Processes

COSC-361

Stephen Marz

TICKLE COLLEGE OF ENGINEERING

MIN H. KAO DEPARTMENT OF ELECTRICAL ENGINEERING & COMPUTER SCIENCE

1

Process

- A process is a conceptual structure of "work" on a CPU.
- The work performed by a process is a task.
  - A process and task are synonymous in Linux.

2

Process Control Block (PCB)

- Information needed to identify one process from another.
  - Process ID (PID)
  - Copy of the registers
  - Heap memory allocations
  - Page map
  - File allocations (descriptors)
  - Device buffers
  - Scheduling information
    - Runtime
    - Context switches

3

Process Creation

- Created by the OS from a system call
  - Linux: fork(), exec*()
  - Windows: CreateProcess()
- A created process must be loaded into memory (at least what is necessary to run).
  - CPU can only fetch instructions from memory.

4

Process Execution

- Operating System's Duties
  - Restore register's state
  - Reprogram MMU
  - Restore driver contexts
  - Set program counter (instruction pointer) to last known instruction to execute.
  - Execute...

5

Process States

- A process may be running, waiting, put in the penalty box, etc.
- No standard "states"
  - Linux
    - RUNNING
    - INTERRUPTIBLE SLEEP
    - UNINTERRUPTIBLE SLEEP
  - Academic
    - NEW
    - RUNNING
    - READY
    - BLOCKED
    - SUSPENDED
    - TERMINATED

6

Parent / Child Relationships

- Reporting system
- A parent can have many children
- A child can have one parent
  - When child dies, parent is notified (SIGCHLD for SYS-V).
- Child memory management
  - A child contains the parent's memory map until the child writes to it.
  - After the child's first write, the memory is copied, and the child has its own copy.

7

PCB Organization

- PCBs are typically in a static array or dynamic list.
  - Dynamic lists require the OS to implement the heap.
    - This is NOT always expected!
- Linked lists
  - Save Memory
  - Allocation Overhead
- Static array
  - Fixed size at compile time
  - Wastes memory not being actively used

8

PCB Wait vs Ready Queues

- Wait queue
  - Contains pointers that point to all processes waiting on I/O, or are sleeping for a period fo time.
- Ready queue
  - Contains pointers that point to all processes waiting on the CPU.

9

Threads

- Heavyweight vs Lightweight Threads
  - Heavyweight threads are new processes that detach from the parent.
    - Contain their own state, memory, etc.
  - Lightweight threads are tied to their parents.
    - Contain parent's memory, state, etc.
      - Saves on the time and memory used to create and execute a new process.

10

Thread Control Blocks

- Contain only the information different from the thread's parent.
- Threads can turn into processes
  - Linux does this when a thread writes to memory.
    - MMU makes child's memory read-only
    - When child writes, it causes a page fault
    - OS handles this and duplicates memory
    - OS resumes thread as a new process

11

12