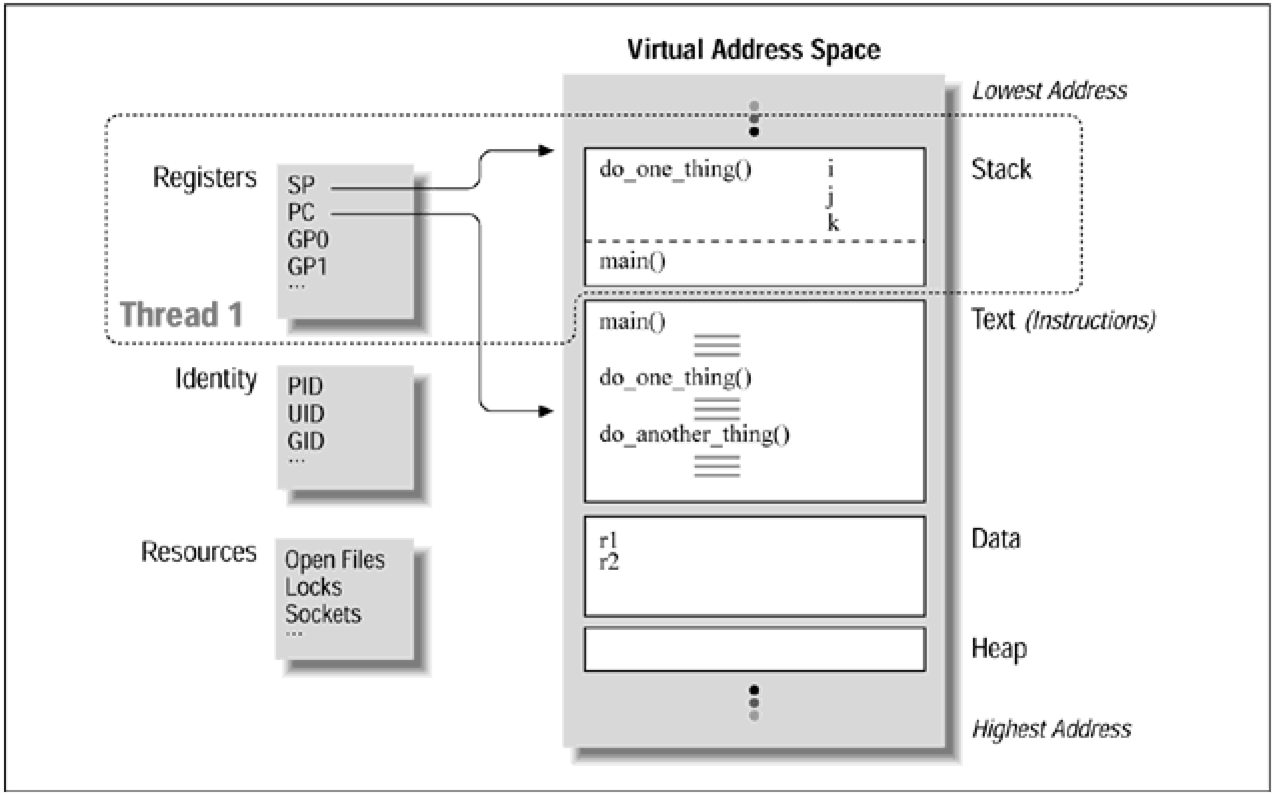


Process Revisited



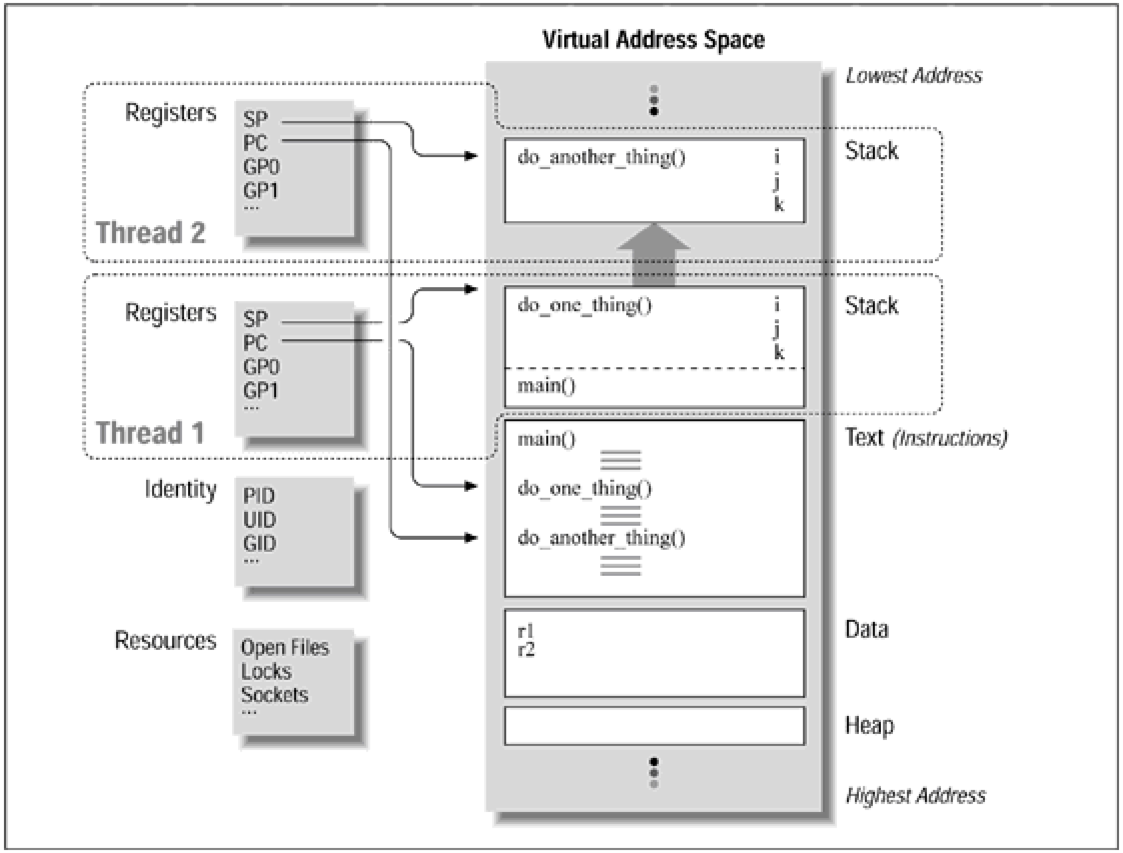
http://maxim.int.ru/bookshelf/PthreadsProgram/htm/r_6.html

Single Thread



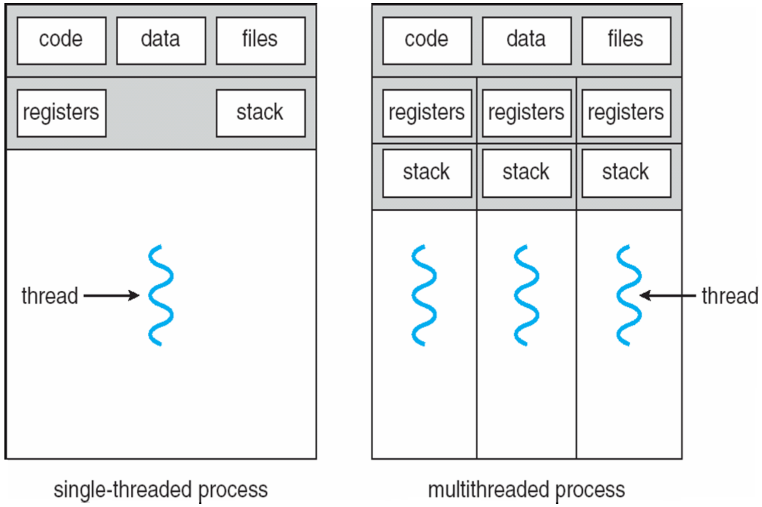
http://maxim.int.ru/bookshelf/PthreadsProgram/htm/r_6.html

Multiple Threads



http://maxim.int.ru/bookshelf/PthreadsProgram/htm/r_6.html

Single and Multithreaded Processes



1. What is shared?

2. How many PCBs?

Word Processor Example

1. Thread 1 - display graphics
2. Thread 2 - respond to keystrokes
3. Thread 3 - spelling & grammar check

Kernel Example

1. Thread 1 - manage devices
2. Thread 2 - manage memory
3. Thread 3 - manage interrupts

Solaris - set of kernel threads for interrupt handling

Benefits of Multithreaded Programming

- Responsiveness - a program can continue running even if part is blocked
  1. What is blocked?
  2. Give an example of running while a portion is blocked.

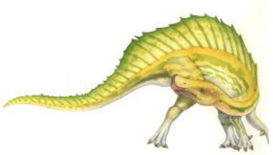
- Resource Sharing
  1. How do processes share resources vs threads?

- Economy
  1. Compare process vs thread creation

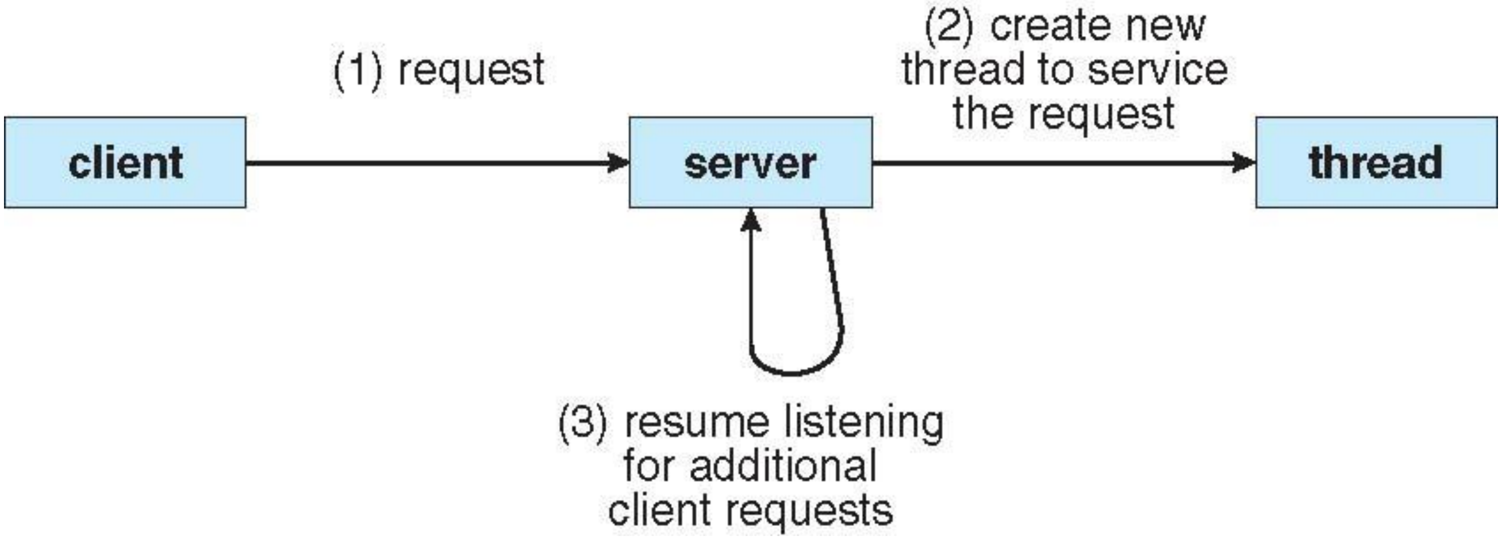
- Scalability
  1. How can multithreading be even more beneficial as the # of cores increase?

Multicore Programming

- Multicore systems putting pressure on programmers, challenges include
  - **Dividing activities**
  - **Balance**
  - **Data splitting**
  - **Data dependency**
  - **Testing and debugging**



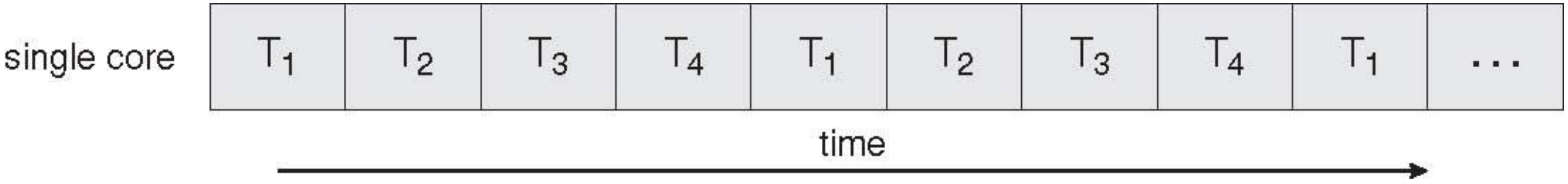
Multithreaded Server Architecture


(1) request
(2) create new thread to service the request
client
server
thread
(3) resume listening for additional client requests

Concurrent vs Parallel Execution

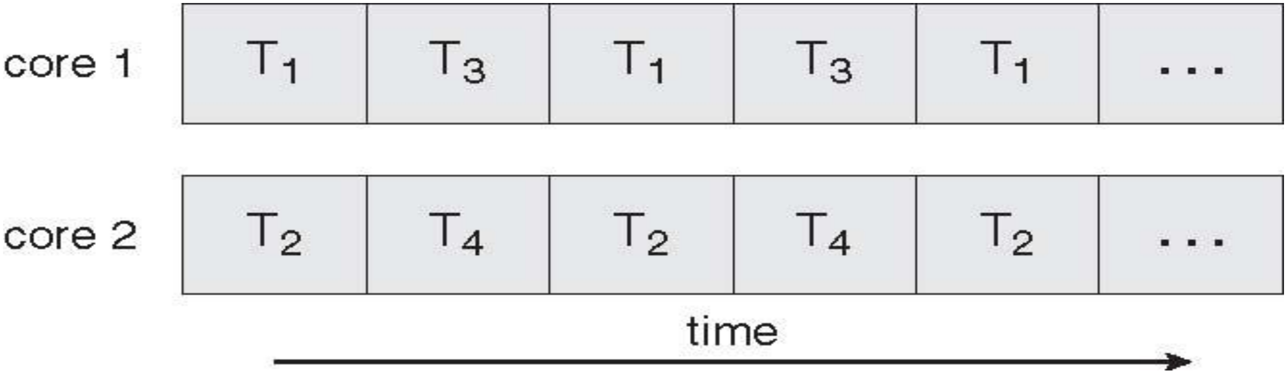
Book Definiton which is not universally accepted:

1. concurrency - supports more than one task by allowing all tasks to make progress



single core	T_1	T_2	T_3	T_4	T_1	T_2	T_3	T_4	T_1	. . .

2. parallelism - more than one task can be executing at once



core 1	T_1	T_3	T_1	T_3	T_1	. . .
core 2	T_2	T_4	T_2	T_4	T_2	. . .