Operating Systems 2-7029110

Lecture 2 - Multi-Programming

Arkady (אהרן) Gorodischer

Ariel University
Computer Science Department

CPU & an Interupt

Threads and Context switch

Definition

- Multi-Programing is an ability to run several threads in a (pseudo) parallel manner

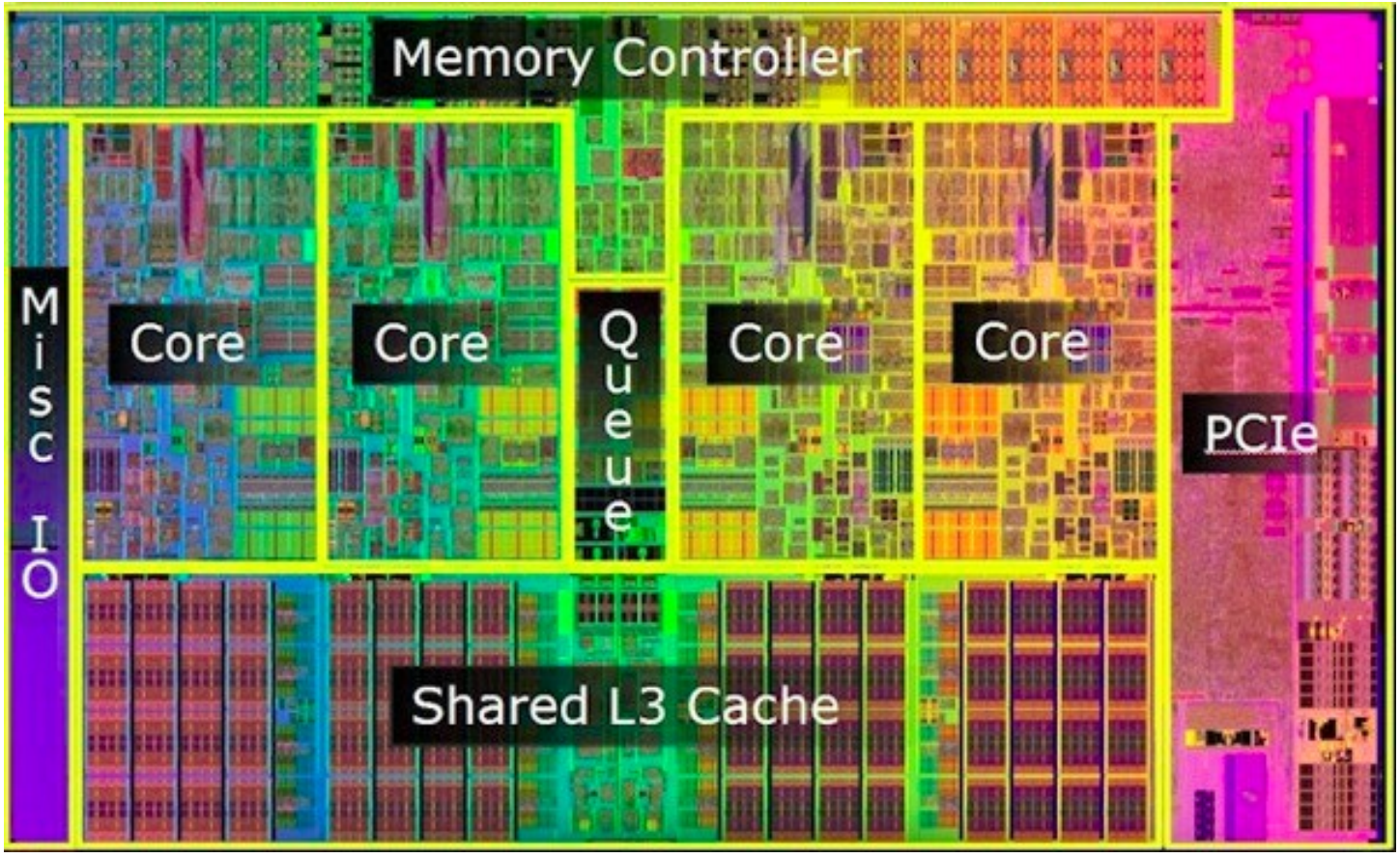
Why is this ?

- Why should we do multi-programming, and not concentrate one the main task ?
- Because we can :) CPU is faster than a human
- Business demand: Web server want to serv as much as posible
- User demand: we want youtube +word + chmore, and some email notofications too

How can we do it ?

- The way for multi-programming – Context Switch.
- Context switch it’s a way to jungle between processes,by doing the next:
  - Say to CPU to stop running the task and mark the task as stopped
  - Save PC, SP, FP and other registers for the task
  - Load PC, SP, FP and other registers from the newcoming task
  - Optionally, flash the translation lookaside buffer (TLB) on memory managment unit (MMU) and/or update caches
  - Update the new task as being running
  - Start executing new process by saying

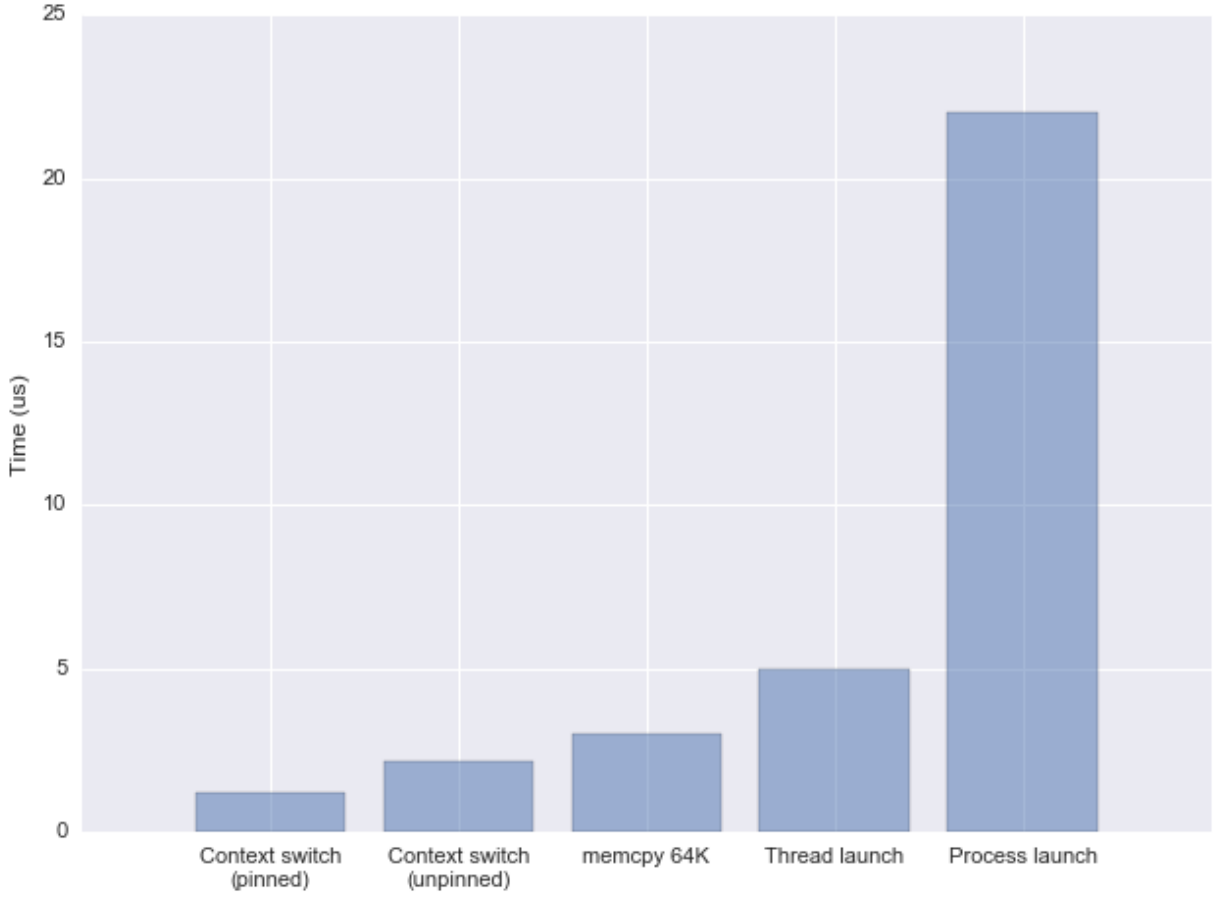
CPU structure and HyperThreading



https://arstechnica.com/gadgets/2009/09/intel-launches-all-new-pc-architecture-with-core-i5i7-cpus/

Intel i7/i5 - 2009

A note about times



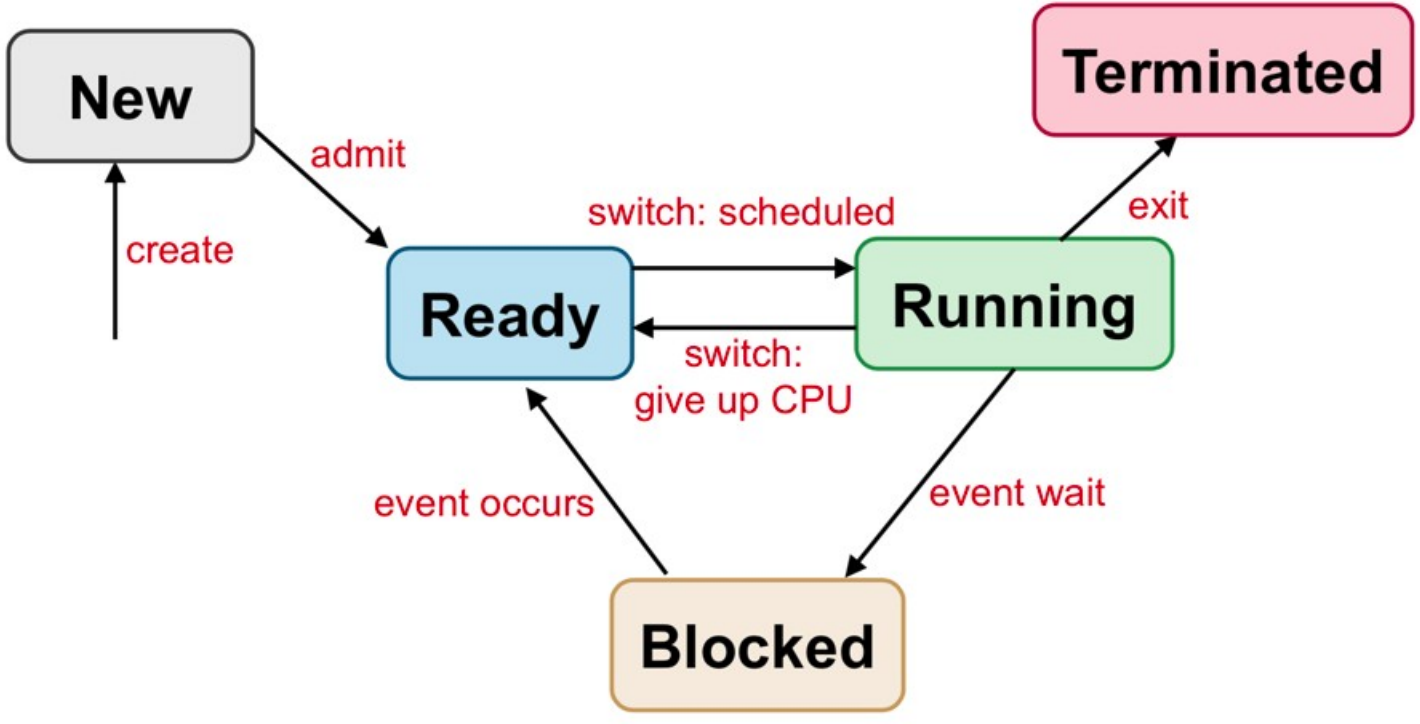
https://eli.thegreenplace.net/2018/measuring-context-switching-and-memory-overheads-for-linux-threads/

State Diagram

State (unix)

- 1. Running (actually using the CPU at that instant).
  2. Ready (runnable; temporarily stopped to let another process run).
  3. Blocked (unable to run until some external event happens).

A note about times



https://cs.stackexchange.com/questions/104463/appreciation-of-the-5-state-process-model

The need for a Process

Process as a structure

Process as a structure

- Thread(s)
- Page table
- Stack (for each thread) + heap
- Code
- Signals settings & handler
- File descriptiors
- Meta: PID,PPID,Permitions

Application types

- Interactive (foreground)
  vs
  Non – Interactive (background)
- Real – Time
  vs
  Batch-processing