

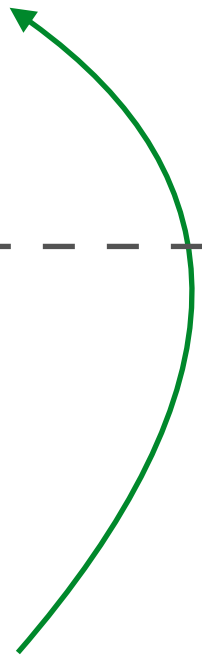
Definition of the process and system calls



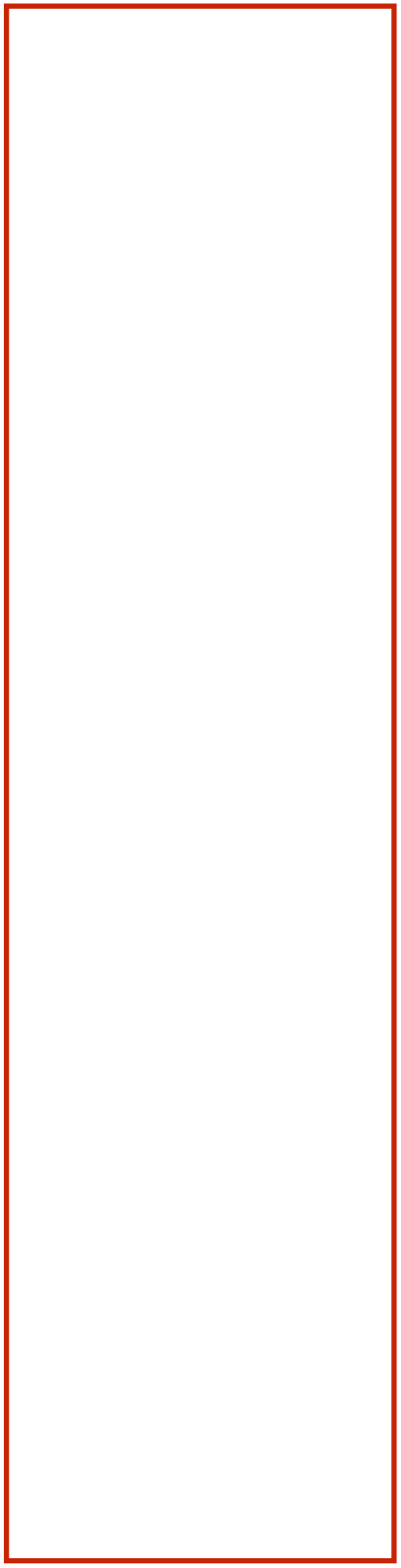
principle 3: process can access
resources through kernel system-calls

user mode

kernel mode



principle 2: the kernel has privileged access to the entire memory (kernel mode)



prog A

stack A

heap A



prog B

stack B

heap B



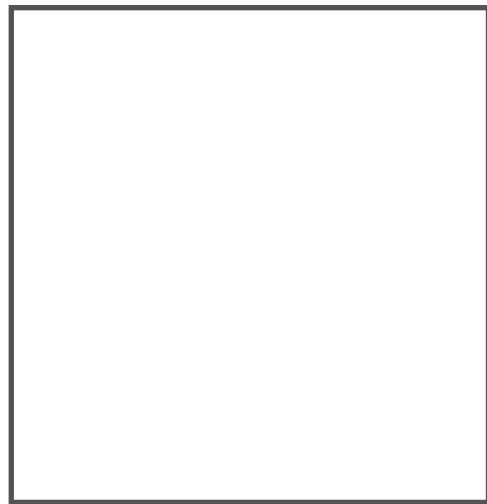
Kernel

kernel stack

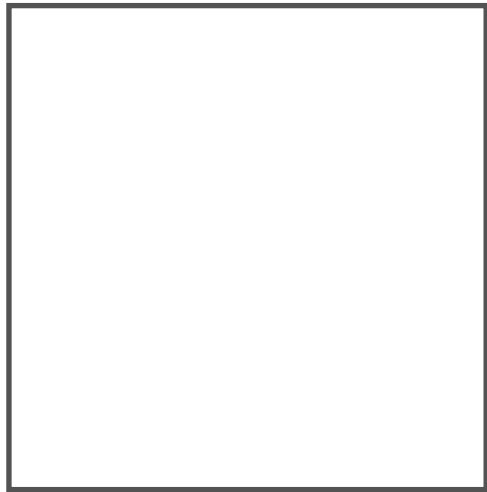
kernel heap



process
B

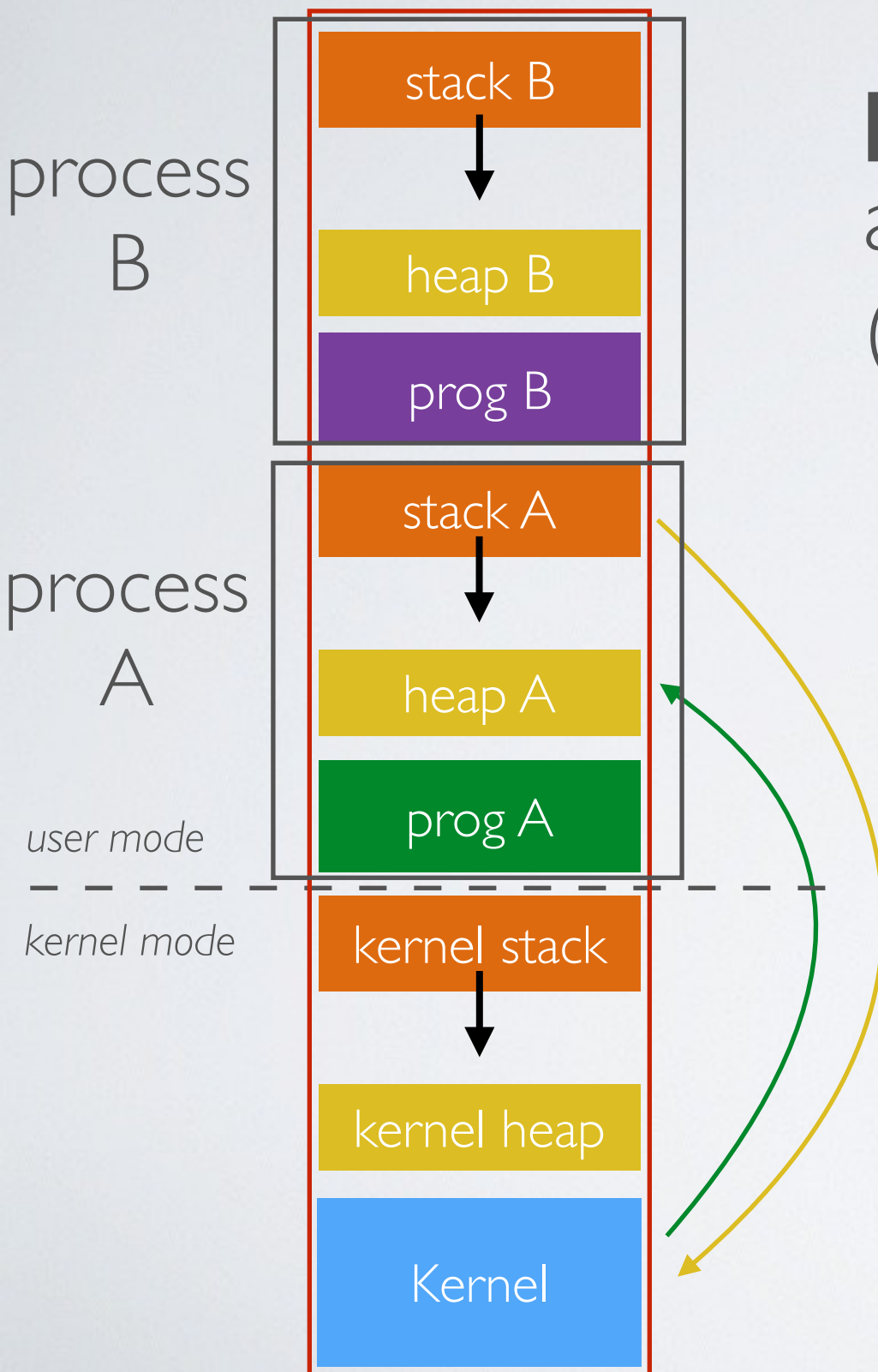


process
A



principle I: user programs are run
as processes isolated from each other
(user mode)

Definition of the process and system calls



principle 1: user programs are run as processes isolated from each other (user mode)

principle 2: the kernel has privileged access to the entire memory (kernel mode)

principle 3: process can access resources through kernel system-calls

How can we isolate processes and kernel memory spaces?

➡ The Virtual Memory (coming soon)

In a nutshell

- User programs do not directly access the memory but the virtual memory (that is somehow mapped onto the real memory)
- The kernel manages the virtual memory for all processes