The Nanvix Operating System

Process Scheduling

Pedro H. Penna

pedrohenriquepenna@gmail.com

January 18, 2017

Background Process Scheduling

Barebones

- Process compete for processor time
- Process scheduler chooses which process to run

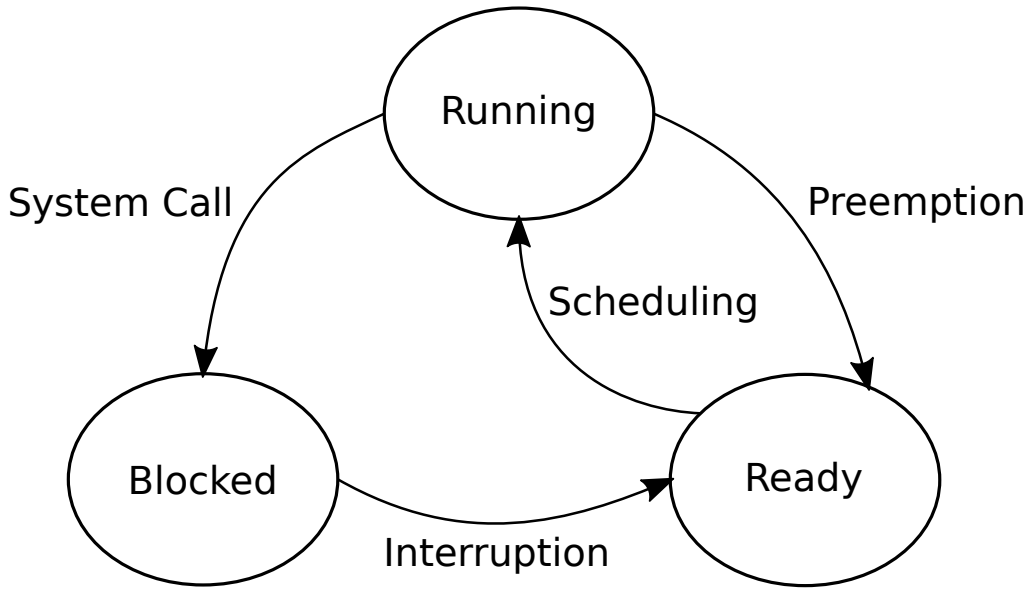

Figure: Process states.

Background on Process Scheduling

Rules of Thumb

- All systems
  - Fairness
  - Policy enforcement
  - Load balancing

Background on Process Scheduling

Rules of Thumb

- All systems
  - Fairness
  - Policy enforcement
  - Load balancing
- Batch systems
  - Throughput
  - Turnaround time

Background on Process Scheduling

Rules of Thumb

- All systems
  - Fairness
  - Policy enforcement
  - Load balancing
- Batch systems
  - Throughput
  - Turnaround time
- Interactive systems
  - Response time
  - Proportionality

Background on Process Scheduling

Classical Algorithms

- Batch systems
  - First-Come First-Served
  - Shortest-Job First
  - Shortest Remaining Time Next

Background on Process Scheduling

Classical Algorithms

- Batch systems
  - First-Come First-Served
  - Shortest-Job First
  - Shortest Remaining Time Next
- Interactive systems
  - Round-Robin Scheduling
  - Priority Scheduling
  - Multiple Queues Scheduling
  - Lottery Scheduling
  - Fair-Share Scheduling

Process Scheduling in Nanvix

Process States

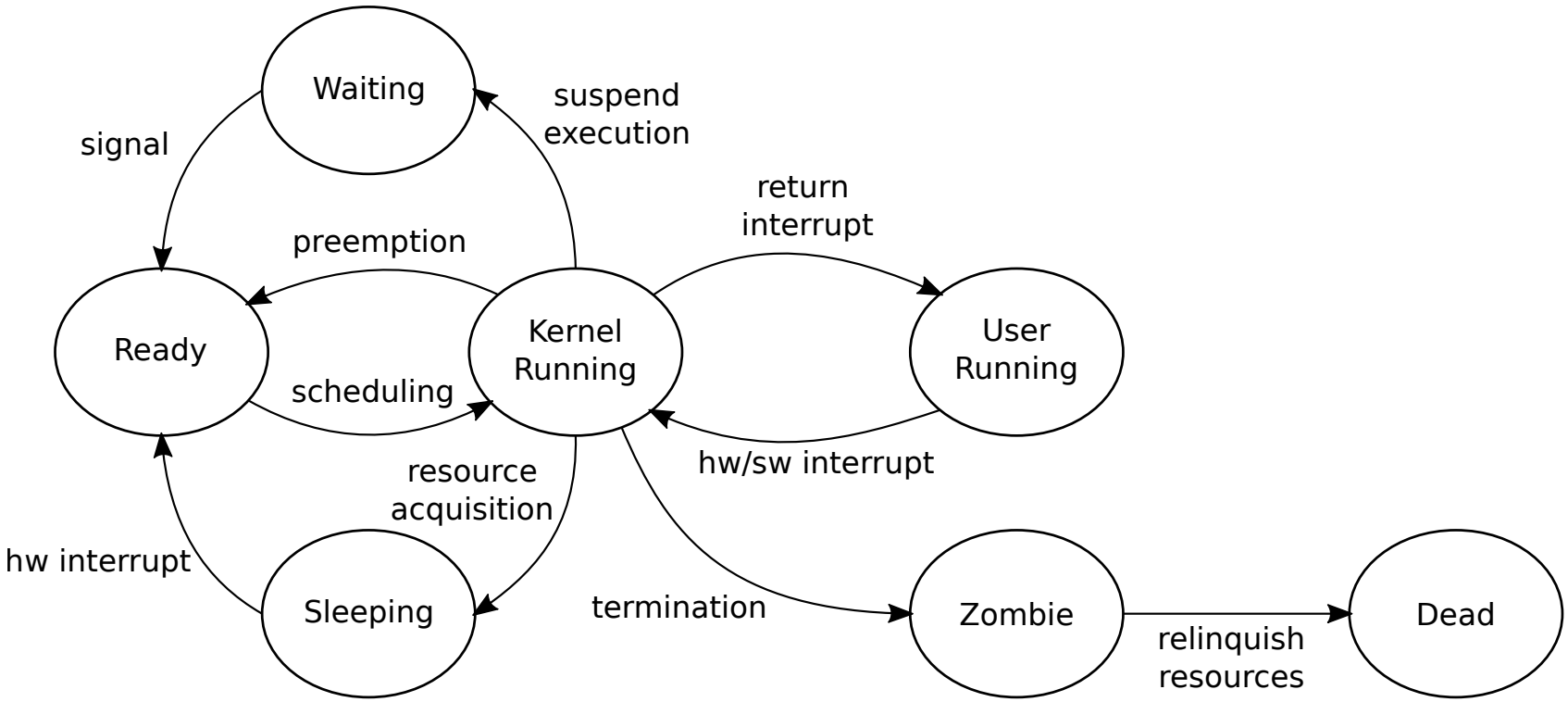

Figure: States of a process in Nanvix.

Process Scheduling in Nanvix

Current Implementation

- Process table entry – `struct process`
  - `include/nanvix/pm.h`

Process Scheduling in Nanvix

Current Implementation

- Process table entry – `struct process`
  - `include/nanvix/pm.h`
- Process scheduler – `yield()`
  - `src/kernel/pm/sched.c`

Process Scheduling in Nanvix

Current Implementation

- Process table entry – `struct process`
  - `include/nanvix/pm.h`
- Process scheduler – `yield()`
  - `src/kernel/pm/sched.c`
- Round-robin scheduling
  - Schedule ready process that is waiting longer
  - Give each process a same processor quantum

Process Scheduling in Nanvix

Current Implementation

- Process table entry – `struct process`
  - `include/nanvix/pm.h`
- Process scheduler – `yield()`
  - `src/kernel/pm/sched.c`
- Round-robin scheduling
  - Schedule ready process that is waiting longer
  - Give each process a same processor quantum
- Support for priority scheduling (unused)
  - Static priority (`priority`)
  - Dynamic priority (`counter`)
  - User priority (`nice`)