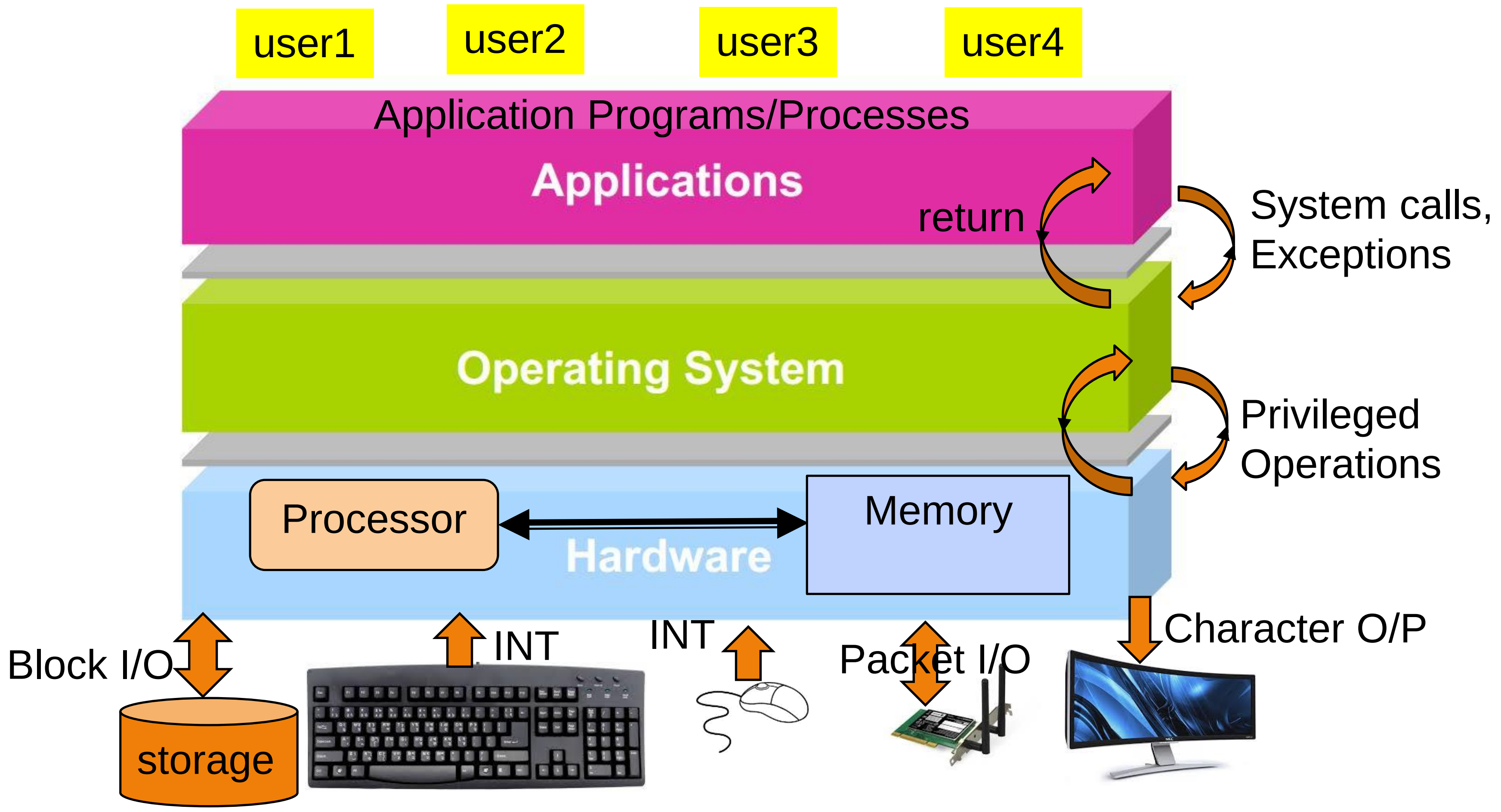
L2 - A CONVERSATION ON SUCCESS IN THIS COURSE AND BEYOND

CSCE-313 Spring 2017

A Quick Recap from our earlier discussion



Operating System is at the heart of it all!

- For the hardware, OS is:
  - Driver (knows how to use h/w, apps do not know)
  - Manager (when to run what … scheduling)
  - Protector (stops malicious ones)
  - Illusionist (tells each app “the h/w is all yours”)
  - ….
- Why H/W needs managing/protecting?
  - Processor
  - Memory
    - Capacity
    - Protection
  - Display

Theme of Today’s Discussion

- What does success in this course look like?
  - Tangibles
  - Non-Tangibles
- Tips for success

Evolution of this course in the last 2 years

	FA'14 (Tyagi)	SP'15 (Tyagi)	FA'15 (Tyagi)	SP'16 (Tyagi)	FA'16 (Ahmed)	SP'17 (Tyagi)	FA'17 (Planed)
Textbook	Previously used Text	**New – Anderson & Dahlin**	No Change	No Change	No Change	No Change	Free Online Text?
Lecture Material	Minor-Mid Refresh	**Major Overhaul**	No Change	Delta Change	Delta Change	Delta Change	Modern Topics
HW/Quiz	HW	HW	In class quizzes	In class quizzes	In class quizzes	Take-home quizzes	**TBD**
Machine Problems	Previously used MP	No Change	1 new MP	No Change	**Major Overhaul**	Delta w/ feedback	Delta w/ feedback
Grading Platform	CSNET	CSNET	E-campus	**Vocareum**	Vocareum	**Vocareum ++**	No Change
DEV Platform	Dept. Linux Server	Dept. Linux Server	Dept. Linux Server	Dept. Linux Server	Dept. Linux Server	Dept. Linux Server	**R-PI?**

Some Observations from Past Semesters - **Exams**

- Exams are mostly problem centric with few sprinkles of objective T/F, Multiple Choice questions
- Midterm – Content is relatively easier so class tends to do better compared to the Finals
- Finals – Content is difficult primarily attributed to threaded code synchronization and inter-process communication. Also, we end up competing with multiple course priorities at the end

Some Observations from Past Semesters – **Machine Problems**

- Machine Problems are now more or less synced with classroom content
- With good teamwork and proper planning, it is fairly easy to earn maximum allocated points
- We continue to struggle in facilitating productive and rewarding teamwork mostly due to large classroom size
  - Move to Vocareum platform should help steer TA and peer teacher time to assisting with execution and quality
- Department Server based instruction acts as a constraint to unfettered learning
  - Long term vision is to migrate to a platform which is extremely forgiving to experimentation ☺

Success in CSCE-313 - **Tangibles**

- Overall grade of B or better
- 80% or better overall score in exams – typically means that student ‘got’ the key concepts right
- 90% or better overall score in machine problems – translates to a demonstrated ability to write quality system programs and understanding of system programming concepts
- 90% or better in quizzes – translates to demonstrated ability to absorb classroom discussion

Tips for Succeeding in Exams

- **Target no more than 10% missed days of attendance**.
  - Offline 'catching up' is tempting but not productive.
- **Take notes**.
  - There's more said on each slide than what is written. Concepts are inter-woven so offline reading of slides carries the risk of missing key linkages.
- **Practice problem solving for each topic**.
  - Provided quizzes will help.
- **Make sure that concepts are clear**.
  - With that, you will be assured of solving any problem with a correct approach.

Machine Problems

ID	Machine Problem	Key Learnings	Complexity
MP1	High Performance Linked List	C++ refresh, cost of system calls	LOW
MP2	Memory Allocator	Memory Management	MED-HIGH
MP3	System Calls and Critical OS Functions	Inner workings of some key system commands	LOW
MP4	UNIX Process	Anatomy and Attributes of a UNIX process	LOW
MP5	UNIX Shell	Creation and Execution of a Unix Shell, basic functions	MED

Machine Problems (contd.)

ID	Machine Problem	Key Learnings	Complexity
MP6	Scheduler	Scheduling Policies	MED
MP7	Threaded Client-Server	Threading	LOW
MP8	Advanced Client-Server	Threading, Synchronization	MED
MP9	IPC Mechanisms	Threading, Synchronization, IPC Mechanisms	MED

Some additional comments on Machine Problems (MP's)

- MP's have three main components
  - Scenario understanding and system/algorithm development
  - Coding, debugging, debugging, and debugging (gdb)
  - Performance analysis
- MP's are closely aligned to subject matter covered in class with one exception:
  - MP1,2 are primarily constructed to act as a refresher for C++ coding and Pointer Arithmetic. MP2 is a build over MP1. MP2 is perhaps the most intense of the MP's.
- Starting with MP6, the remaining MP's incrementally build on top of the previous one to seamlessly tie concepts learned in class.

Tips for Succeeding in Machine Problems

- **On the day of MP release**: Independently Review and build a 1st cut understanding.
    - Don’t defer this, because many MPs have only 1-wk deadline
- **By the 2nd day of MP release**: Meet with your partner and create a micro-schedule (key phases and timelines) of how you plan to develop and execute the solution. Take each other’s schedule and commitments into account.
- **Resist the rush to start coding.** Spend adequate time understanding the problem and building your plan (algo, structure). Tie it all the way to the end game.
- **Lookout for postings** (questions, clarifications, announcements) on Piazza. Set your alerts appropriately.

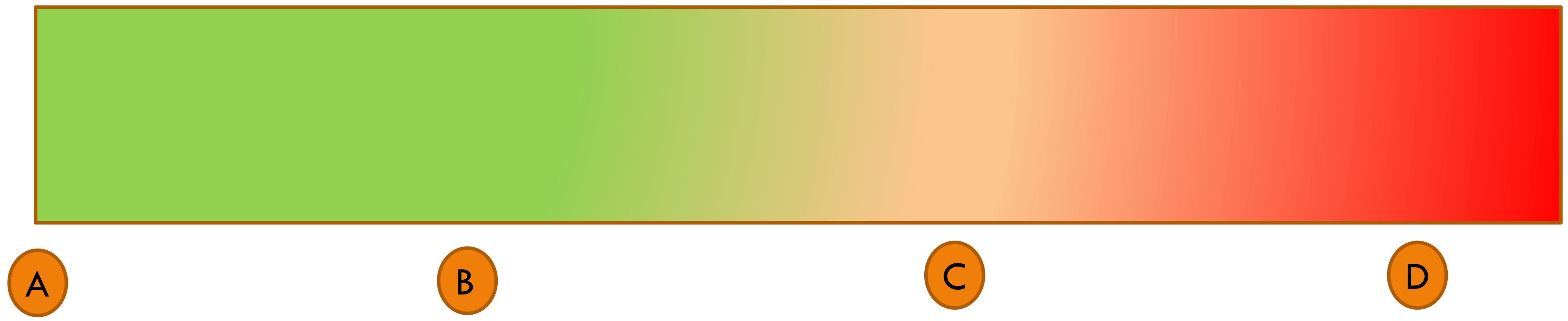
Tips for Succeeding in Machine Problems

- **Engage me, your TAs, and peer teachers** to seek timely clarifications
- **Set your own deadline at least 2 days ahead of the published deadline.**
  - Inevitably you will find issues (compilation, inability to run across all test cases, poor code performance, etc.) that may take up the final days to recover and still meet deadline.
- **Enjoy the problem while you work on it.**
  - The problems are designed to mimic (somewhat) what you will encounter in a professional environment.

Teamwork

- **Observation**: The best functioning teams are those where each partner contributes equal amount and brings complementary skills (technical and organizational) to the table
- **Avoid shortcuts and over-reliance on your teammate**.
  - On our side, we will be looking for indicators to verify equity of effort.
  - On your side, pitch in equal amount of effort just because solving these problems will infuse strong credibility in your resume!

Collaboration Guidelines



LETTER	DESCRIPTION
A	(OKAY) – Completely Independent Original Work
B	(OKAY) – Collaborate across teams on ideas and architecture
C	(NOT-OKAY) – Submitting portions of someone else’s code/detailed pseudo code
D	(NOT-OKAY) – Submitting someone else’s code as yours

Open QnA