The Kernel Abstraction

Slides adopted from CSE 451 class at UW, CS162 class at Berkeley and CSE 421/521 class at UB

อดีต

- Single task system

ปัจจุบัน

- Multitasking system

Activity #1

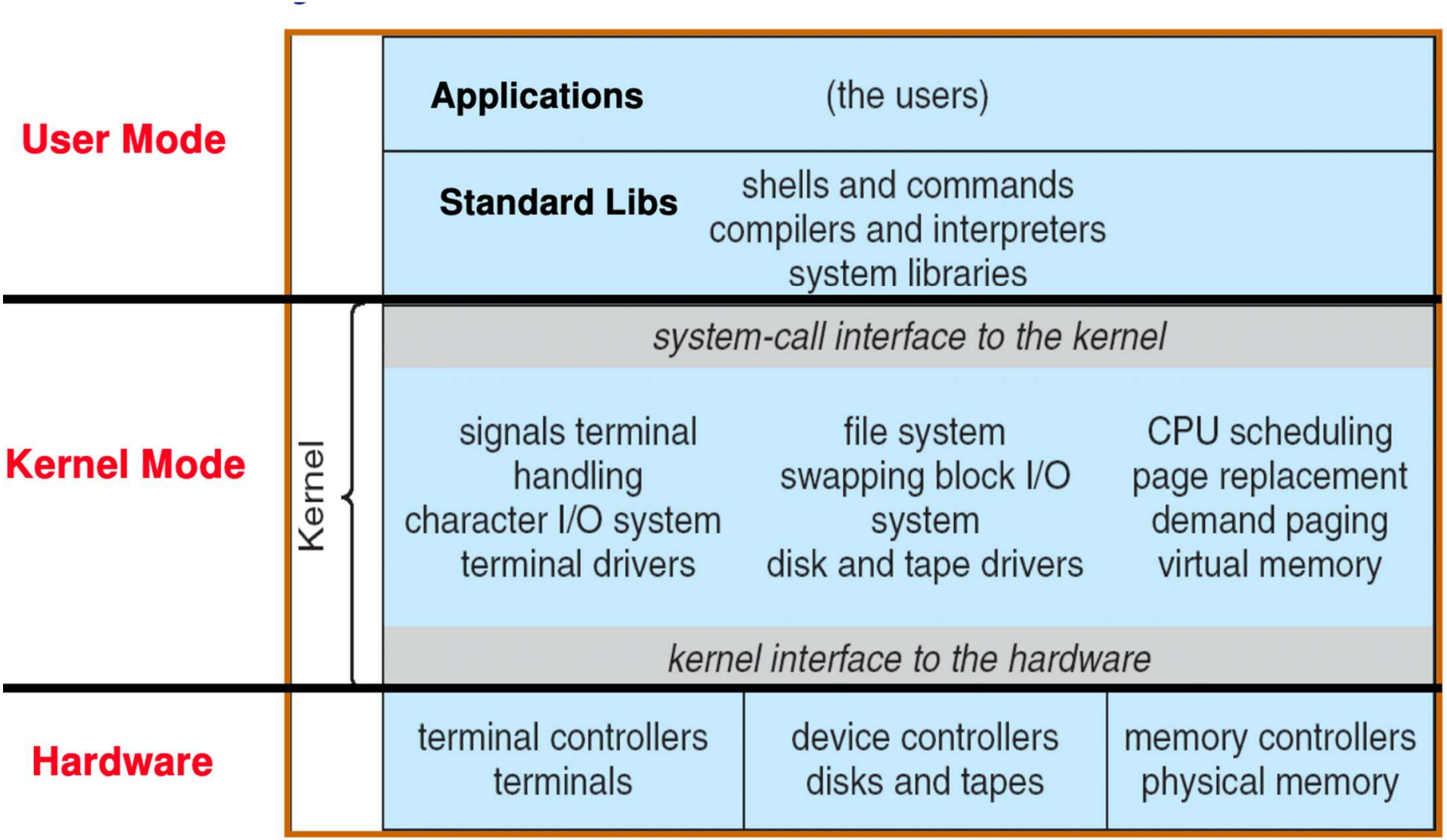
- การเปลี่ยนแปลงจากระบบแบบ single task ไปเป็นระบบแบบ Multitask ... สิ่งที่จะต้องมีการปรับแต่งหรือเพิ่มเติมเข้ามาใน OS ได้แก่...

What does an OS do...

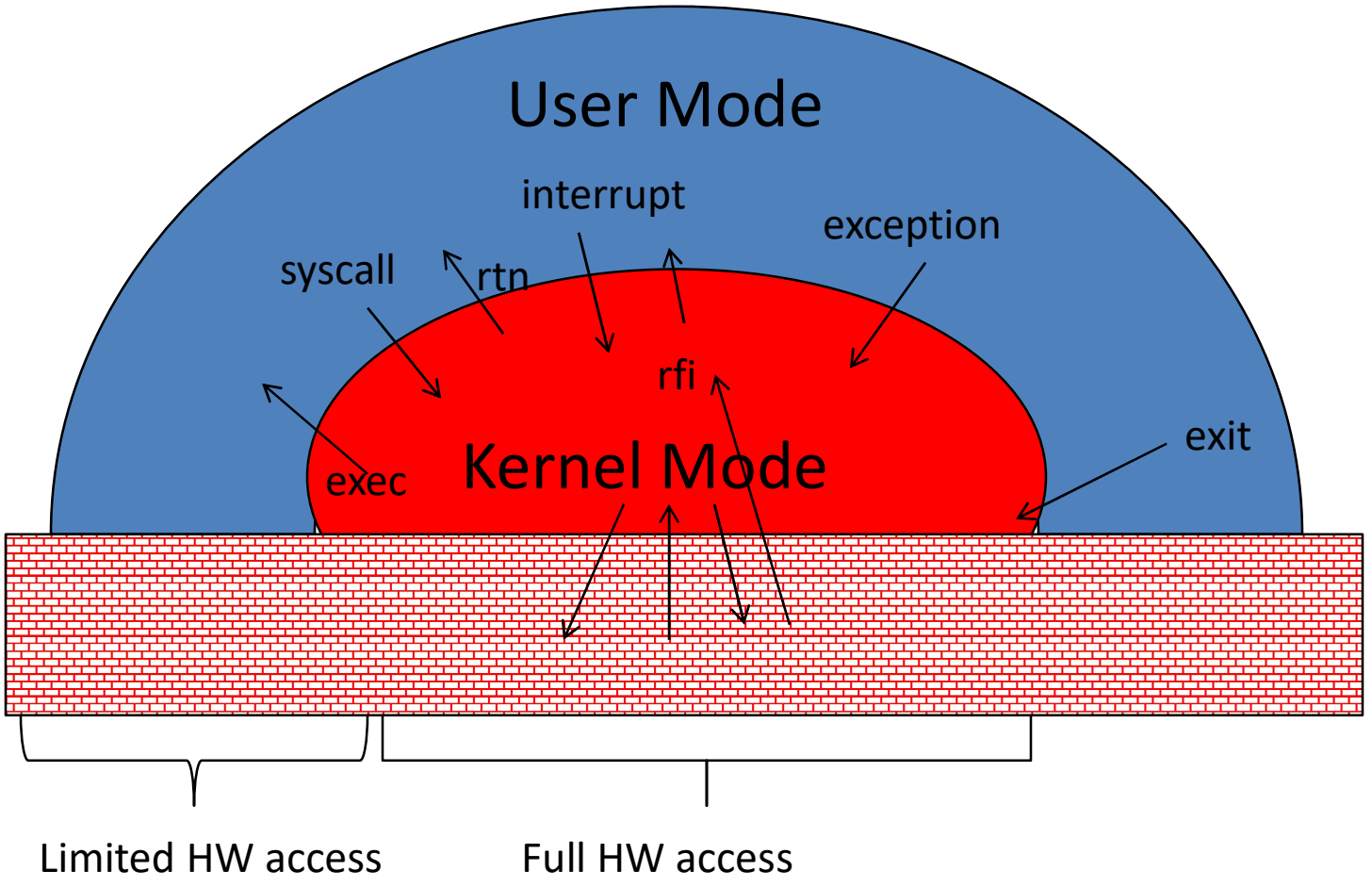
- Hiding Complexity

- Kernel is the part of the OS that running all the time on the computer

UNIX Structure



User/Kernel (Privileged) Mode



One of the major goals of OS is...

- Protecting Process and the Kernel
  - Running multiple programs
    - Keep them from interfering with the OS - kernel
    - Keep them from interfering with each other

Activity #2: Protection: WHY?

เวลา 10 นาที

การ protect Process และ Kernel ทำให้เกิด impact อะไรกับระบบบ้าง และยังต้อง protect อะไรอีกบ้าง เพื่ออะไร

Protection: How? (HW/SW)

เวลา 10 นาที

Hardware Support: Dual-Mode Operation

- Kernel mode
  - Execution with the full privileges of the hardware
  - Read/write to any memory, access any I/O device, read/write any disk sector, send/read any packet
- User mode
  - Limited privileges
  - Only those granted by the operating system kernel
- On the x86, mode stored in EFLAGS register
- On the MIPS, mode in the status register

Hardware Support: Dual-Mode Operation

- Privileged instructions
  - Available to kernel
  - Not available to user code
- Limits on memory accesses
  - To prevent user code from overwriting the kernel
- Timer
  - To regain control from a user program in a loop
- Safe way to switch from user mode to kernel mode, and vice versa

Privileged instructions

- Examples?

- What should happen if a user program attempts to execute a privileged instruction?

User Mode

- Application program
  - Running in process

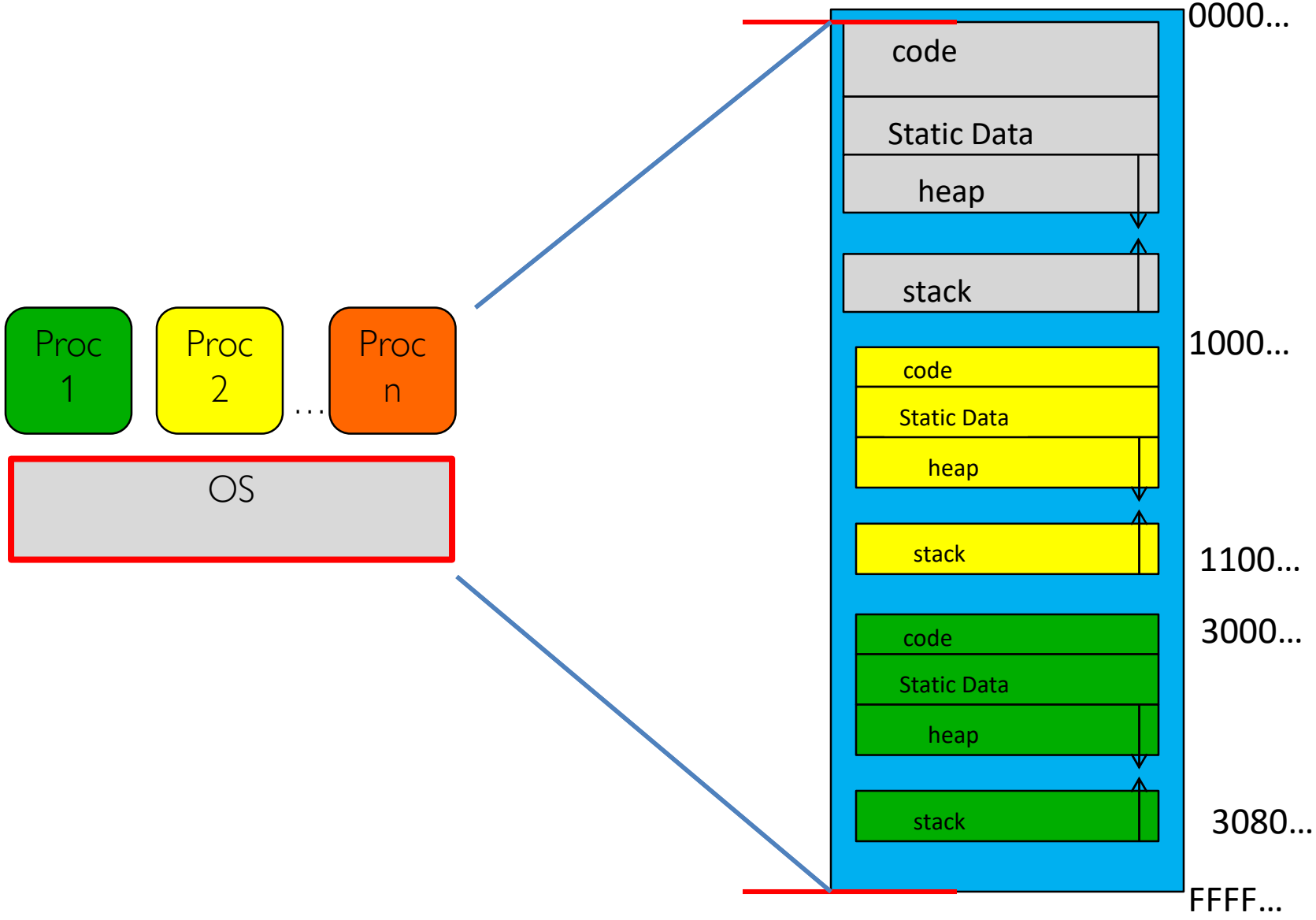
Virtual Machine:VM

- Software emulation of an abstract machine
  - Give programs illusion they own the machine
  - Make it look like HW has feature you want
- 2 types of VM
  - Process VM
    - Supports the execution of a single program (one of the basic function of the OS)
  - System VM
    - Supports the execution of an entire OS and its applications

Process VMs

- GOAL:
  - Provide an isolation to a program
  - Portability (Program)

Kernel mode & User mode

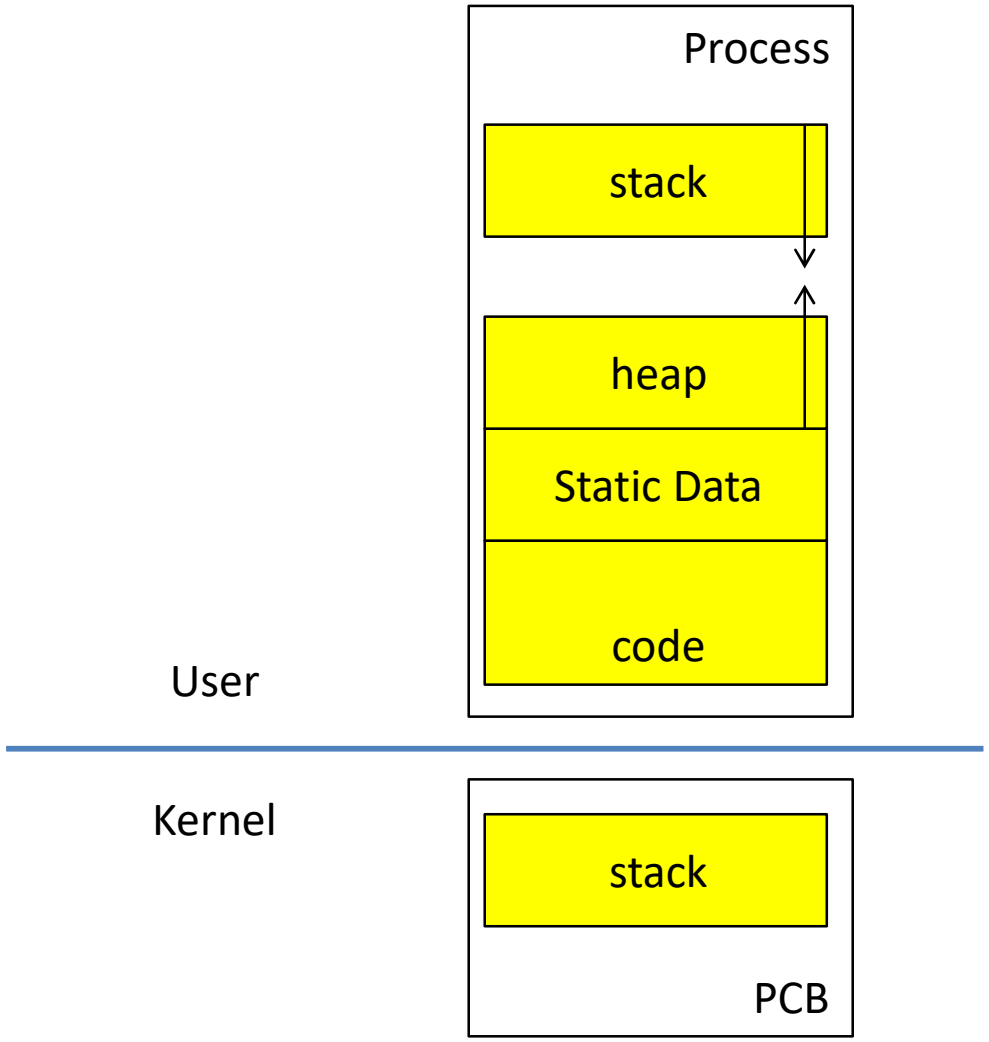


Process Abstraction

- Process: an *instance* of a program, running with limited rights
  - Thread: a sequence of instructions within a process
    - Potentially many threads per process (for now 1:1)
  - Address space: set of rights of a process
    - Memory that the process can access
    - Other permissions the process has (e.g., which system calls it can make, what files it can access)

Process

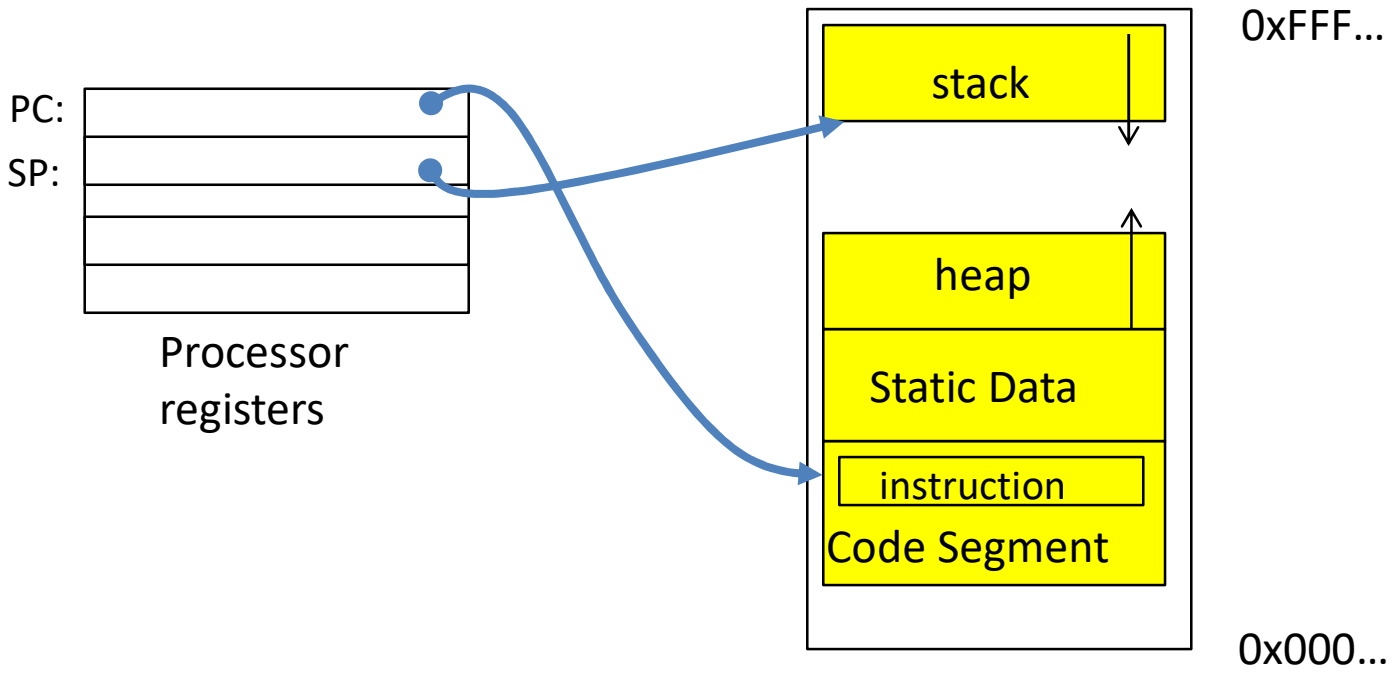
- 2 parts
  - PCB in kernel
  - Others in user



Process Control Block: PCB

- Kernel represents each process as a process control block (PCB)
  - Status (running, ready, blocked, ...)
  - Registers, SP, ... (when not running)
  - Process ID (PID), User, Executable, Priority, ...
  - Execution time, ...
  - Memory space, translation tables, ...
- Kernel Scheduler maintains a data structure containing the PCBs
- Scheduling algorithm selects the next one to run

Address Space: In a Picture



Break