

Topics in Operating Systems

Tue, Fri 3.30PM - 5PM @ KD 102

Course and instructors

\$whereis cs730

<https://www.cse.iitk.ac.in/users/deba/cs730/>

Piazza link: <https://piazza.com/class/jqf0fw7lsuw7nf?cid=4>

Moodle (TBA)

\$whereis deba

KD 212, deba@cse.iitk.ac.in , Meeting hours: 11AM - 1PM Thursday

\$ where is TA

S V Shanmuga Sunder (vgnsunder@cse.iitk.ac.in)

Course policy

Add/Drop

Course registration:- Ideally before next class

Last date of drop:- 25-Jan-2018 (strict, no further drops allowed)

Class guidelines

Keep your mobile phones switched off / silent

Ask questions and interact

Be on time!

Evaluation

1. Two programming assignments (25%)
2. Paper review, presentation and discussion (15%)
3. Project (40%)
4. End-semester (20%)

References

Operating Systems: Three Easy Pieces. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau.

Understanding the Linux Kernel, Daniel P. Bovet, Marco Cesati.

Linux Kernel Development, 3rd Edition, Robert Love.

Linux Device Drivers, 3rd Edition, By Jonathan Corbet, Greg Kroah-Hartman, Alessandro Rubini.

Linux kernel documentation

Research papers

Evaluation

1. Two programming assignments (25%)
2. Paper review, presentation and discussion (15%)
3. Project (40%)
4. End-semester (20%)

“Take pride in honest hard work ”

“Cheating implies accepting defeat”

“If you are here to learn, never defeat the purpose by cheating”

<https://www.cse.iitk.ac.in/pages/AntiCheatingPolicy.html>

CS730 enables you to ...

- Not panic if someone asks “Can you write code in kernel?”
- Provide guidance to conceive and critically analyse new directions
- Empirically analyse different designs at a finer granularity
- Think alternate design choices to
 - Meet application level demands
 - Exploit advanced hardwares
- Build confidence in implementing new ideas in an OS

CS730 is not ...

- Another or advanced (in traditional sense) OS course
- Writing a new operating system
- Creating a new linux distribution
- Understanding Linux kernel in entirety

Logistics and homework -1

- Personal laptops with 1.5hr backup desirable
- **HW1: Setup a Virtual machine for the course (Due before next lecture)**
 - Create a Linux VM (Ubuntu Linux recommended) (KVM is preferred)
 - Download the Linux kernel version - linux-4.19.13
 - Compile and boot the latest kernel

OS refresher: True/False

1. Superuser (e.g., root user in UNIX) in a multi-tasking OS can execute all instructions provided by the hardware instruction set architecture (ISA).
2. Every process in a computer system is guaranteed to be in running state at least once during its life time.
3. A critical section consisting of a single instruction may require mutual exclusion.
4. A user process interrupted by a device interrupt is always scheduled immediately after the interrupt is serviced.
5. A page fault can be handled without changing any page table entry.

OS refresher: Quiz (1)

In a uniprocessor system, in which of the following case(s), a process executing in user mode can cause an entry into the OS?

- A. accessing a general purpose register like RAX
- B. executing a JMP (jump) instruction
- C. decrementing an unsigned integer value stored in a register beyond zero
- D. executing a printf() statement
- E. returning from a function

OS refresher: Quiz(2)

You have added code at the beginning of the functions mentioned in the question to print the stack pointer every time the function is executed. Which of the following statement(s) are true regarding the value of stack pointer?

- A. For a given user-level function, value of stack pointer can be different.
- B. For a given OS-level function which is exclusively used by a system call handler (no other OS code calls this function), value of stack pointer can be different.
- C. For two different OS-level functions, the value of stack pointer can be same.
- D. For two different functions from two different application, value of stack pointer can be same.