University of BRISTOL

Computer Systems B

COMS20012

Introduction to Operating Systems and Security

bristol.ac.uk

University of BRISTOL

Threads and zombies

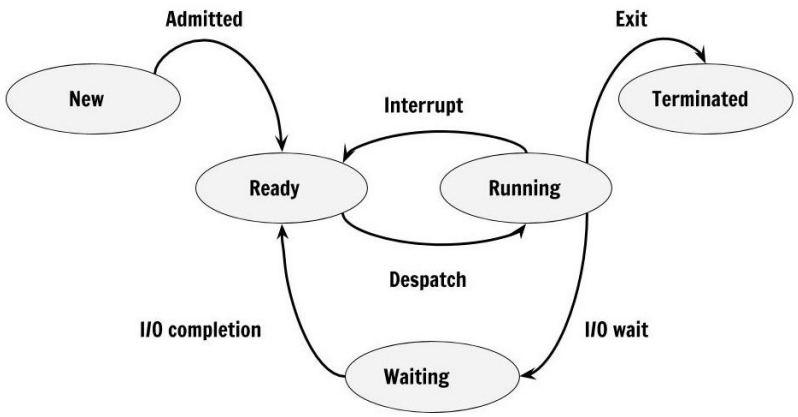
bristol.ac.uk

Checkpoint

- thread_yield
- Core logic where the scheduler switch between threads
- Implemented in *kern/thread/thread.c* (line 500)

bristol.ac.uk

Thread states



bristol.ac.uk

Thread states

- New
  - Newly created thread
  - This is the state while the thread is being initialized
  - Does not exists as such in OS161
- Ready
  - The thread has been initialized
  - It is waiting to be run
- Run
  - The thread is currently being executed
- Waiting
  - The thread is waiting on some IO completion
  - **Sleeping** in OS161
- Terminated
  - The thread has finished executed and is waiting to be cleaned up
  - This are the **zombies** in OS161

bristol.ac.uk

Note

- Some schedulers may implement more states
- This is the basic and minimum logic you need

bristol.ac.uk

University of BRISTOL

- Pause the video
- Open *kern/thread/thread.c*
- Resume the video

bristol.ac.uk

thread_yield

- Line 500: function start
- Line 616-639 switch to a new state
  - Go to sleep (i.e., wait on IO)
  - Go back to ready (i.e., let another thread execute due to the scheduler)
  - Become a zombie (i.e., wait for the data to be cleaned up)
- On line 620 calls thread_make_runnable
  - Implemented line 465
  - Interesting bit is line 482
    - Add thread to the tail of a per cpu queue
- Line 660-669
  - Pick the head of the per cpu queue to be the next one to execute
- Line 678-682
  - Switch between the threads (see last week videos)
- Line 731-end
  - Clean up things and get things ready

bristol.ac.uk

Checkpoint

- This is how you implement a **basic round robin scheduler**
- In the next videos we will see more complex algorithms
- The more adventurous can try to implement more complex algorithms for OS/161 (this is **entirely optional**)
- We have seen all the basic building blocks to do so at this point

bristol.ac.uk

University of BRISTOL

Thank you

bristol.ac.uk