Context Switch

Process A
Control Block

Threads
A1
A2
A3

thread state

Core
Hardware Registers

R0
R1
RN
SP

thread state

Process B
Control Block

Threads
B1

A3 Stack

B1 Stack

A3 Stack

CS 111 Lecture Notes: Threads, Processes, and Dispatching
Context Switch

Process A Control Block

Threads

A1
A2
A3

thread state

Core

Hardware Registers

R0
R1
RN
SP

A3 Stack

Saved Registers (all but SP)

B1 Stack

Process B Control Block

Threads

B1

thread state

A2 Stack

Control Block

A3 Stack

Control Block

B1 Stack
Context Switch

Process A
Control Block

Threads
A1
A2
A3

thread state

Core

Hardware Registers

R0
R1
RN
SP

thread state

Process B
Control Block

Threads

B1

A3 Stack

B1 Stack

A2 Stack
Context Switch

Process A
Control Block

Threads
A1
A2
A3

thread state

Core

Hardware Registers

thread state

B1

Process B
Control Block

Threads
B1

A3 Stack

SP

B1 Stack

A3 Stack

R0
R1
RN
SP

Process A

Control Block

Threads
A1
A2
A3

thread state

Core

Hardware Registers

thread state

B1

Process B
Control Block

Threads
B1

A3 Stack

SP

B1 Stack

A3 Stack

R0
R1
RN
SP

Context Switch
Context Switch

- Thread states
- Process A Control Block
  - Threads: A1, A2, A3
  - Stack: A3 Stack
- Process B Control Block
  - Threads: B1
  - Stack: B1 Stack
- Core
  - Hardware Registers: R0, R1, ..., RN
  - Stack: SP
- Control Blocks
  - Process A Control Block
  - Process B Control Block
Linux Fork/Exec Example

```c
int pid = fork();
if (pid == 0) {
    execvp("ls", argv);
} else {
    waitpid(pid, &status, options);
}
```

- **Child process**
- **Parent process**
Windows Process Creation

```c
BOOL CreateProcess(
    LPCTSTR lpApplicationName,
    LPTSTR lpCommandLine,
    LPSECURITY_ATTRIBUTES lpProcessAttributes,
    LPSECURITY_ATTRIBUTES lpThreadAttributes,
    BOOL bInheritHandles,
    DWORD dwCreationFlags,
    LPVOID lpEnvironment,
    LPCTSTR lpCurrentDirectory,
    LPSTARTUPINFO lpStartupInfo,
    LPPROCESS_INFORMATION lpProcessInformation
);

WaitForSingleObject(lpProcessInformation->hProcess, INFINITE);
```