

CSCE 313-200
Introduction to Computer Systems
Spring 2018

Preliminaries

Dmitri Loguinov
Texas A&M University

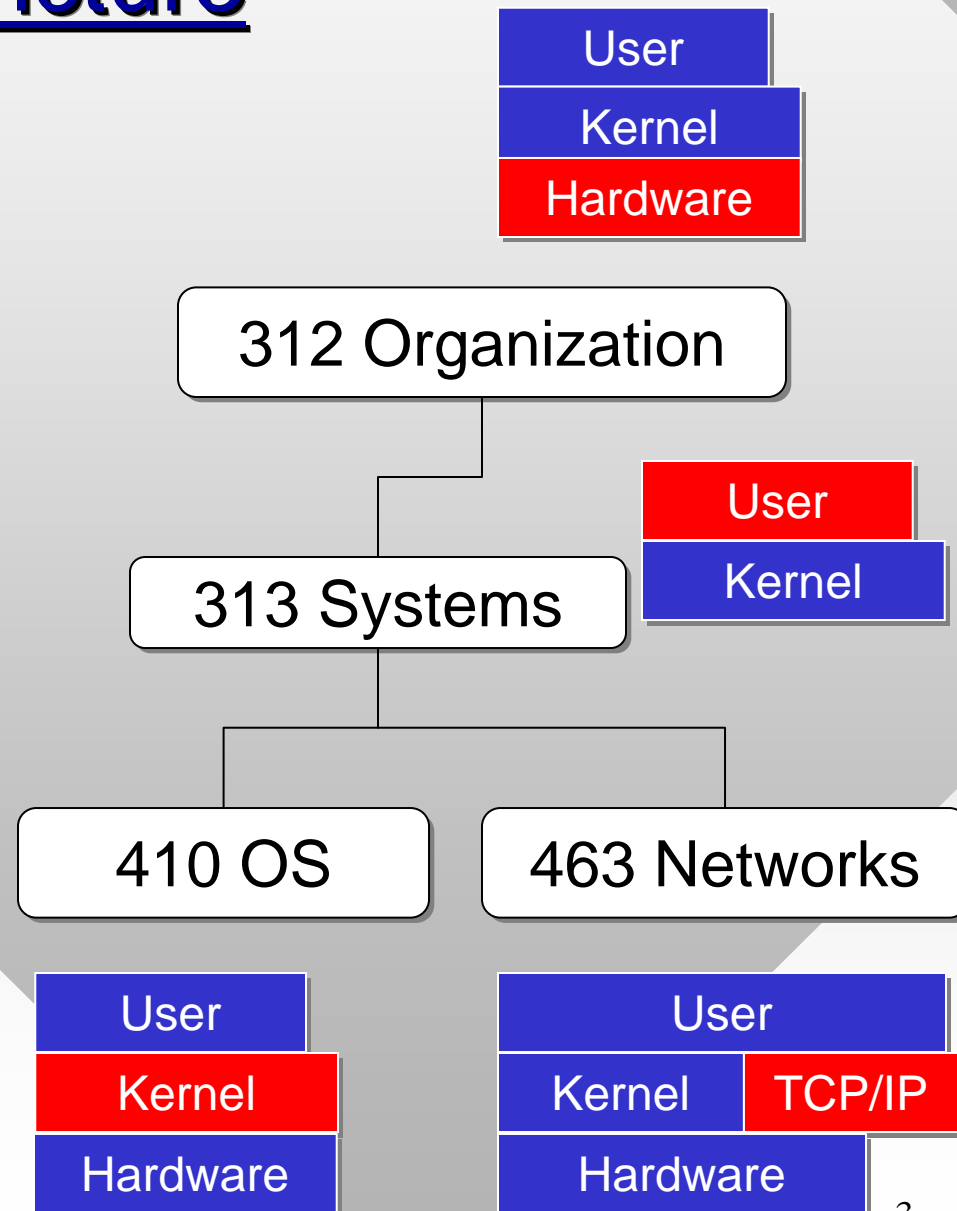
January 16, 2018

Preliminaries: Agenda

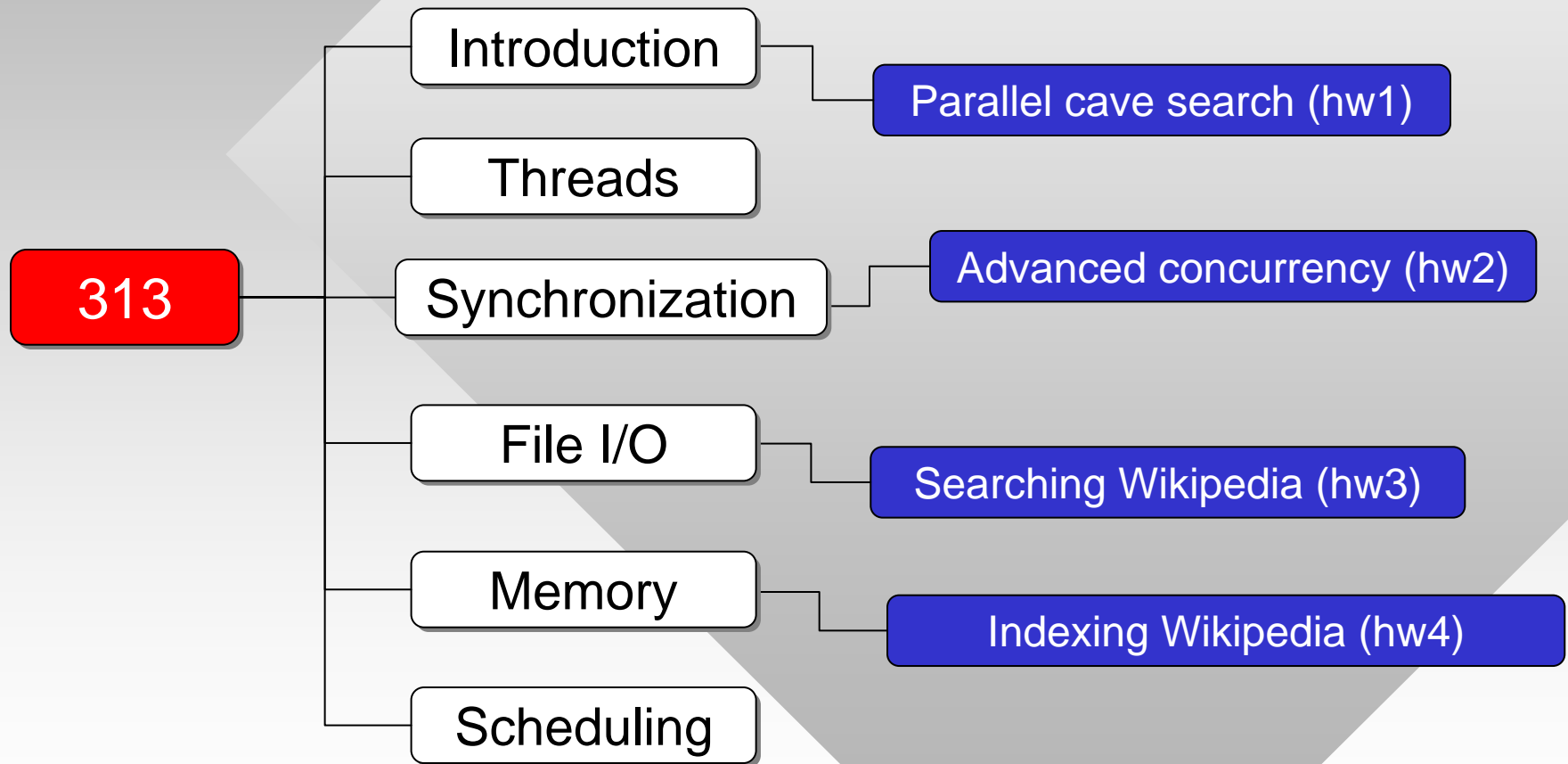
- **Big picture and roadmap**
- Syllabus
- Academic integrity
- Homework expectations
- VS 2017 demo
- Wrap-up

Preliminaries: Big Picture

- This course covers the **user** level
- Serves as a foundation for 410 and 463
 - They go deeper into the kernel
- **Systems programming**
 - Key factor is app **performance**
 - Benchmarking, optimization, efficiency will be our focus



Preliminaries: Roadmap



Preliminaries: Agenda

- Big picture and roadmap
- **Syllabus**
- Academic integrity
- Homework expectations
- VS 2017 demo
- Wrap-up

Preliminaries: Syllabus

- Instructor: **Dmitri Loguinov**
 - Office hours: TR 6:45-7:45 pm in HRBB 515C
- TA: **Di Xiao**
 - Office hours: MF 11-12 pm in HRBB 501C
- Books:
 - W. Stallings, “Operating Systems: Internals and Design Principles,” Pearson, 9th edition (2017)
 - J.M. Hart, “Windows System Programming,” 2010
 - J. Duffy, “Concurrent Programming on Windows,” 2008
- Hardware (helpful):
 - Any multicore computer with at least 16GB of RAM

Preliminaries: Syllabus 2

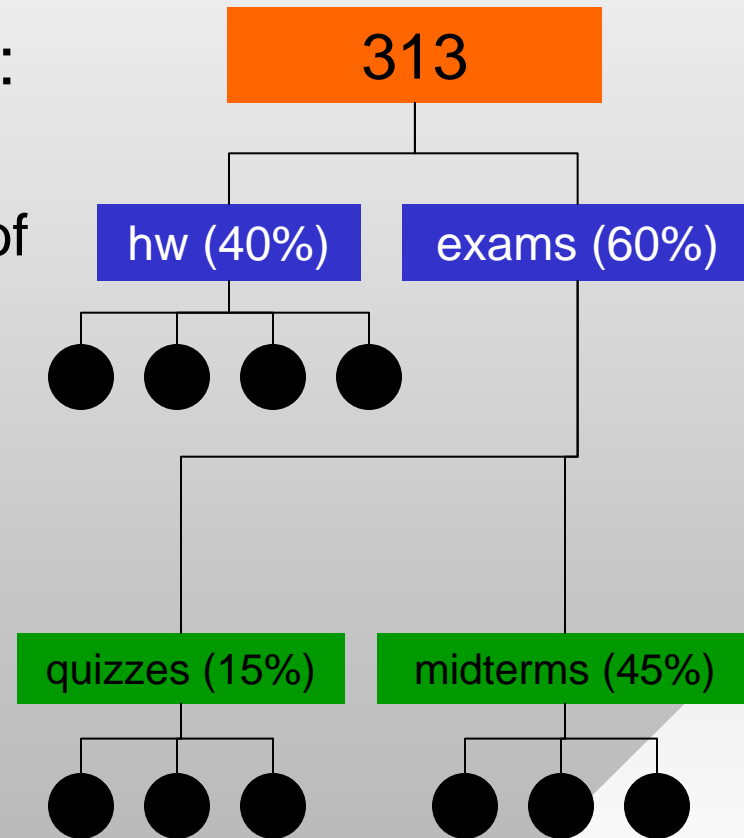
- Class based on Windows and Visual Studio
 - Must use Microsoft APIs
 - Obtain VS 2017 for free from Dreamspark (see <https://engineering.tamu.edu/cse/cse-internal/microsoft-dreamspark-for-academic-institutions>)
 - Or download the Community Edition from Microsoft
- Website: <http://irl.cse.tamu.edu/courses/313>
 - Slides and future test dates
 - Homework assignments
 - Links to useful material
- Piazza: piazza.com/tamu/spring2018/csce313200

Preliminaries: Syllabus 3

- Prerequisites (mandatory)
 - CSCE 312: Computer Organization
 - CSCE 222: Discrete Structures for Computing
 - CSCE 221: Data Structures and Algorithms
 - Queues, sets, hash tables, BFS/DFS search in graphs, heaps
 - CSCE 121: C++
 - Ability to write and debug code
- Helpful background
 - CSCE 311 (analysis of algorithms), C++ STL (standard template library), Visual Studio, Microsoft APIs
 - C++ pointers

Preliminaries: Syllabus 4

- Homework (40% of final grade):
 - 4 programming assignments
 - Each explores a different aspect of computer systems
- Exams (60% of final grade):
 - **Closed-book, no cheat-sheets**
 - 3 quizzes (15% of final grade):
 - C++ coding and synchronization problems, pointers, review questions and problems from the book
 - 3 midterms (45% of final grade):
 - Cover lecture/homework topics



Preliminaries: Syllabus 5

- Grade distribution
 - 90-100% (A), 80-89% (B), 70-79% (C), 60-69% (D), 0-59% (F)
- Do not hesitate to ask for help
 - Homework may be time-consuming if you're stuck on basic things (compilation, threading, deadlocks, APIs)
 - Multi-threaded programs are generally hard to debug, but they don't have to be if the code is designed/written well
- Where to ask questions
 - Office hours (bring a laptop), during class, through Piazza (general concepts), and email (code-specific)
- If you're totally stuck:
 - Send me your code with a **clear description of the problem, where it occurs, and what you have done to debug it**

Preliminaries: Syllabus 6

- Best route
 - Realize that others might have experienced similar problems (e.g., stackoverflow has tons of useful answers)
 - Perform initial investigation, obtain insight into the issue
 - Google the problem first
- If issue solved, answer your own question!
 - Help others on piazza
- Read online tutorials about VS, its debugger, precompiled headers, and watch videos
 - You will be quizzed on such things as the call stack, breakpoints, stepping thru code, immediate & watch windows

Preliminaries: Agenda

- Big picture and roadmap
- Syllabus
- **Academic integrity**
- Homework expectations
- VS 2017 demo
- Wrap-up

Preliminaries: Academic Integrity

- No teamwork is allowed
 - General discussion is acceptable, but **no part** of an assignment or report may be copied from another student
- Academic Rules, Section 20
 - All sources must be properly acknowledged (including MSDN and sample code provided in the handout or class website)
 - To receive credit, work must be **original and yours**
 - Expected that you can explain how you code works and what rationale led you to a particular design
- **All** parties involved in cheating will be punished equally
 - **Any occurrence: F* in class or expulsion from university**

Preliminaries: Agenda

- Big picture and roadmap
- Syllabus
- Academic integrity
- **Homework expectations**
- VS 2017 demo
- Wrap-up

Preliminaries: HW Expectations

- Homework:
 - Due at **noon**, 20% penalty per day (no points after 5 days)
 - Delays for personal reasons must be requested **in advance**
- Conform to the statement of the problem
- Must use VS 2017 + Windows 10.0.16299 SDK
 - Profile your code for bottlenecks, optimize whenever practical
 - If the CPU supports it, compile in 64-bit mode
 - Read MSDN for description of Windows APIs
 - Comment your code, use descriptive variable names, split functionality into independent classes/modules/functions
 - Avoid long loops (over a page), which are hard to debug

Preliminaries: HW Expectations 2

- Provide a **detailed** written report
 - **If multiple parts to hw, only the last one needs a report**
 - Explain what your code does and how it accomplishes the required functionality
 - Describe tricky or interesting parts of your implementation
 - Provide analysis of your results
 - Answer questions posed in the problem statement
- Sample runs
 - Capture screenshots or print into a file what your code does on test input data
- Goal: demonstrate in your report that you **understood** the material

Preliminaries: HW Expectations 3

- **Hard copy** submission:
 - Both report and code must be submitted before class starts
- **Soft copy** submission:
 - Add a comment to the top of each cpp/h file with your full name, class, and semester (i.e., CSCE 313-200 Spring 2018)
 - Create a zip with the files needed to compile the project (i.e., *.sln, *.cpp, *.h, *.vc*proj*), **delete everything else**
 - *Preserve the original directory structure inside the zip*
 - Upload to `csnet.cse.tamu.edu`
- Department servers for this class
 - `ts.cse.tamu.edu` and `ts2.cse.tamu.edu`
 - Use Windows Remote Desktop client to login
 - Expect the servers to get slow near hw deadlines

Preliminaries: Agenda

- Big picture and roadmap
- Syllabus
- Academic integrity
- Homework expectations
- **VS 2017 demo**
- Wrap-up

Wrap-up

- Homework #1 is due in 3 parts
 - Part 1 (1/25/18 **next Thursday!**): connect to 1 flybot, obtain its room, disconnect cleanly (25%)
 - Part 2 (2/8/18): single-threaded search (25%)
 - Part 3 (2/22/18): full multi-threaded version + report (50%)
- **Before next class**
 - Read hw1p1, study my systems programming tutorial, and think of questions to ask
 - Experiment with VS 2017