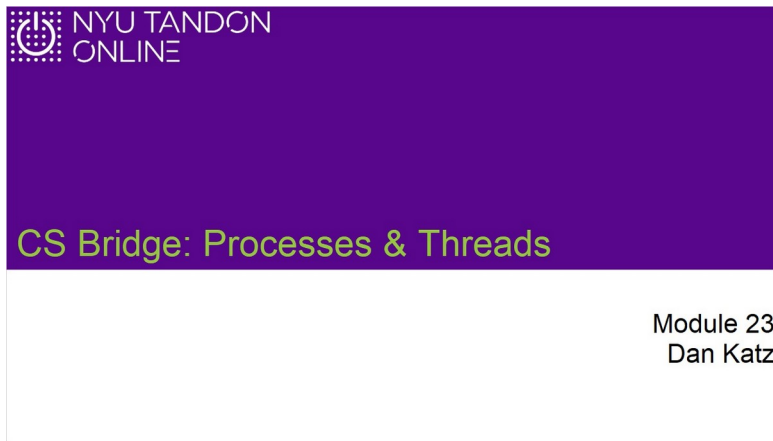


# CS Bridge Module 23 Processes and Threads

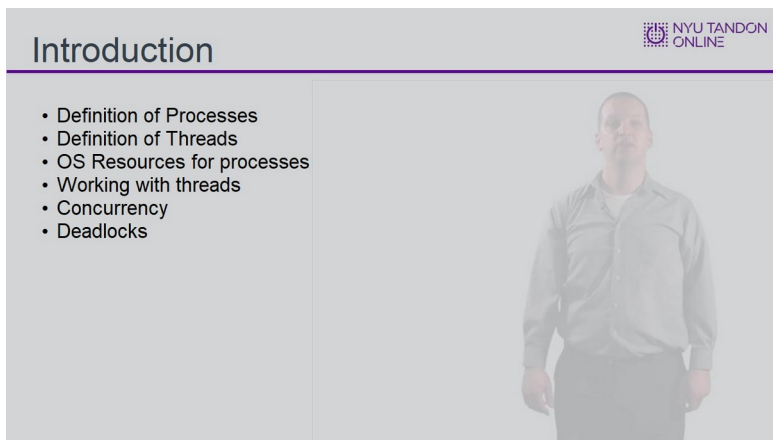
## 1. Processes and Threads

### 1.1 CS Bridge: Processes & Threads



Notes:

### 1.2 Introduction

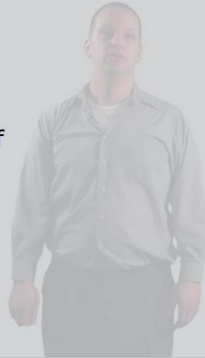


Notes:

### 1.3 What is a Process?

#### What is a Process?

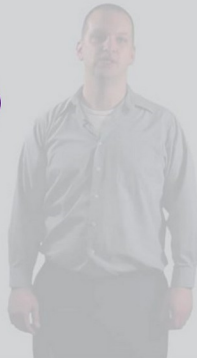
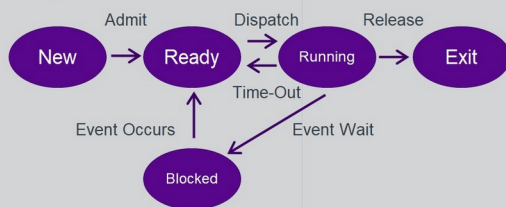
- Code, data and Context
- A running program in a system state
- Code is all in sequential memory
- Process is created by OS to keep track of
  - State of the running program
  - Resources assigned to the running program



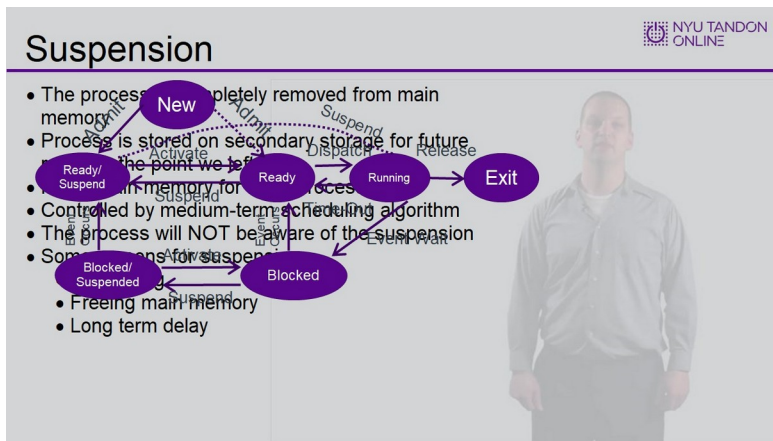
### 1.4 5 state process model

#### 5 State Process Model

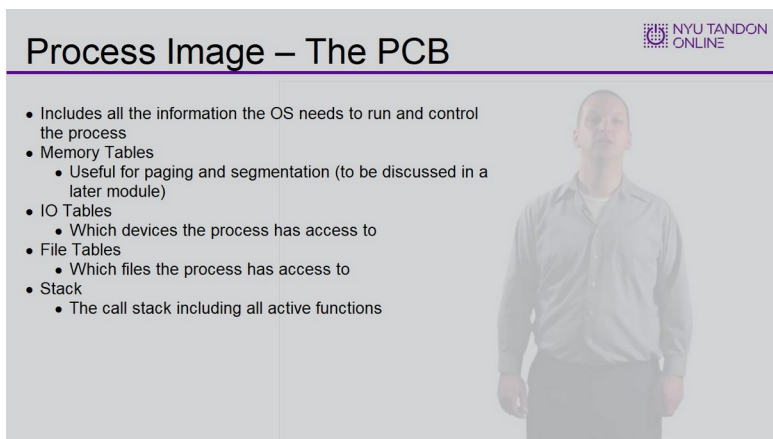
- On condition that the process will spend a significant amount of time in



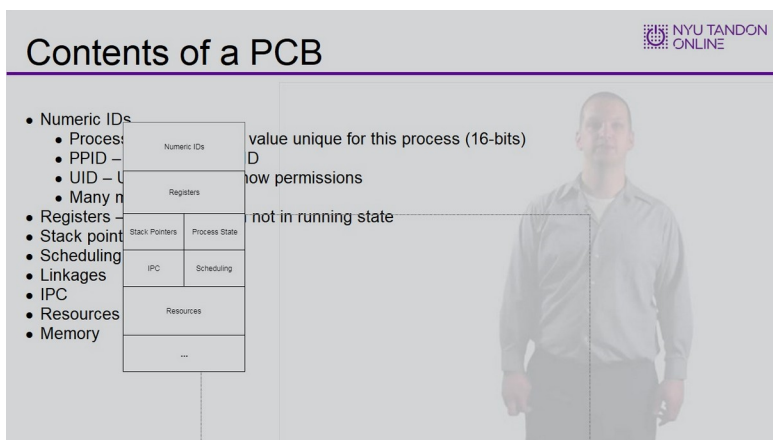
## 1.5 Suspension



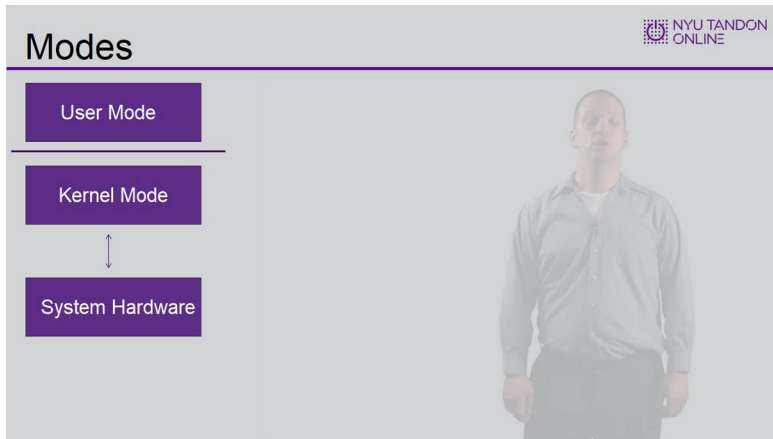
## 1.6 Process Image - The PCB



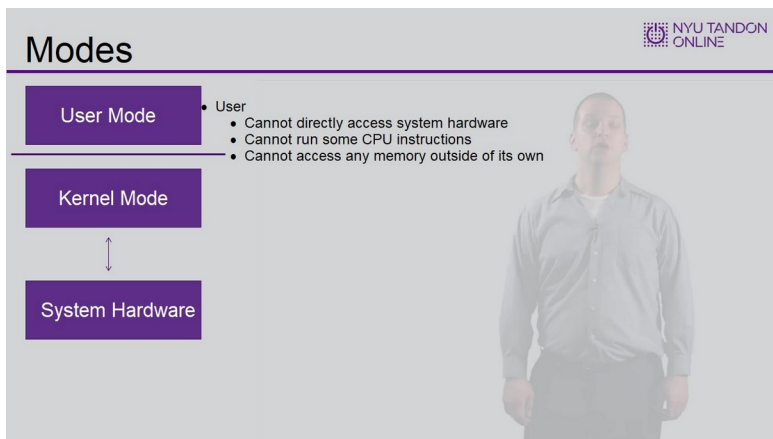
## 1.7 Contents of a PCB



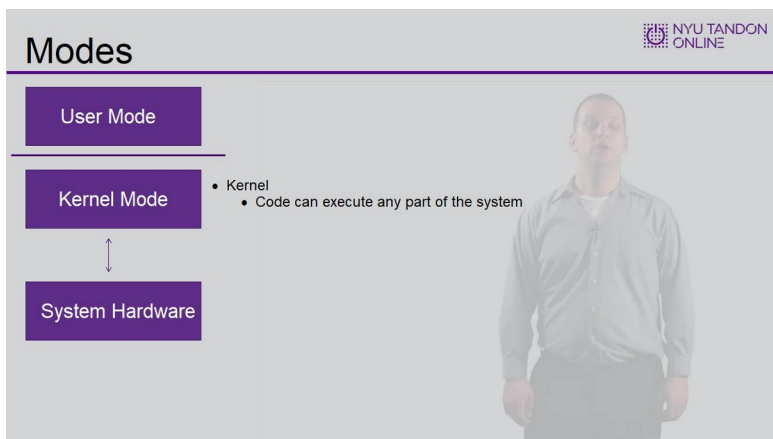
## 1.8 Modes



### User Mode (Slide Layer)



### Kernel Mode (Slide Layer)



## System Hardware (Slide Layer)

### Modes

- User Mode
- Kernel Mode
- System Hardware

- In most processors, the Program Status Word (PSW) tells the system which mode it's in
- Switching
  - Easy to go from Kernel to User
  - User to Kernel happens automatically upon certain events

## 1.9 Process Switching

### Process Switching

- When does a switch occur?
- Interrupt
- Trap
- Blocking System Call
- Process Switching


## Interrupt (Slide Layer)

### Process Switching

- When does a switch occur?
- Interrupt
  - A hardware signal indicating that the hardware needs servicing
- Trap
- Blocking System Call
- Process Switching

## Process (Slide Layer)

### Process Switching



- When does a switch occur?


Interrupt

Trap

Blocking System Call


Process Switching

- Process Switching
  - Save the context into the PCB
  - Update accounting
  - Move the PCB to the appropriate queue
  - Choose another process
  - Update memory management
  - Restore its context
- Multiprocessing means these problems become more complex and happen more often.



## Trap (Slide Layer)

### Process Switching



- When does a switch occur?


Interrupt

Trap

Blocking System Call


Process Switching

- A condition which requires OS support



## Blocking System Call (Slide Layer)

### Process Switching



- When does a switch occur?

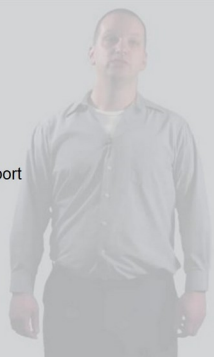
Interrupt

Trap

Blocking System Call


Process Switching

- A request from the process for OS support




## 1.10 Threads

### Threads




- Resource ownership and Execution are two different issues
  - Resource ownership now becomes the only concern of the process
  - Execution – scheduling and running PARTS, yes there will be many, of the process become threads.




## 1.11 What Is Where In The Multithreaded Environment

### What Is Where In The Multithreaded Environment




- Process
  - All of the PCB contents
  - Memory allocation
  - Files
  - Linkages
  - In an OS with threads, does a process exist if it has no threads???
  - Hello World
- Thread
  - Context (processor registers)
  - Stack (incl. Local variables)
  - Access to all of the resources of the thread

**Does a process exist if it has no threads?**




## 1.12 Reasons for multithreading

### Reasons for Multithreading




- Foreground/Background
- Asynchronous processing
- Synchronous processing
- Infrequent tasks
- Speed reading
- Modular program structure




### 1.13 Performance example

#### Performance Example




- File server which does IO for 80% of the task
  - No threads, all asynchronous = 100tps
  - Threads, CPU synchronous with IO = 125tps
  - Caching, CPU bound = 400tps
  - SMP, IO bound = 500tps

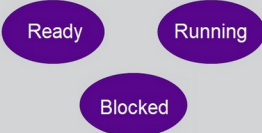


### 1.14 Thread states/Operation

#### Thread States/Operation



- Ready, Running and Blocked
  - threads don't need New or Exit?
  - Suspension is a process level concept



### Untitled Layer 1 (Slide Layer)

#### Thread States/Operation



- Ready, Running and Blocked
  - threads don't need New or Exit?
  - Suspension is a process level concept





## 1.15 What are the downsides

### What Are The Downsides

NYU TANDON ONLINE

- CONCURRENCY!!!!
- Over use of threads leading to confusion

Concurrency (Multiple Queues, One Resource) ->

Parallelism (Multiple Queues, Multiple Resources) ->

Notes:

## 1.16 Implementation of Threads

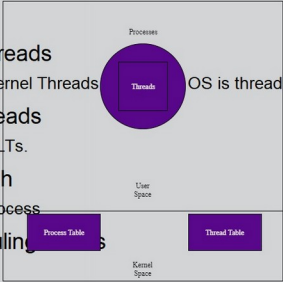
### Implementation of Threads

NYU TANDON ONLINE

- Kernel-Level threads
  - Different from Kernel Threads (OS is threaded)
- User Level Threads
  - Downsides of ULTs.
- Hybrid approach
  - Light Weight Process
- Thread Scheduling


## Untitled Layer 1 (Slide Layer)

### Implementation of Threads




The diagram illustrates the implementation of threads across two spaces: User Space and Kernel Space. In the User Space, a 'Process' box contains a 'Thread' box, with a note '(OS is threaded)'. In the Kernel Space, there are two tables: a 'Process Table' and a 'Thread Table'. Arrows indicate the mapping between threads in user space and their corresponding entries in the kernel tables.

- Kernel-Level threads
  - Different from Kernel Threads
- User Level Threads
  - Downsides of ULTs.
- Hybrid approach
  - Light Weight Process
- Thread Scheduling



NYU TANDON  
ONLINE

## 1.17 End of Module



NYU TANDON  
ONLINE

### End of Module

Exit