University of BRISTOL

Computer Systems B

COMS20012

Introduction to Operating Systems and Security

University of BRISTOL

Threads

bristol.ac.uk

What is a thread?

… an abstraction for the CPU

… a sequence of instructions to execute

- A “normal” **sequential program** consist of a single thread
- Threads are a way to express **concurrency**
- In threaded **concurrent programs** there are multiple threads executing at the same time
  - Threads may perform the same task
  - Threads may perform different task
  - See Firefox example

Why threads?

- Problems the OS is trying to address
  - **Utilization**: there is only one CPU and it is much faster than anything else
    - Programs will wait on resources
    - We saw previously busy waiting is not great
    - We need a mechanism to organize CPU utilization
  - **Priority**: allocate CPU time based on "importance" of a task
  - **Modularization**: separate task responsibility (e.g., check Firefox example)
  - **Responsiveness:** application can use thread to "hide" delay

Batch scheduling

- How task were handled in the early days (see Week 5 Video 1)
- Run job sequentially until completion
- Slow devices idle the CPU (Utilization problem)
- Important tasks may get stuck behind (Prioritization problem)

Creating the illusion of concurrency

- Assume a single core system for now
- OS can create illusion of concurrency by quickly switching between tasks
- **Hardware timer interrupt** at regular interval (Week 5 Video 3)
- On interrupt switch to another thread to execute
- If a thread need to wait for a resource **yield** to another thread

Context switch

- OS **abstractions must hide complexity**
- Timer interrupts means a thread could be stopped at any time
- The OS must make it appear as if nothing happens

A thread is:

- Registers
- Stack
- The rest is shared within the process (Week 6 Video 1)

Context switch

- Context switch is not free
  - Need to save and restore threads states (registers)
- Cost incurred
  - Entering the kernel (e.g., on timer interrupt)
  - Saving current thread states
  - Restoring new thread states
  - + some extra steps if also switching process
- Rate need to be selected to allow good parallel progress…
- … but not too high or the switch cost would dominate

Checkpoint

- **Abstraction:** thread (this and previous videos)
- **Mechanism:** context switching (next videos)
- **Policy:** scheduling (next week)

University of BRISTOL

Thank you

bristol.ac.uk