Operating systems

Processes
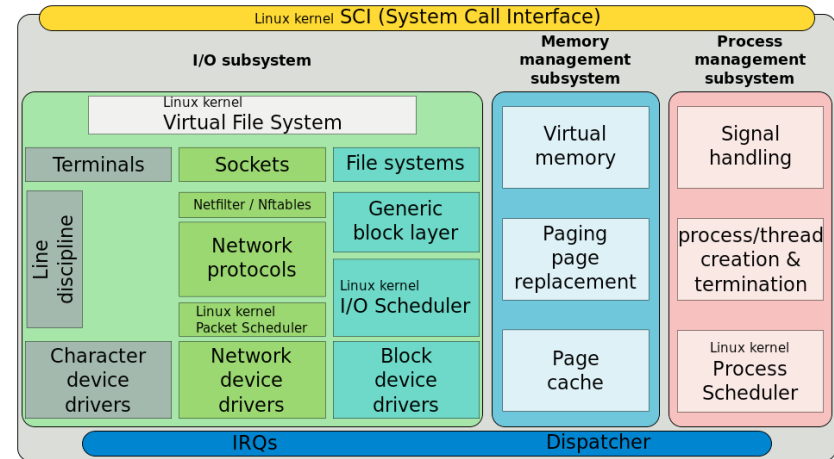
What is a process?

a **running** program

a process is an object

state

← operation

← operation

← operation
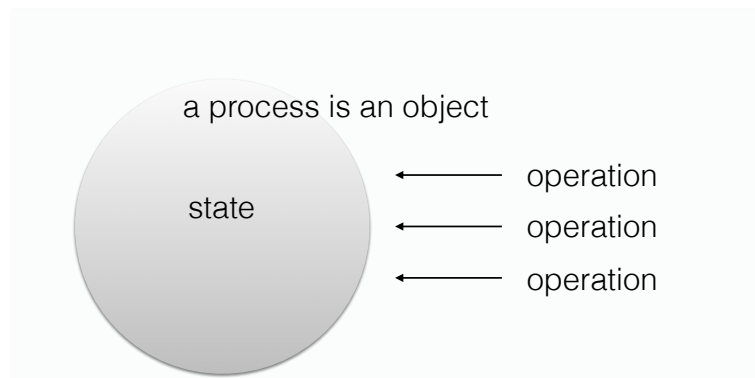
---

parent

resources

reference

references

stack frame   registers

process state

Slide 1: process creation

resources

process creation

Slide 2: process creation tree

process creation tree
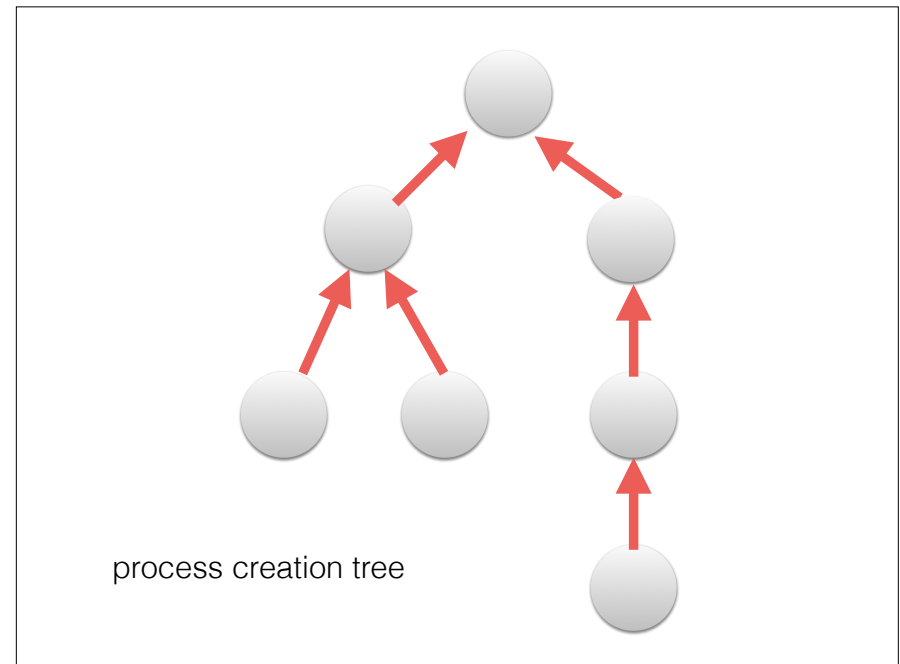
Slide 3: process

process

fork()/create()

wait()

exec()

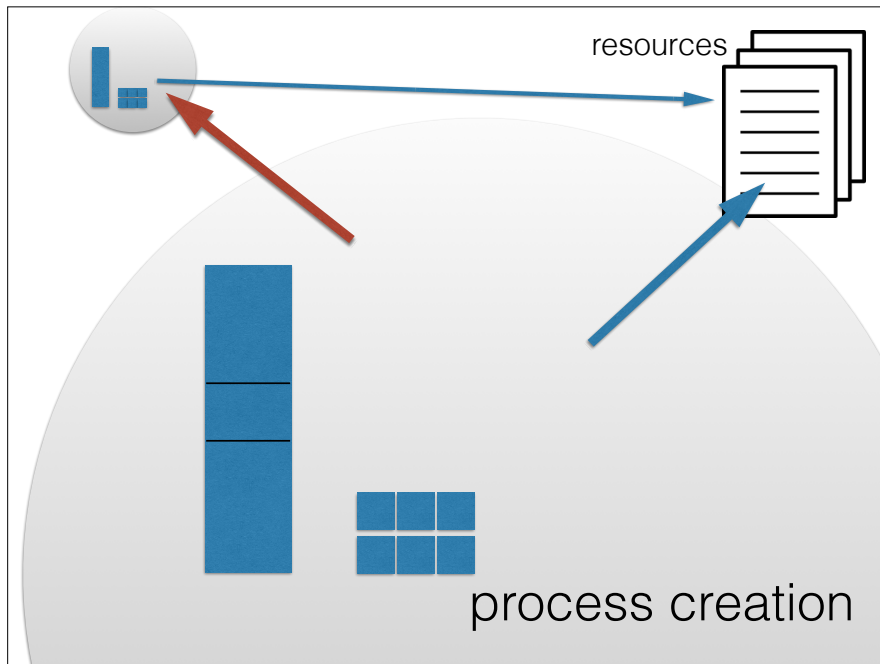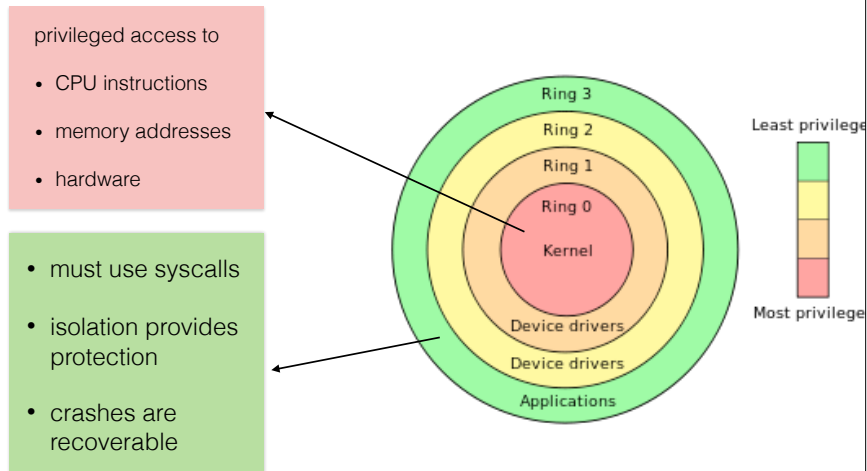## Slide 4: Processes and Java
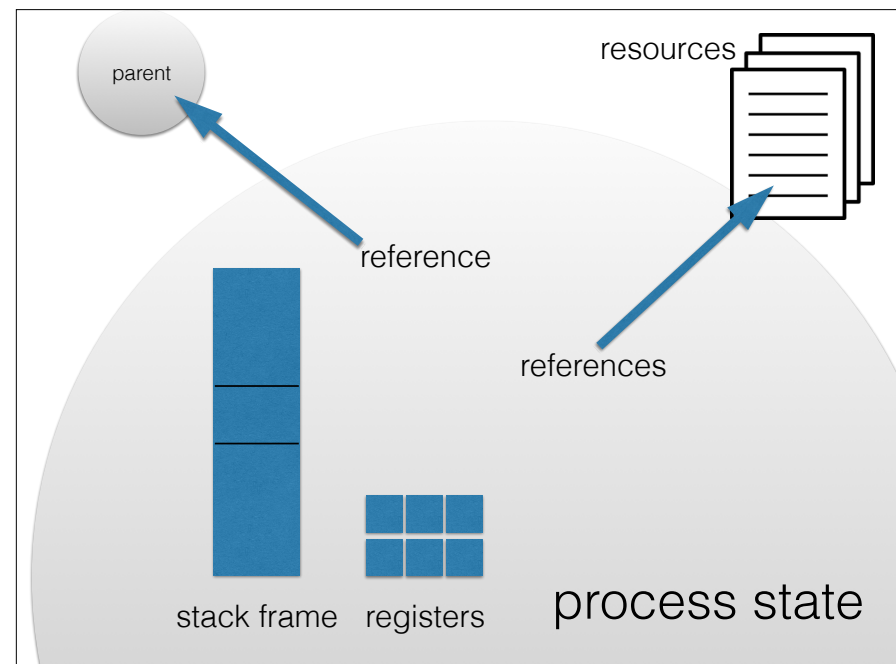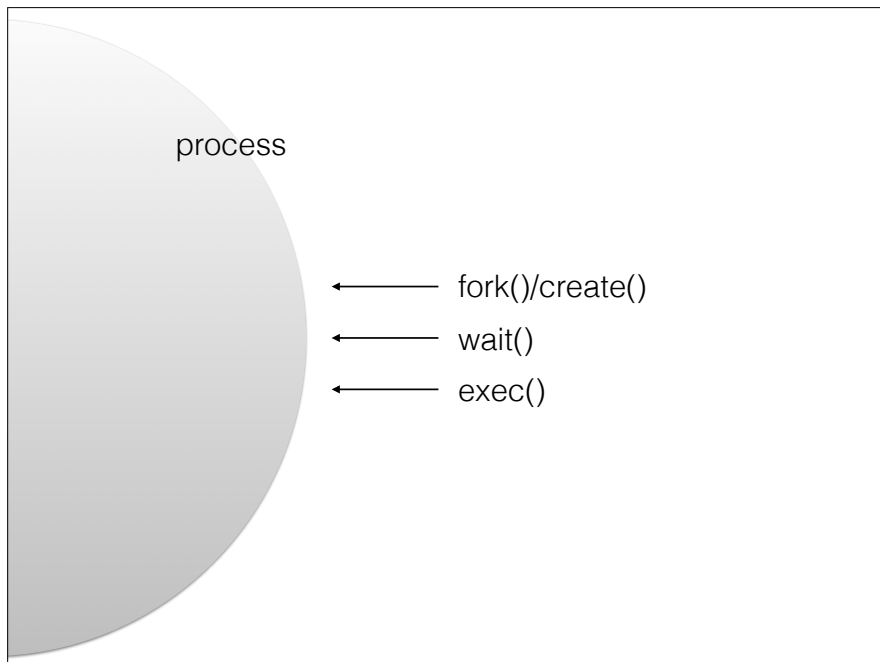
Processes and Java

- one process per JVM

- similar ideas in Java Threads API which run "Runnable" objects (see https://docs.oracle.com/javase/8/docs/api/java/lang/Thread.html)

- BUT significant differences between processes and threads (later)

User vs kernel mode

privileged access to
- CPU instructions
- memory addresses
- hardware

- must use syscalls
- isolation provides protection
- crashes are recoverable

Ring 3
Ring 2
Ring 1
Ring 0
Kernel
Device drivers
Device drivers
Applications

Least privilege

Most privilege

Limited direct execution

- What?
- Why?
- How?

Traps

hardware traps (interrupts): generated by hardware in need of attention

- clock chip interrupts every 100 msec
- disk block ready to be retrieved
- more difficult to think about as may happen concurrenlty

trap: caused by current running process

- system call/software interrupt
- exceptions: division by 0, illegal memory access

Trap handling

normal processor execution

- read instruction
- advance program counter
- execute instruction
- repeat

upon trap, trap handler

- saves register context on stack
- switch from user to kernel mode

System call trap gates

Application Program

instr ; instr ; instr ; trap ; instr ; instr ;

user mode
supervisor mode

PS/PC
PS/PC
PS/PC
PS/PC

TRAP vector table

return to user mode

1st level trap handler

2nd level handler (system service implmementation)

system call dispatch
table