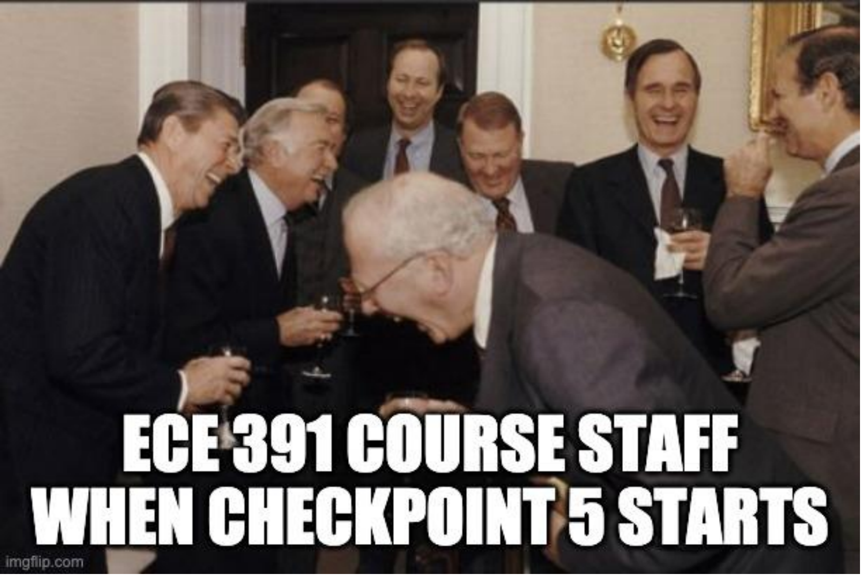
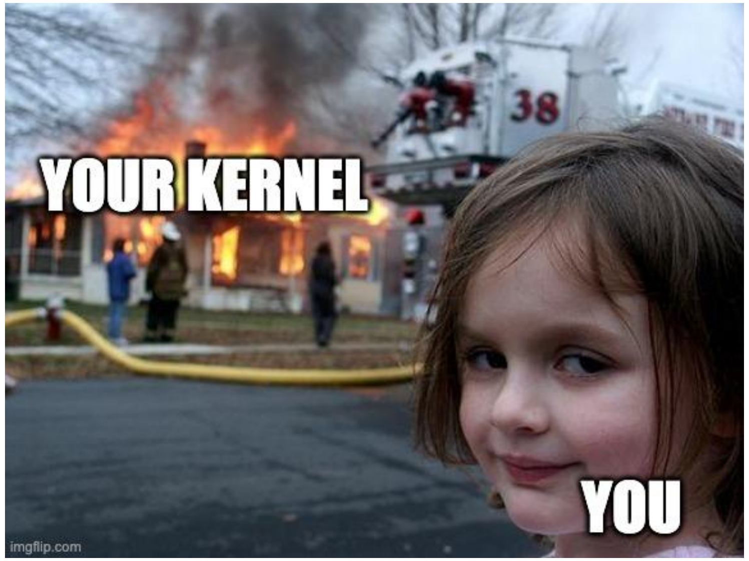
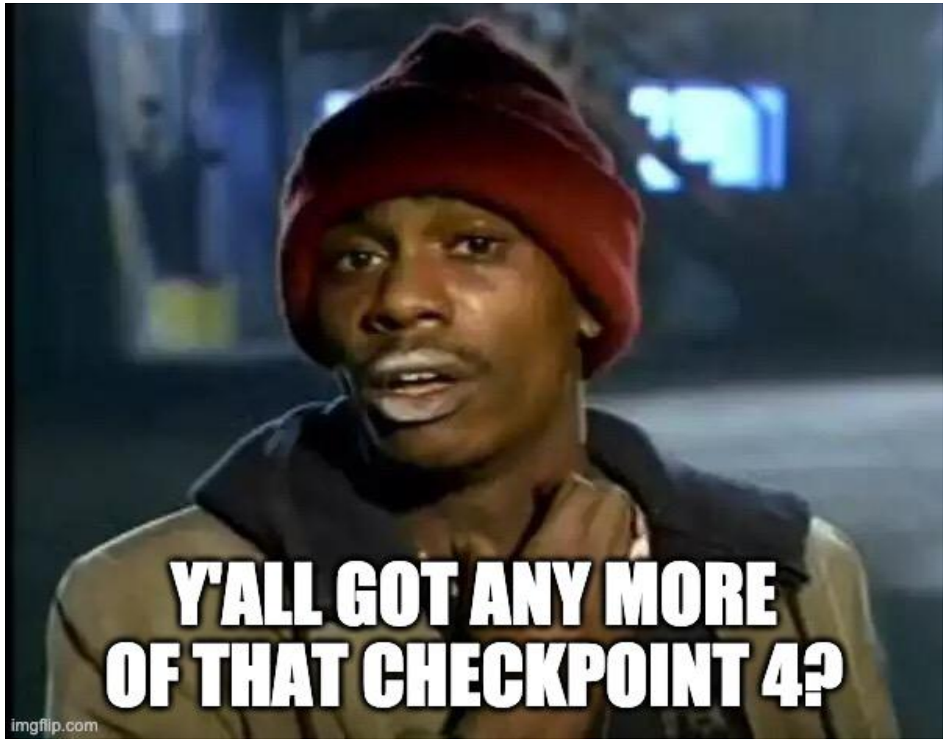
ECE 391 Discussion Week 13


ECE 391 COURSE STAFF WHEN CHECKPOINT 5 STARTS
imgflip.com


YOUR KERNEL
YOU
imgflip.com

WHEN YOU REALIZE THIS IS THE LAST CHECKPOINT FOR MP3
BUT YOU START IMPLEMENTING SCHEDULING
imgflip.com


Y'ALL GOT ANY MORE OF THAT CHECKPOINT 4?
imgflip.com

Announcements & Reminders

- MP3.5 is due Sunday (Dec 4th) at 5:59:59pm in gitlab
- Final demo be on Sunday 4th night, Monday 5th and Tuesday 6th
  - Details will be posted later
  - All Team members *MUST* be present at final demo
- Final exam
  - Tuesday (Dec 13th) at 7:00pm to 10:00pm

MP3.5 Multiple Terminals

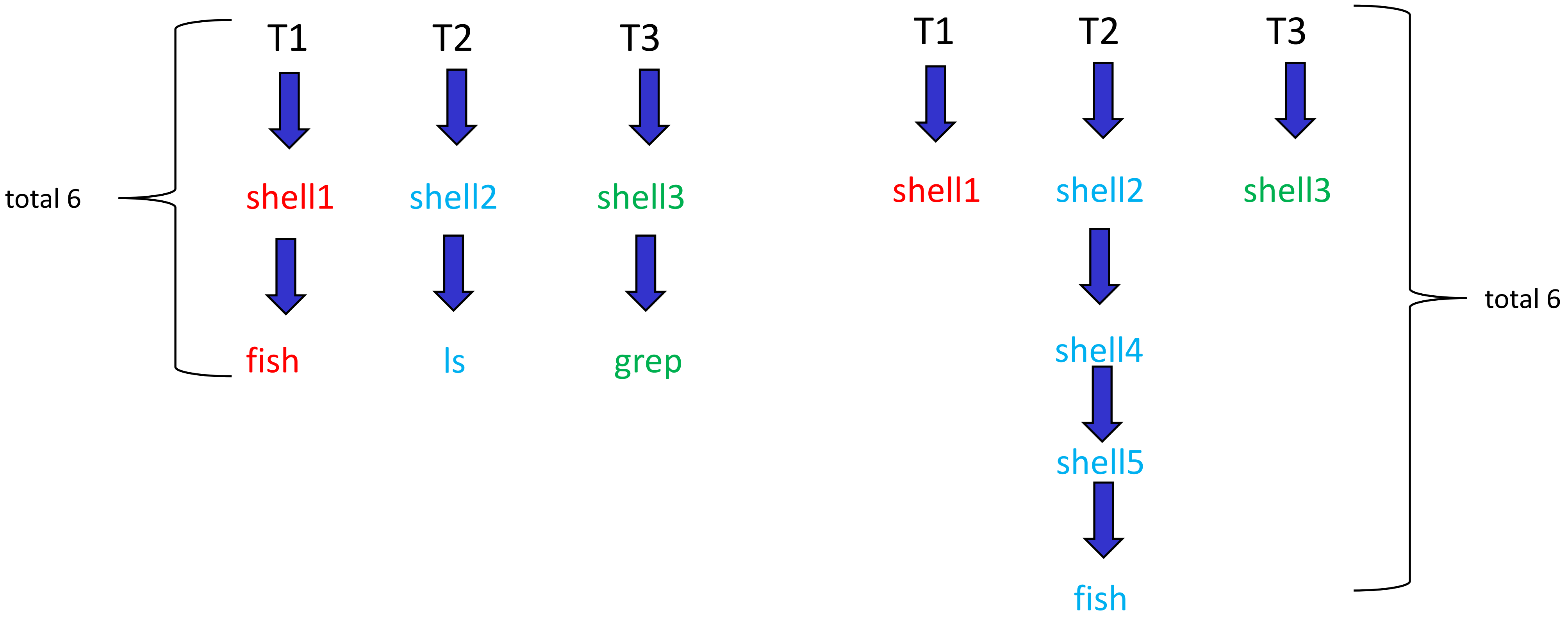
- Support 3 terminals (ALT+F# to switch between)
  - Initial bootup method: have all 3 terminals running shell
  - After bootup method: first time you press ALT+F2/ALT+F3, boot up terminal 2/3
  - Should be able to switch as fast as possible – no crashing!

Example coming up

MP3.5 Multiple Terminals

Support at least 6 processes total

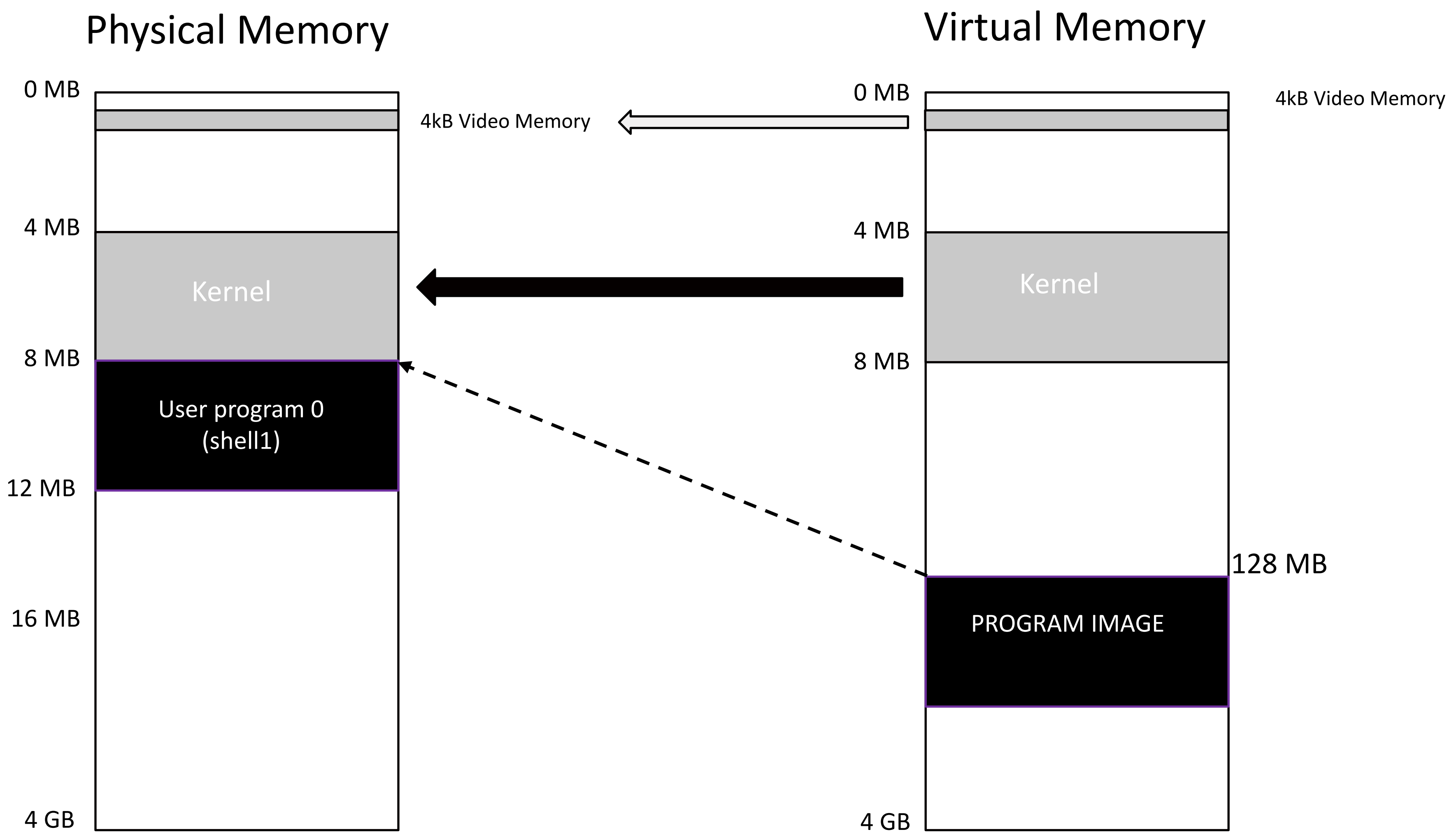
- How to manage 6 processes? Up to you…
- Don't crash or page fault on the 7th



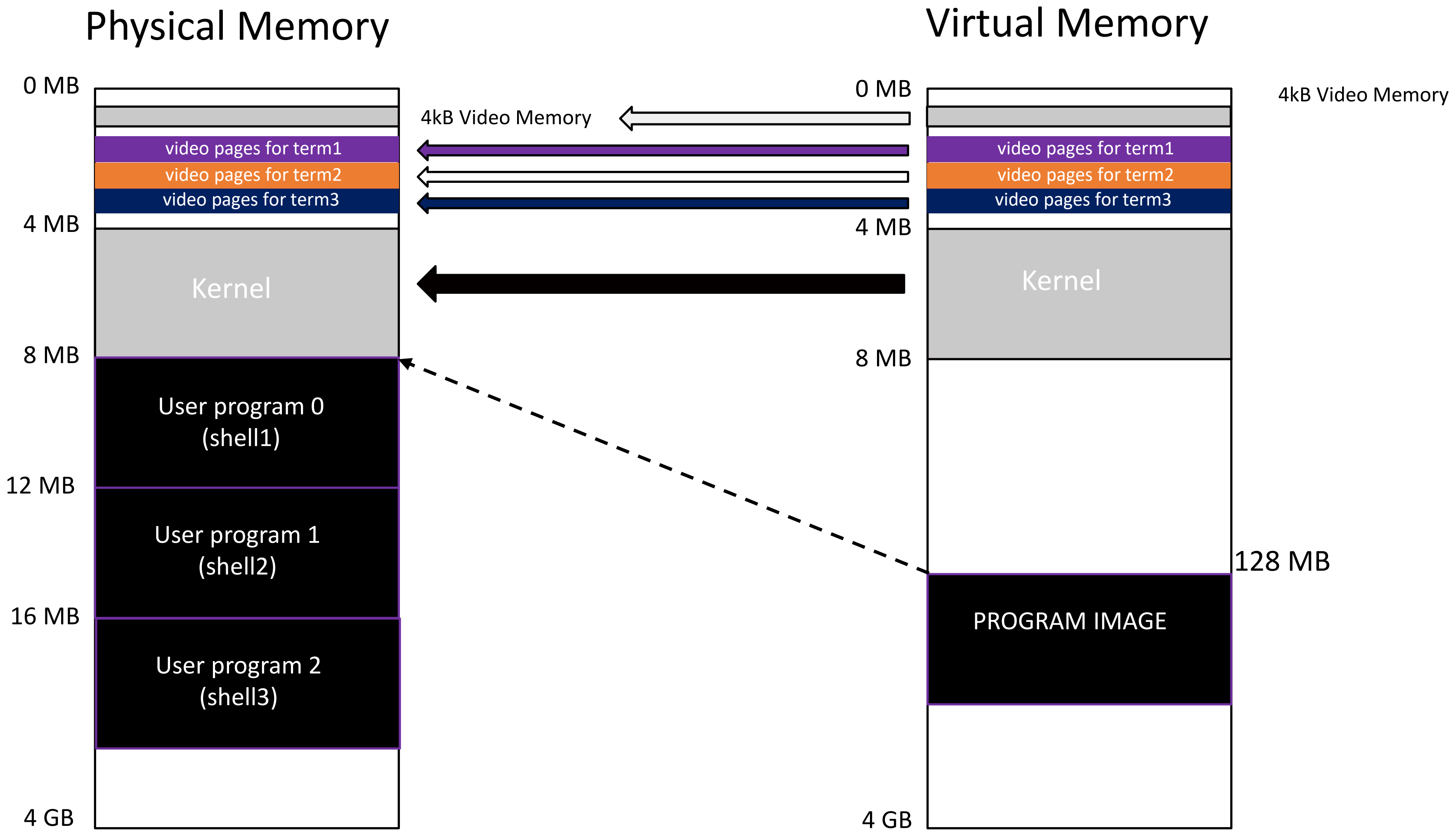
MP3.5 Multiple Terminals

- Separate input buffer per terminal, save current text screen/cursor position/anything relevant to your OS
- Exiting any shell does not exit other shells
- Exiting last shell: re-launch shell or prevent last one from being halted

Single Terminal



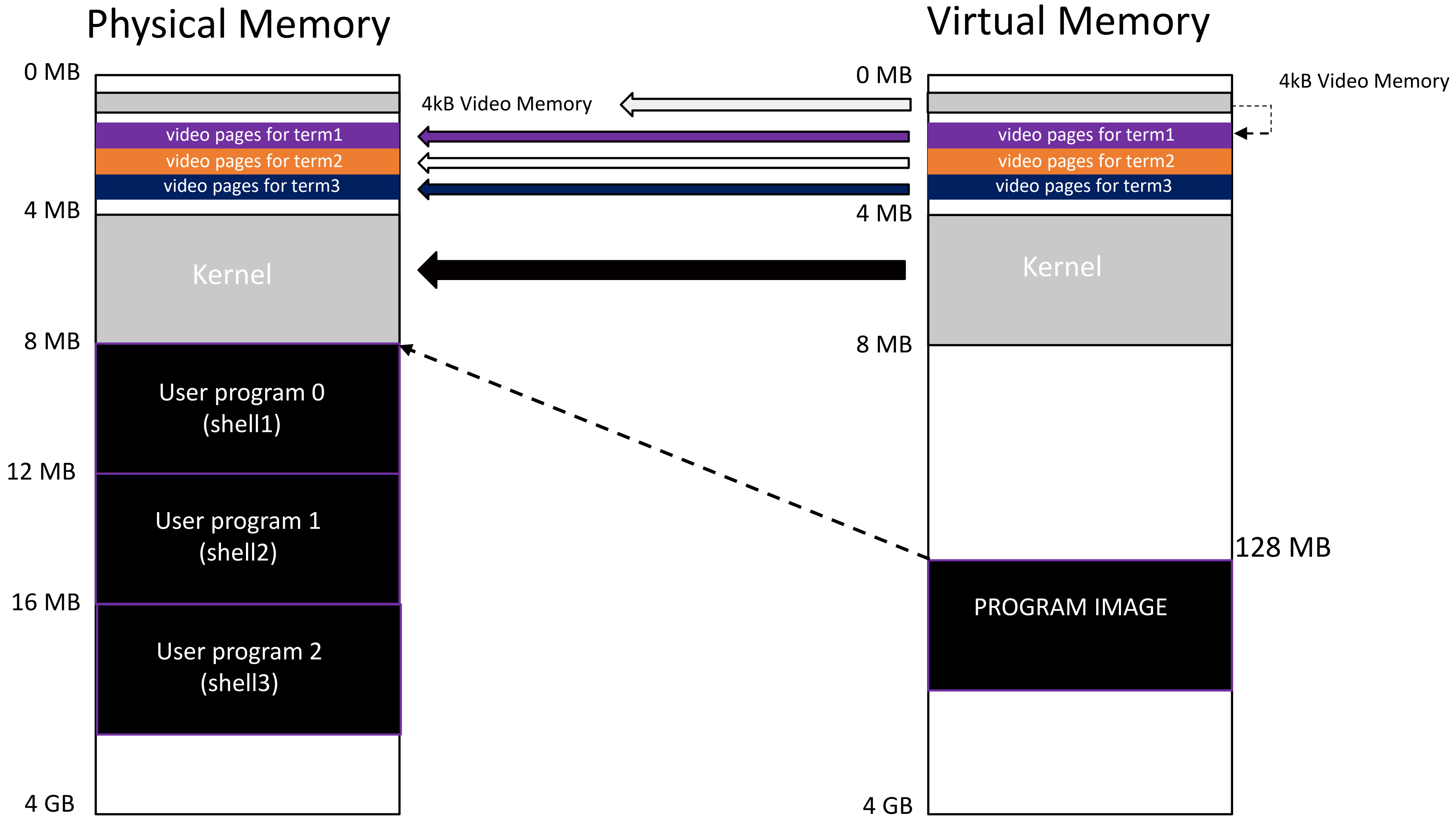
Three Terminals: Terminal 1 in use



What happens when you switch to another terminal?

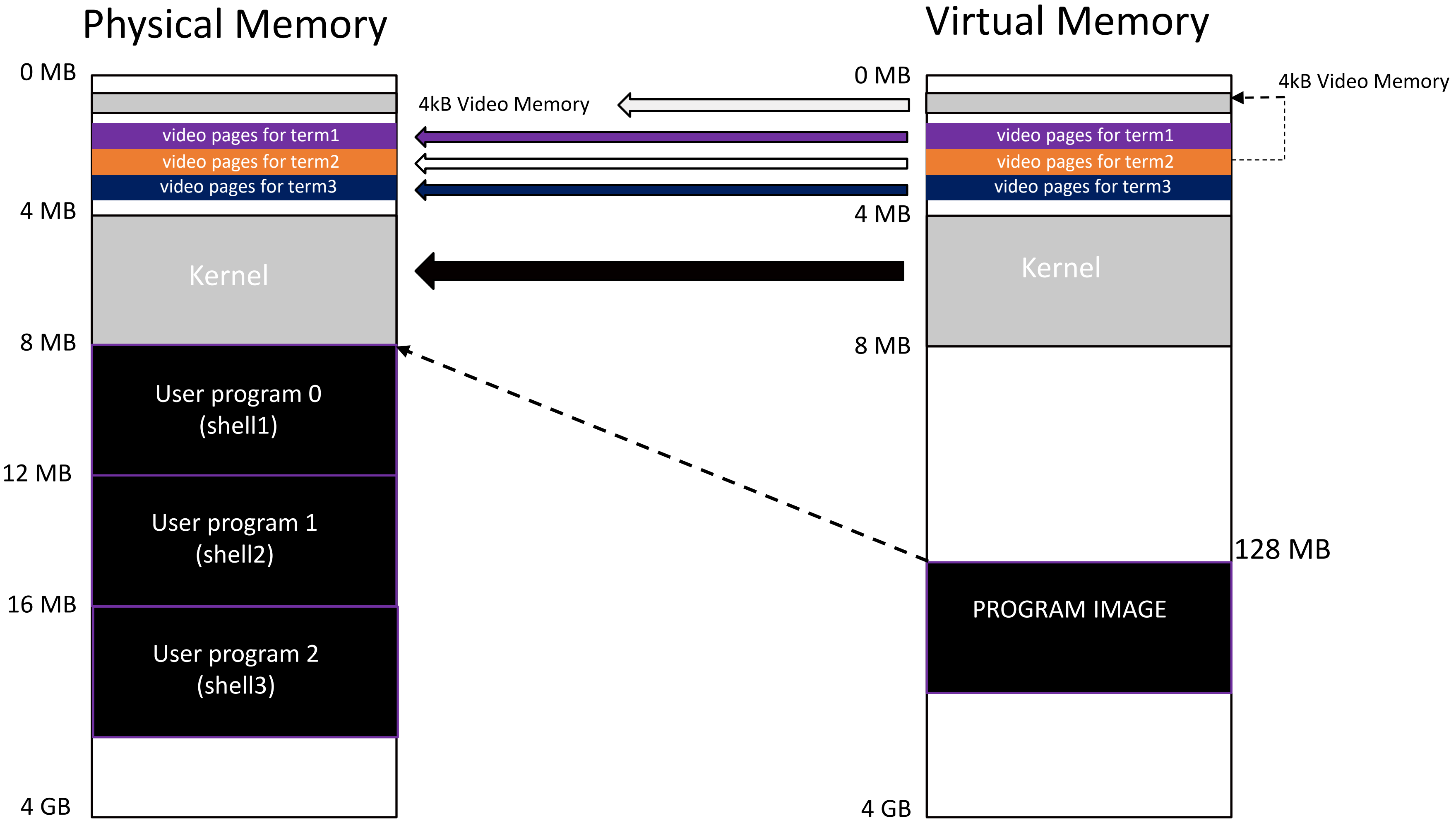
Three Terminals: Terminal 1 -> Terminal 2

1. Save terminal 1 screen to video page assigned for it



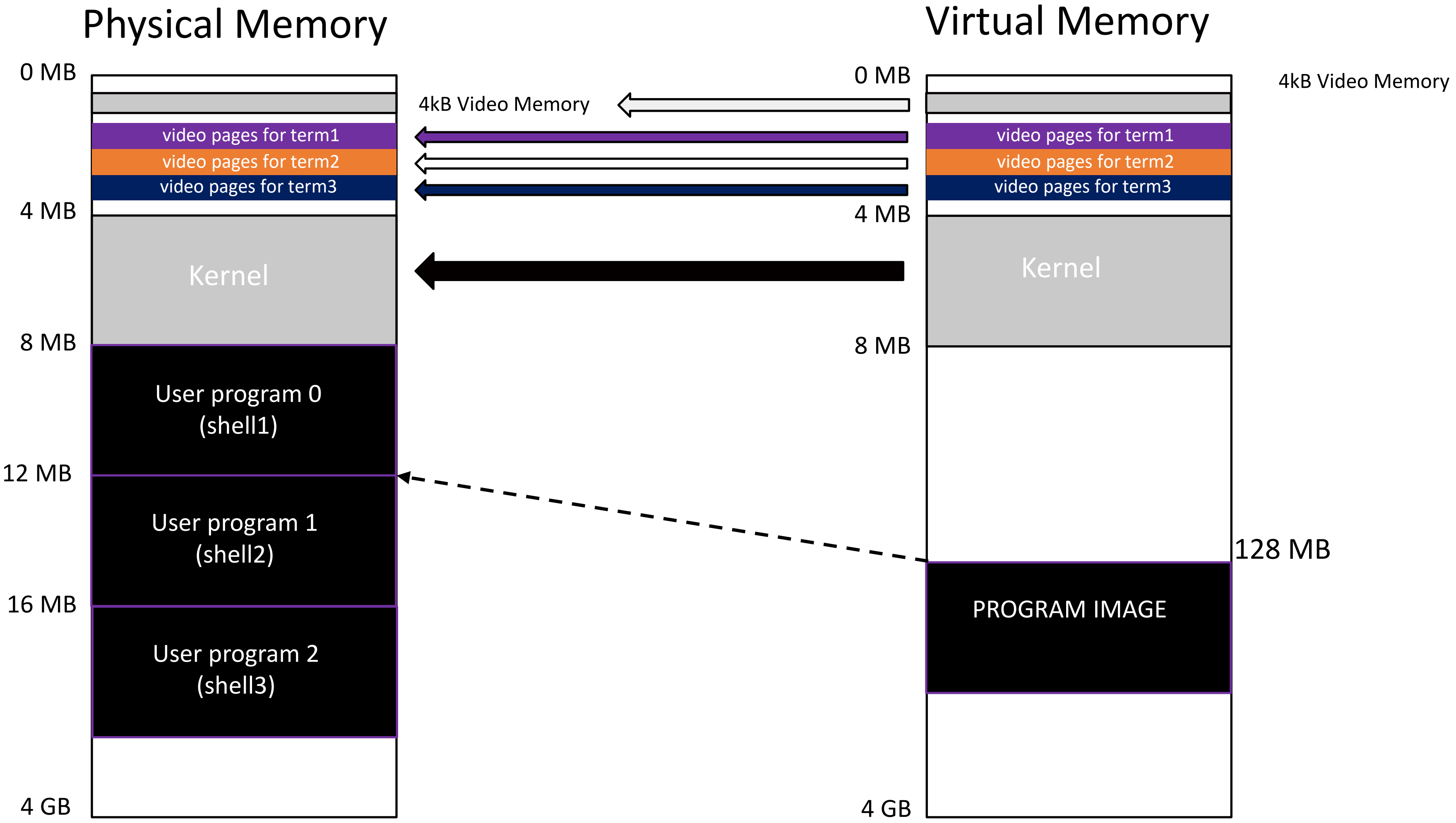
Three Terminals: Terminal 1 -> Terminal 2

2. Restore terminal 2’s screen to video memory



Three Terminals: Terminal 1 -> Terminal 2

3. Switch execution to terminal 2's user program



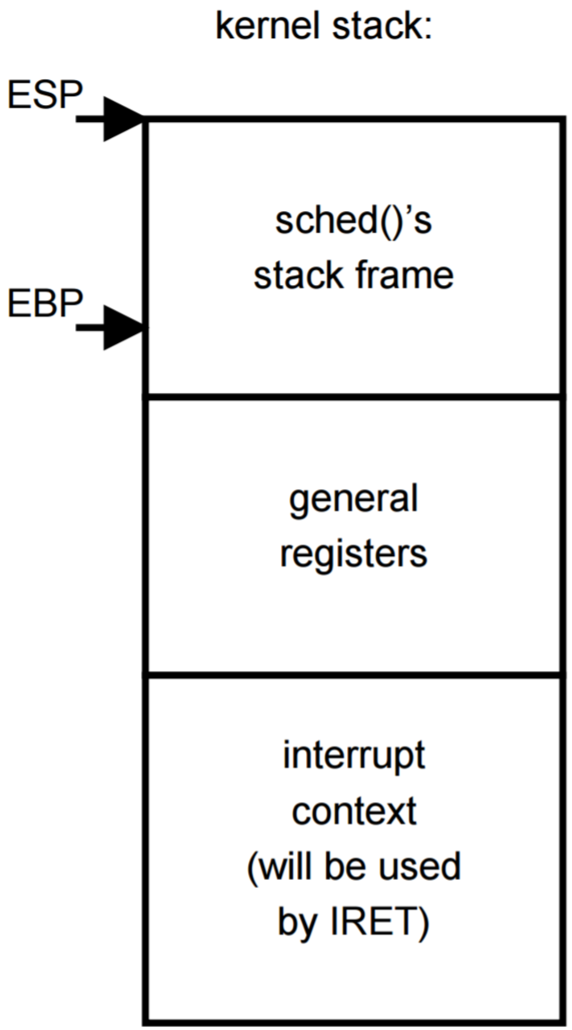
MP3.5 Scheduling

- Fixed/equal time slices with round-robin
- Use PIT (not RTC…why?)
  - Think about device priority and frequencies
- When typing, the characters should appear on the visible terminal (not the scheduled terminal)
- Test with counter, pingpong, fish
- Be mindful of timing/synchronization issues

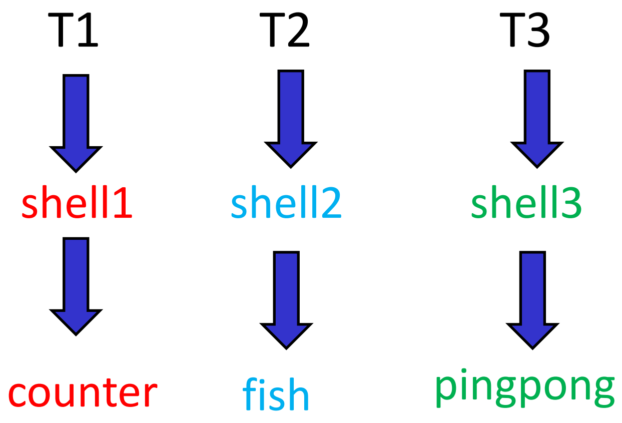
MP3.5 Scheduling

- Utilize the kernel stack (think about what you did for
  - You will be using assembly to do the context switch
  - Switch ESP/EBP to next process’ kernel stack
  - Restore next process’ TSS
  - Flush TLB on process switch

Example coming up



MP3.5 Scheduling

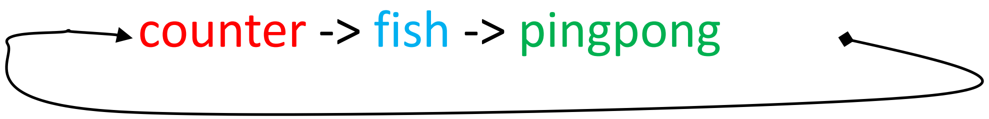


T1 → shell1 → counter

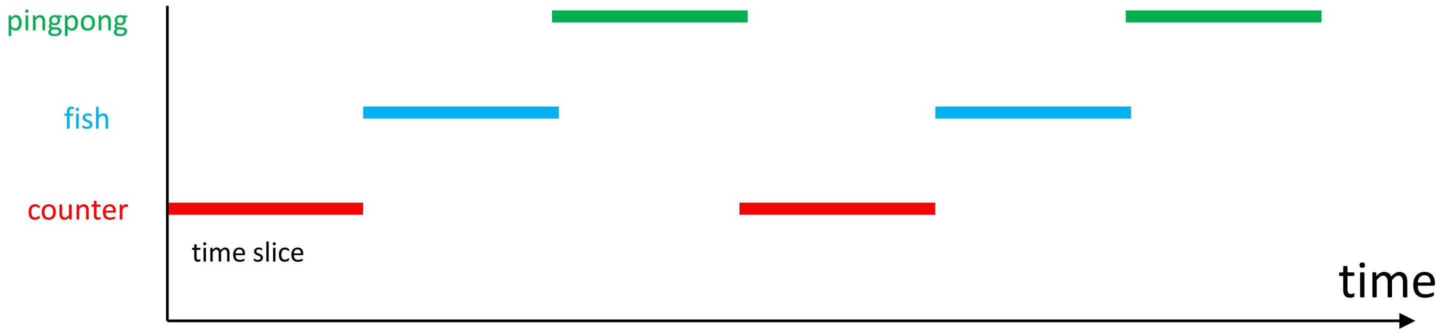
T2 → shell2 → fish

T3 → shell3 → pingpong

Round Robin scheduling



counter -> fish -> pingpong

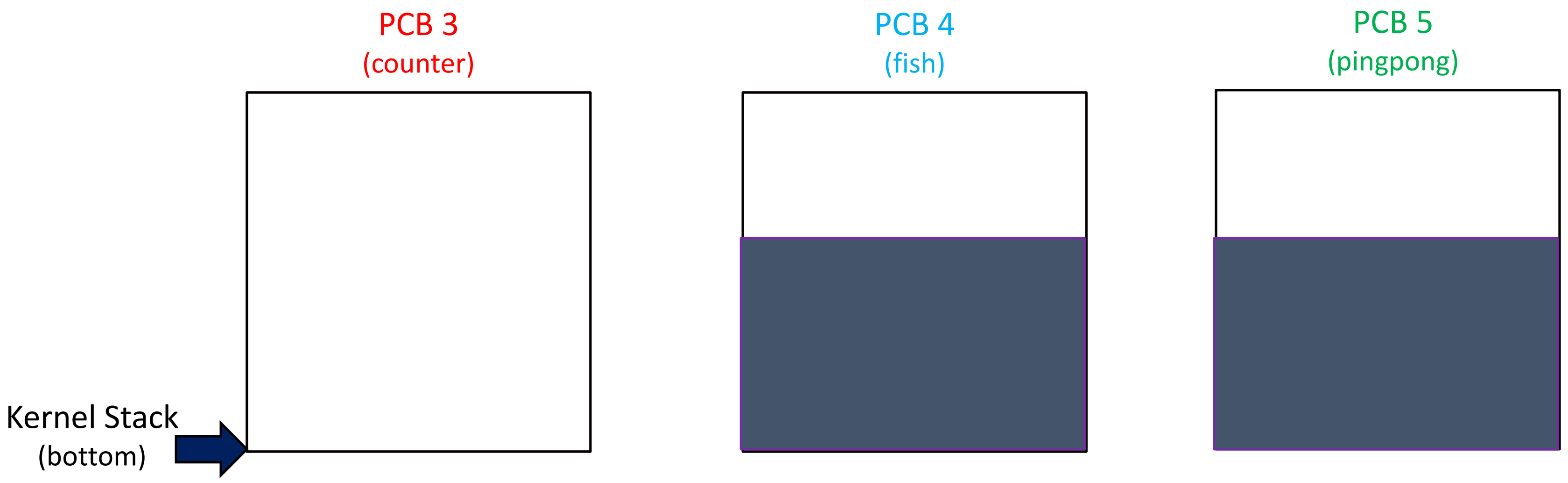


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

1) Counter is executing

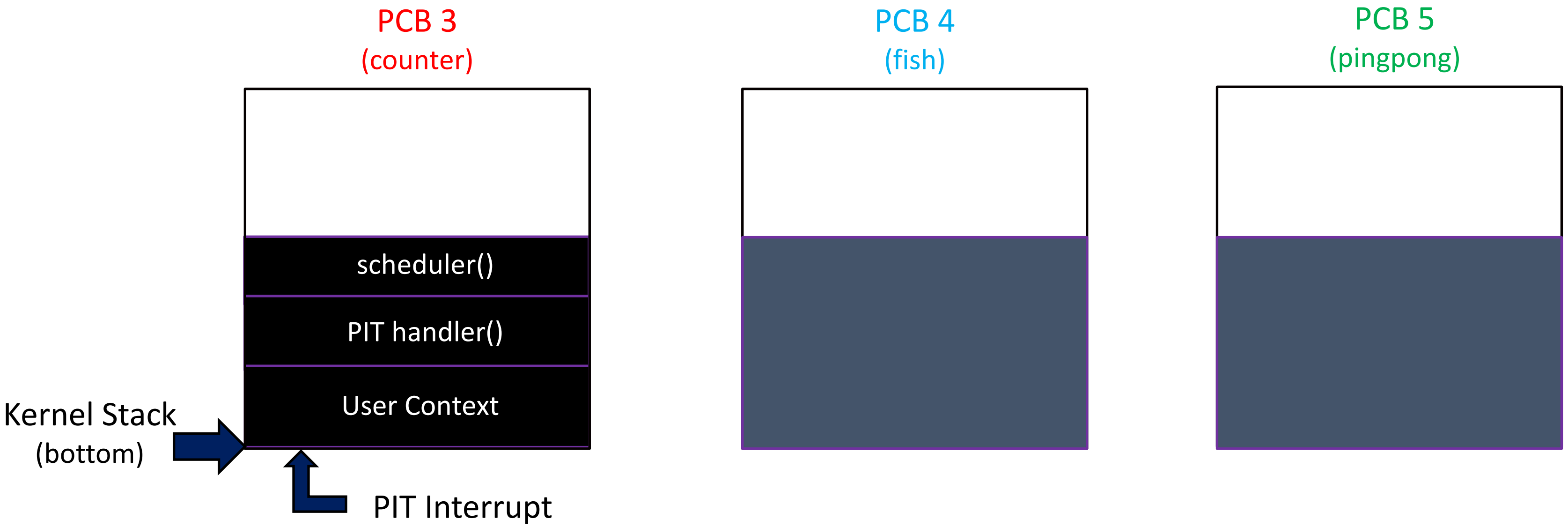


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

2) A PIT Interrupt occurs -> PIT handler executes -> Calls scheduling algorithm

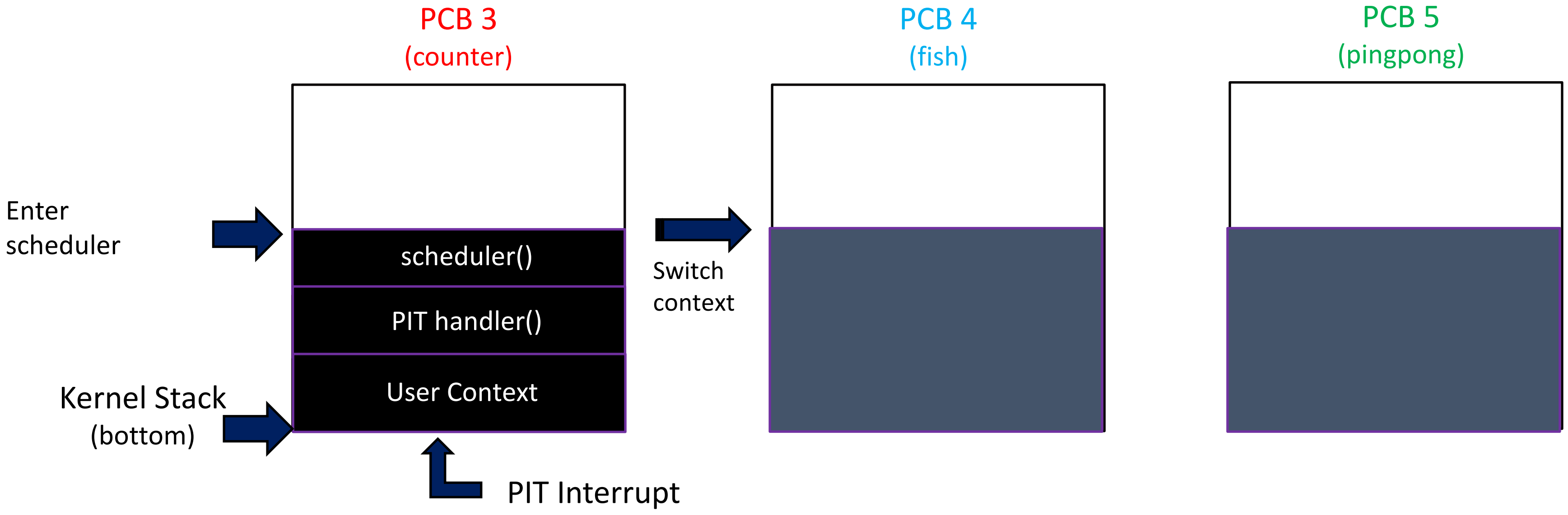


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

3) Context switch to next program in scheduling queue

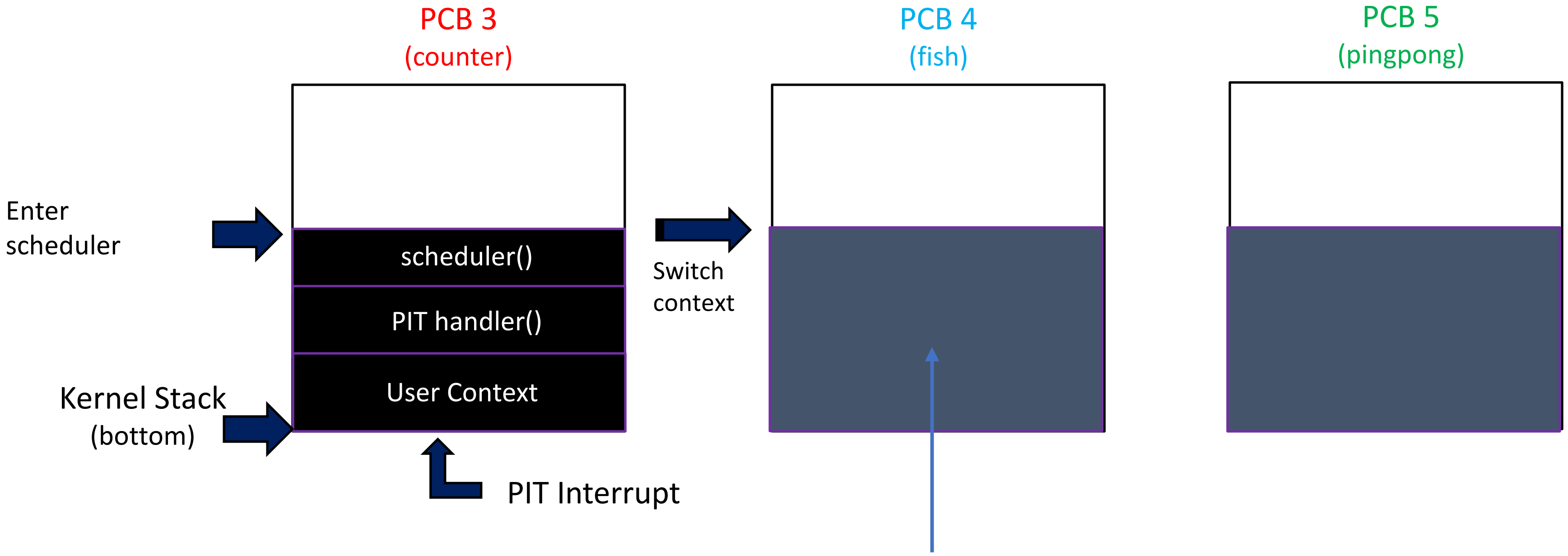


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

3) Context switch to next program in scheduling queue



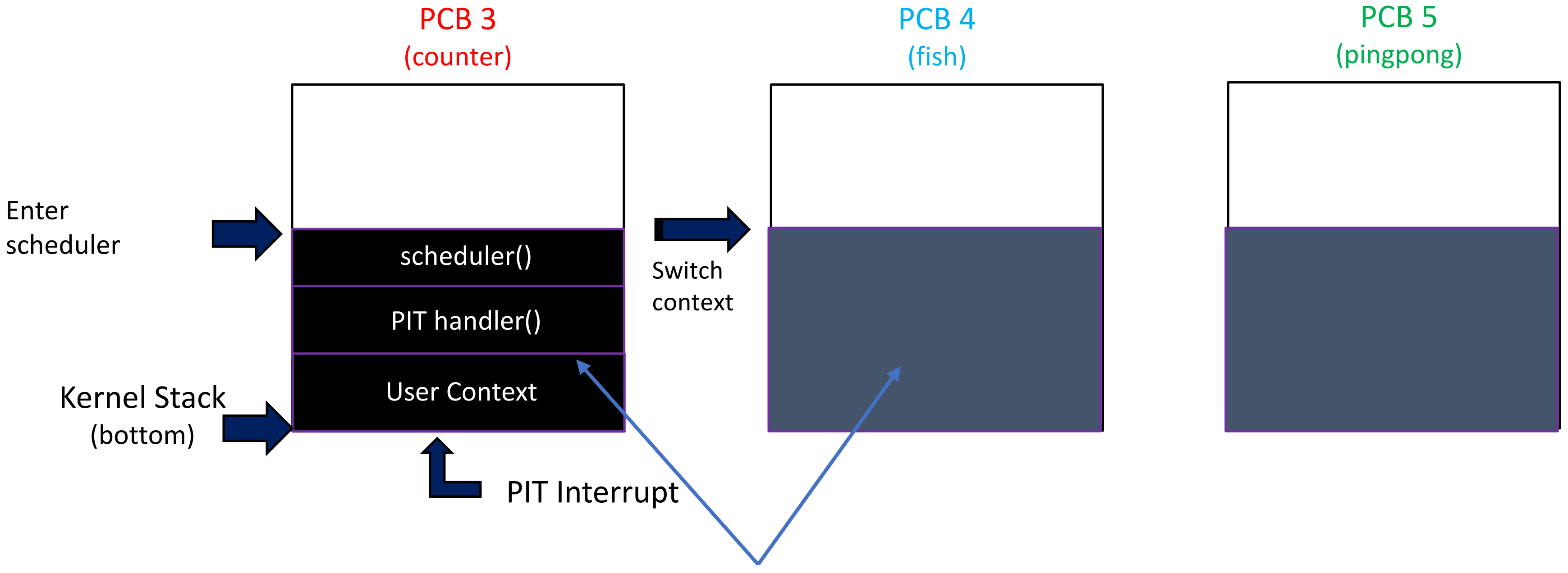
But whats here?

MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

3) Context switch to next program in scheduling queue



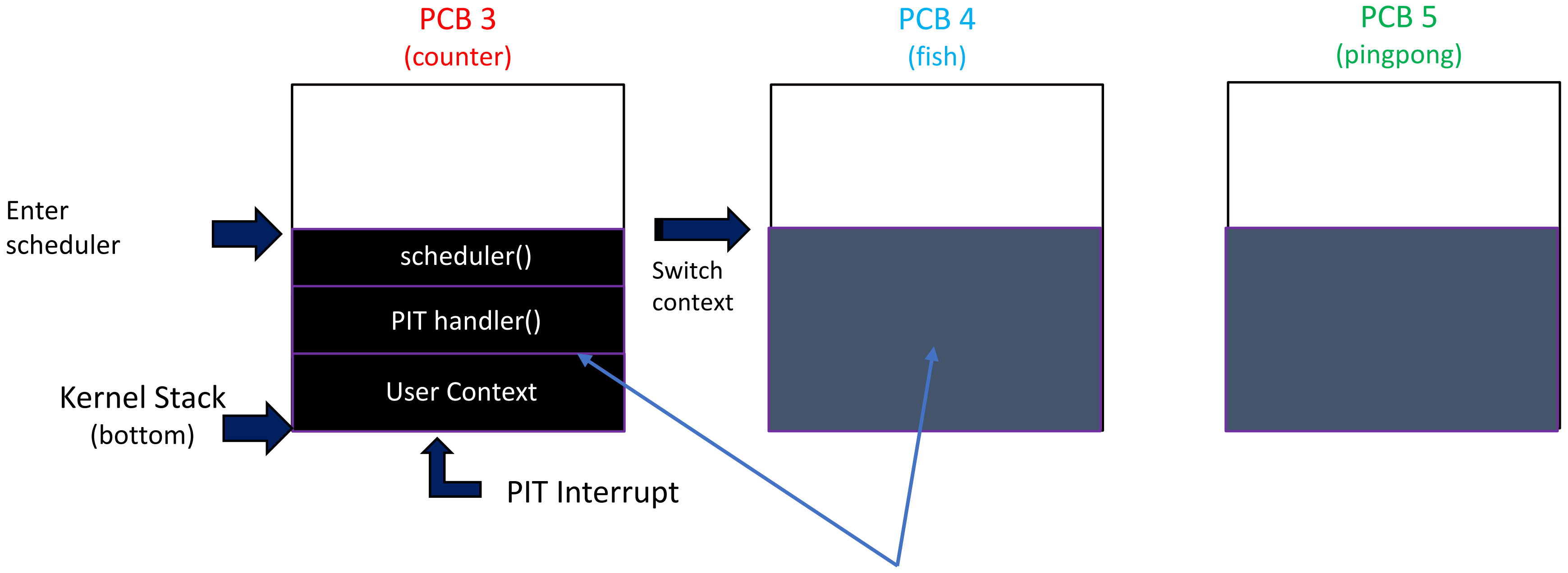
The same thing that’s here. Why?

MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

3) Context switch to next program in scheduling queue



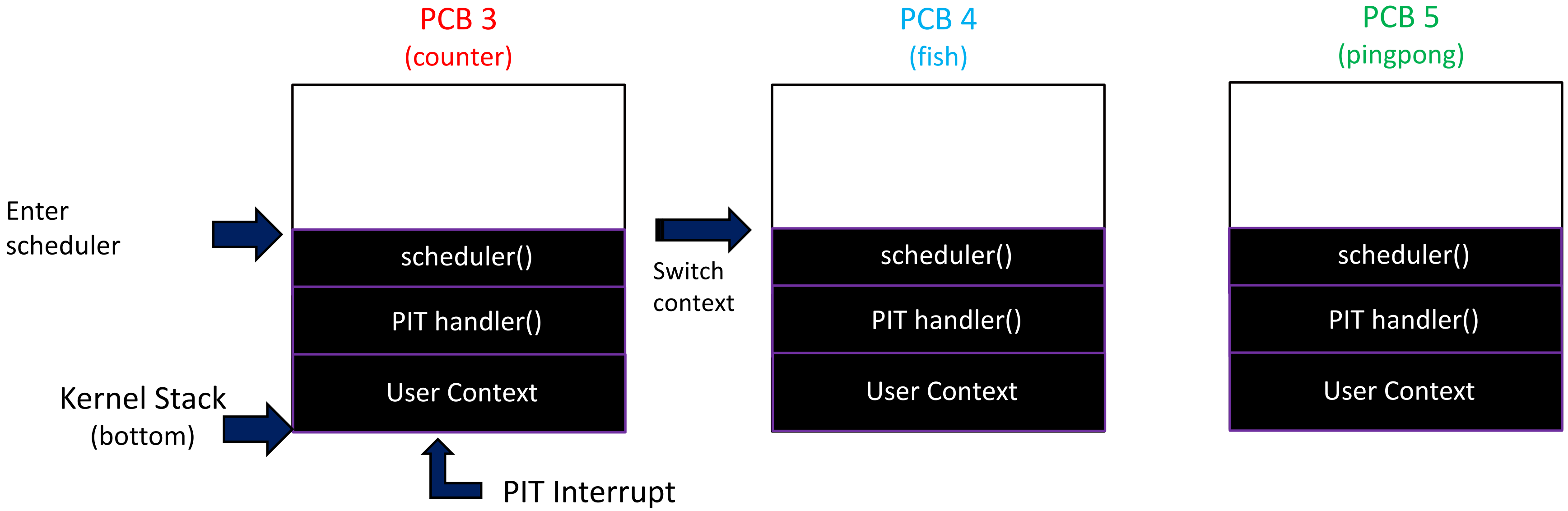
Only one way to get out of a process - the scheduler!

MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

3) Context switch to next program in scheduling queue

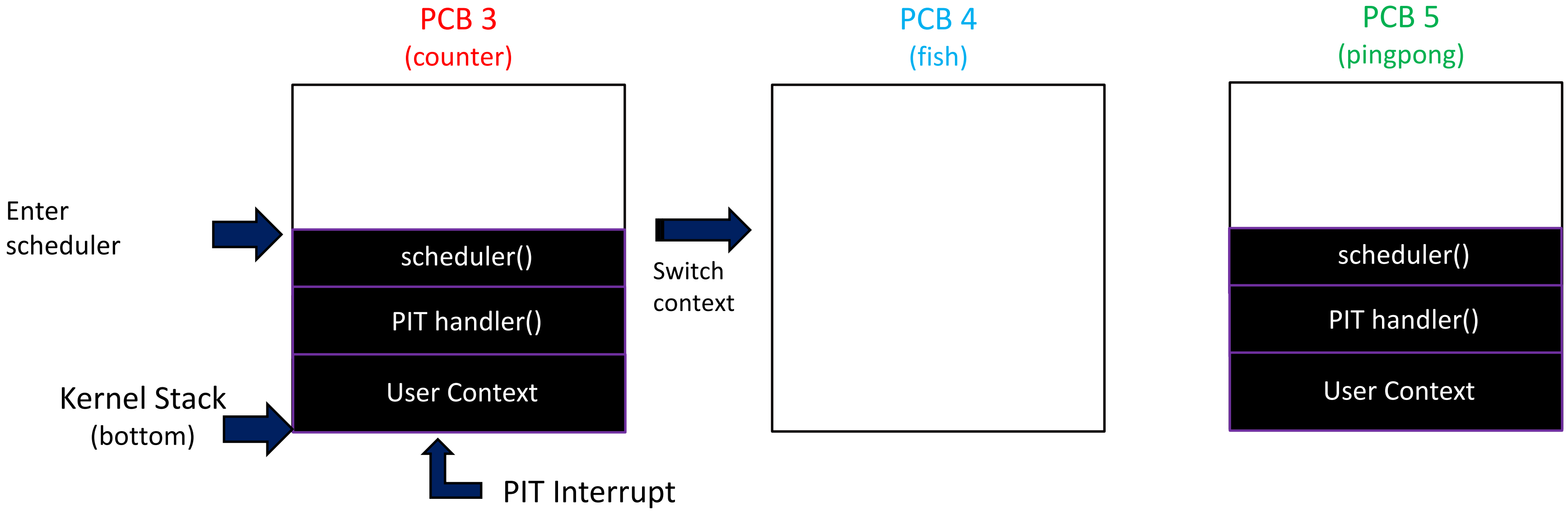


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

4) Return from PIT handler and execute fish

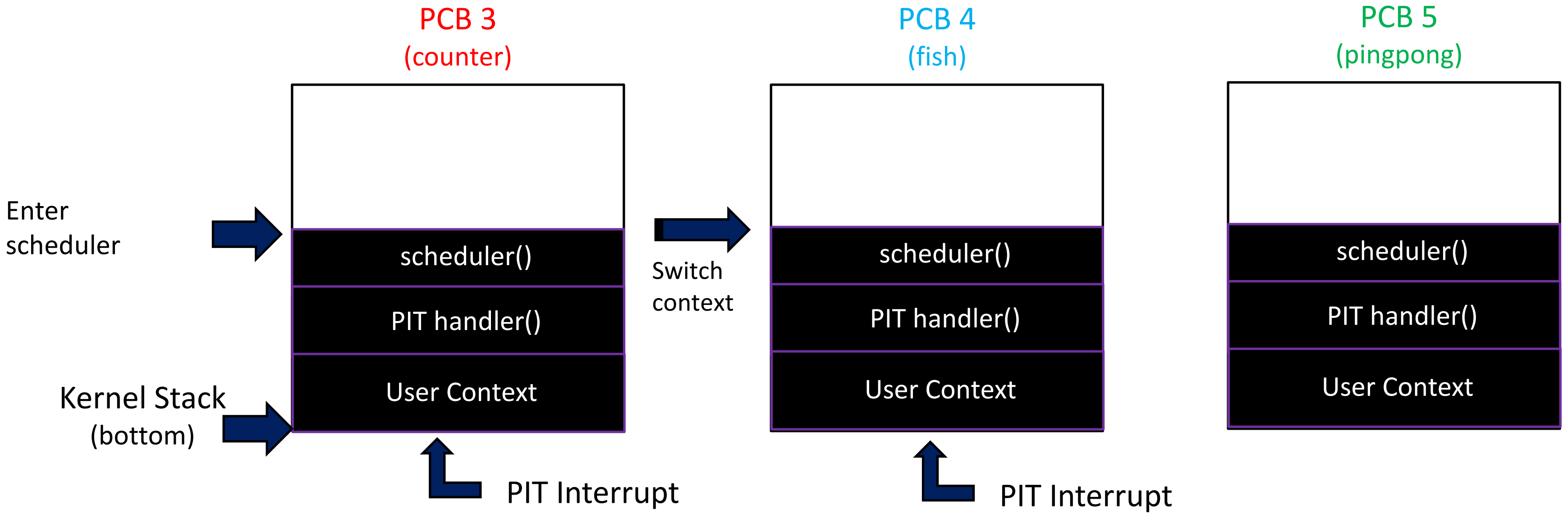


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

5) Another PIT interrupt occurs!

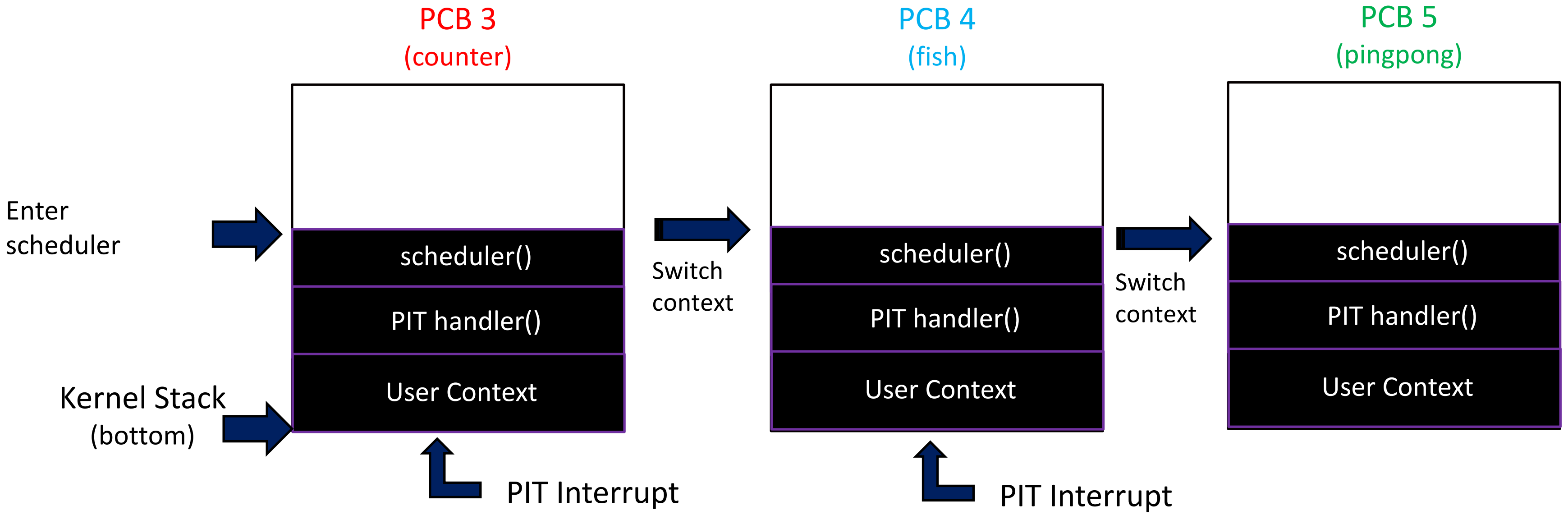


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

6) Context switch to pingpong

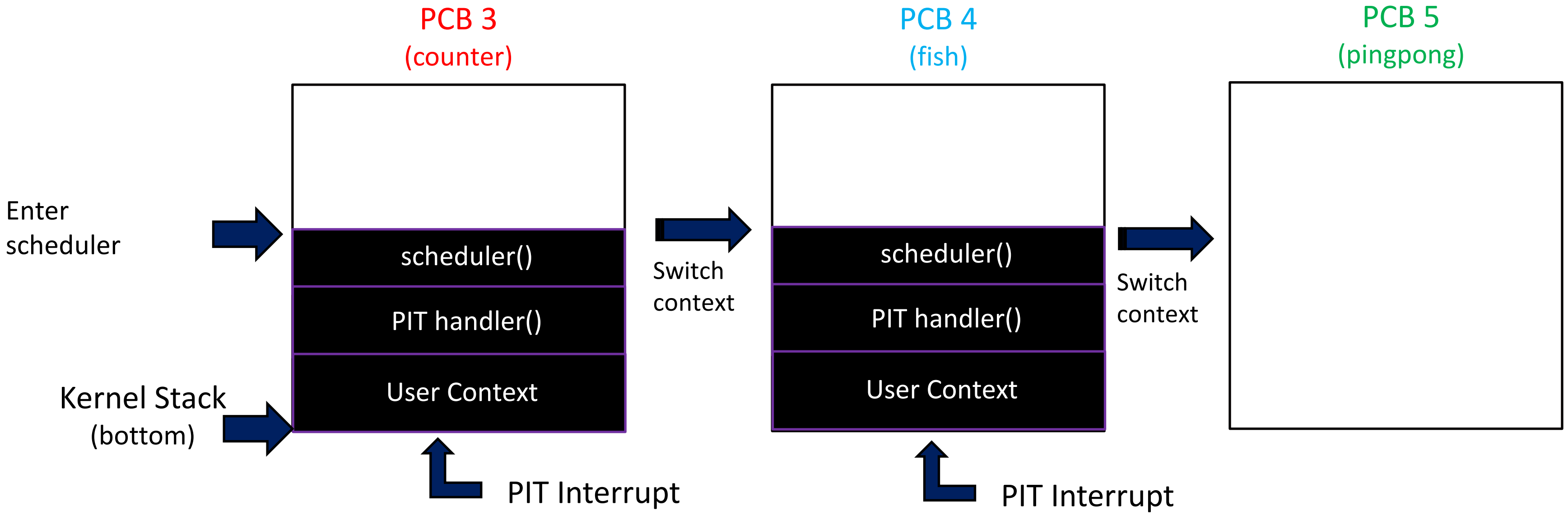


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

7) Return from PIT Handler and execute pingpong

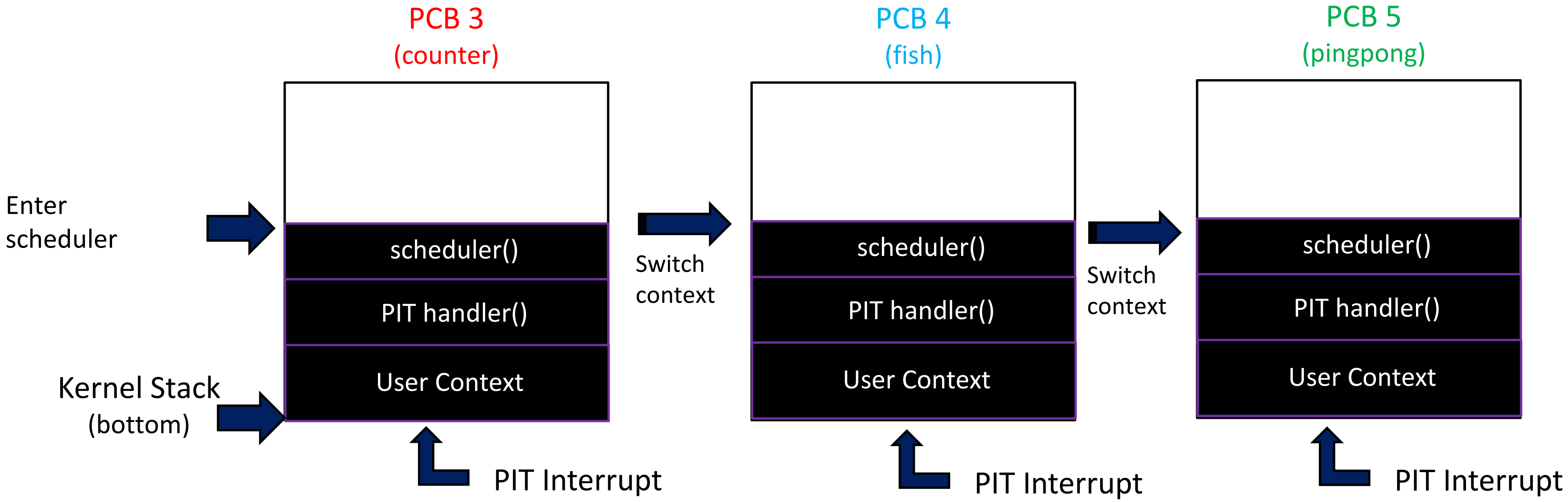


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

8) Another PIT Interrupt occurs!

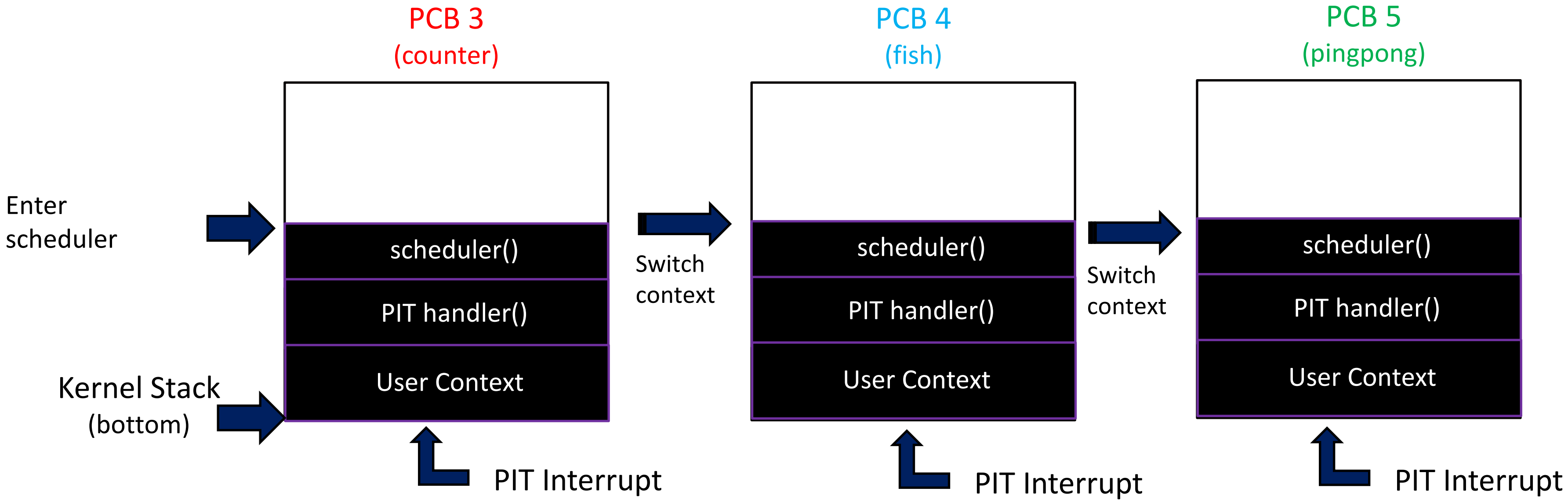


MP3.5 Scheduling

Setup:

Terminal 1: P0 shell -> P3 counter
Terminal 1: P1 shell -> P4 fish
Terminal 1: P2 shell -> P5 pingpong

9) Context switch to counter and execute it



AND
ANOTHER ONE

Sounds hard right?

It is.
Use your 2 weeks wisely

But wait, there's more!

They have to work together

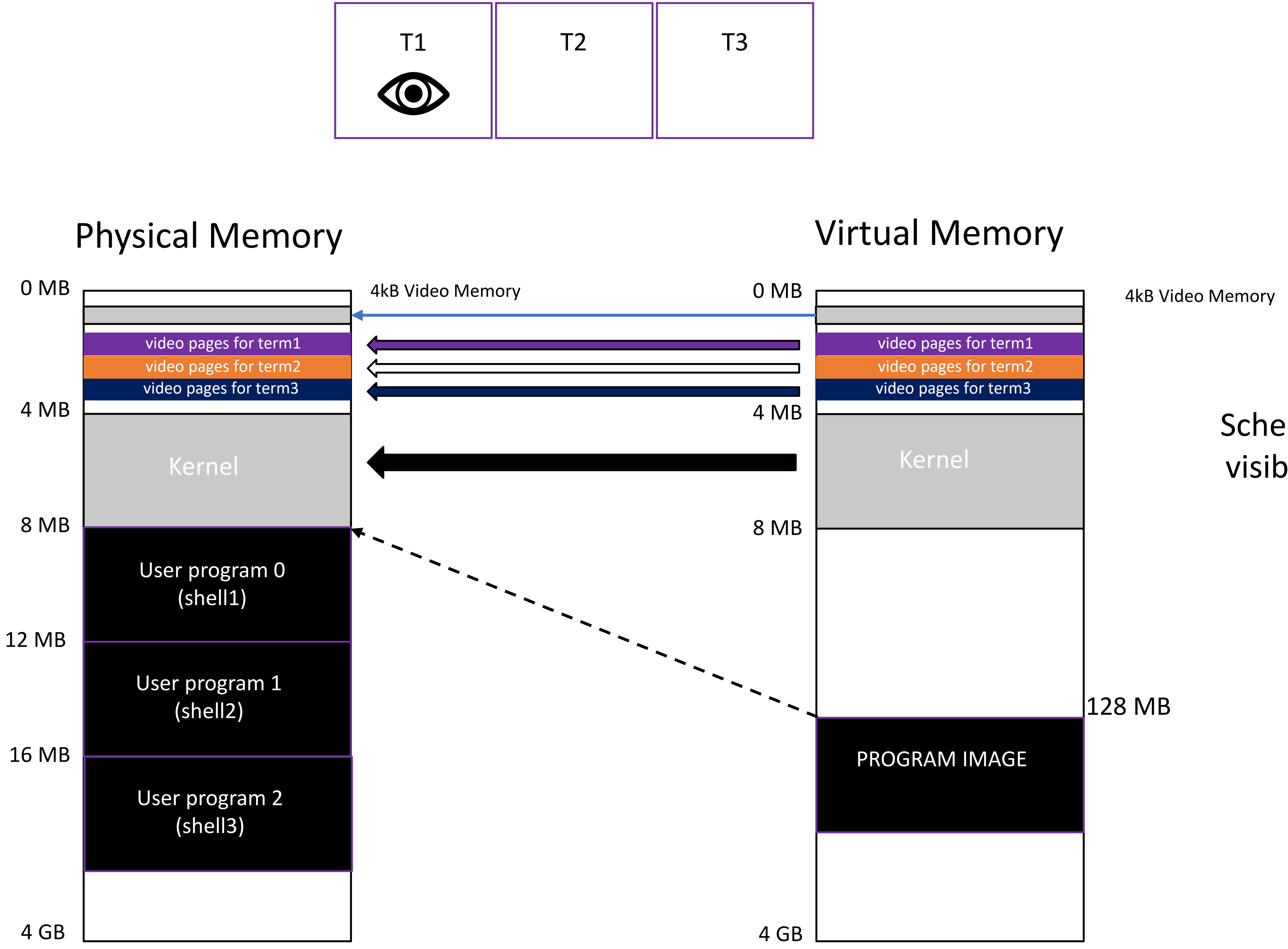
Implement Scheduler and Multiple Terminals separately

Implement Scheduler and Multiple Terminals together

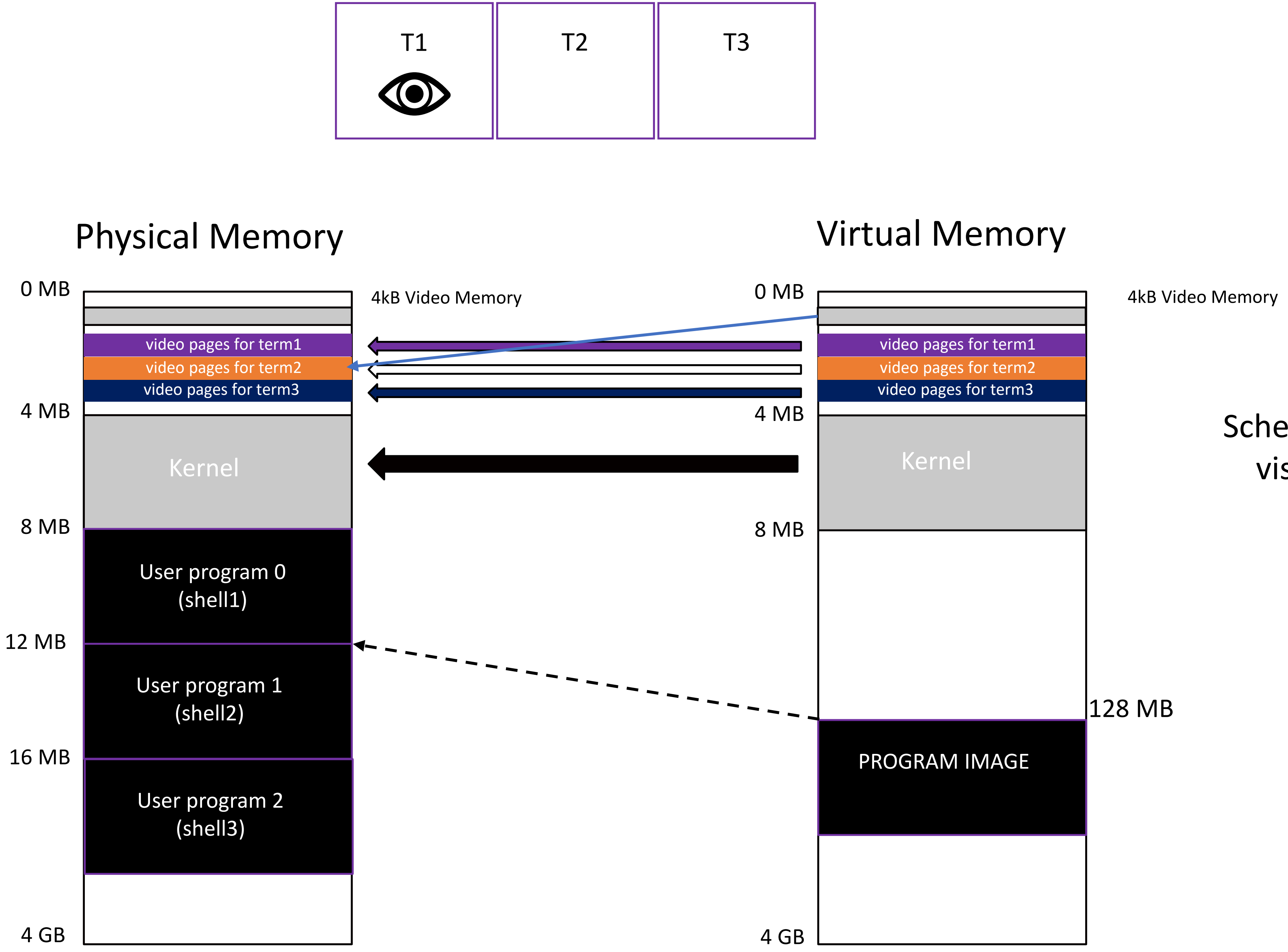
MP3.5

- The current scheduled process DOES NOT have to be the one running on the current terminal
- If currently scheduled process is not visible, write to background, not to main video memory
- Be careful with vidmap!

Example coming up



Scheduled process and visible process are the same



T1

T2

T3

Physical Memory

0 MB

4kB Video Memory

video pages for term1

video pages for term2

video pages for term3

4 MB

Kernel

8 MB

User program 0
(shell1)

12 MB

User program 1
(shell2)

16 MB

User program 2
(shell3)

4 GB

Virtual Memory

0 MB

4kB Video Memory

video pages for term1

video pages for term2

video pages for term3

4 MB

Kernel

8 MB

128 MB

PROGRAM IMAGE

4 GB

Scheduled process and visible process are different