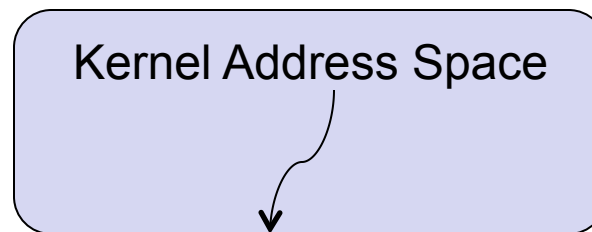
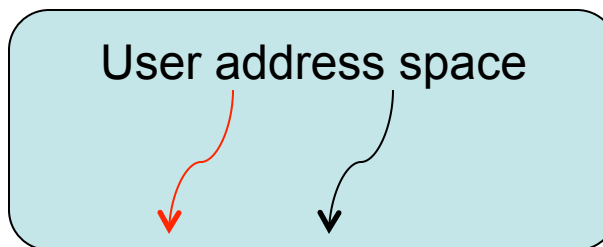


Group Exercises

- Learning Objectives:
 - Be able to follow threads of execution across domain crossings
 - Have a clear mental model of the critical data structures and state you'll need to maintain in the operating system in order to implement user-level processes.

Process Cartoons

- If we ask you to draw “process cartoons,” we mean diagrams like this one – you may have to draw a sequence of them or come up with a suitable way to represent animation. You should also include a bit more detail such as kernel stacks.



Exercise 1: Warmup

- Draw a cartoon of a user process making a system call (e.g., getpid) that the OS can handle immediately, returning to the invoking process.
 - Does this involve a thread switch?
 - Does this involve a domain crossing?
 - Does this involve more than one domain crossing?
- After you have completed your drawing, think about what your drawing implies for assignment 2. Does it suggest any particular data structures or standard functions you'll need?

Exercise 2: A bit trickier

- This time, draw a cartoon of a process making a system call that is going to block (e.g., read), causing the kernel to run some other thread.
- Again, after you've completed your drawing, discuss any implications this sequence of events has on the design of assignment 2.

Exercise 3: The Biggie

- Draw a cartoon of a fork system call.
 - Think carefully about what it means to create a new process. What structures do you have to conjure up? What data structures do you need to allocate? Where should those data structures live?
- And, once again, after you've got a diagram or sequence of diagrams, think about the implications for your design and implementation in assignment 2.