CS 361 Computer Systems II

James Madison University, Spring 2025

Concurrent systems–software that performs multiple operations at the same time–are everywhere. They allow Visa to process billions of transactions per year accurately, help BitTorrent distribute files efficiently world-wide, and facilitate modeling climate phenomena. At the same time, their failures have destroyed equipment ($100,000,000 for the NASA Mars Polar Lander), left millions of New Yorkers without power for two weeks (2003 blackout), and played a role in killing people (Toyota and Boeing crashes).

The problems arise when independent pieces of code share data or communicate in other ways. These interactions lead to behaviors that often cannot be predicted. There are principles and design patterns that we can apply to make concurrent systems safer and more reliable, though it's important to understand the limits of these techniques.

In this course, we will explore the models used to build and analyze complex systems software. We will use multiple techniques for exchanging information between code running on a single machine or across a network. We will also explore techniques used to prevent data corruption and discuss the drawbacks of these approaches. We will end with an introduction to foundational concepts in parallel and distributed computing.

Course & Instructor Information

Website:	`https://w3.cs.jmu.edu/kirkpams/361/`		
Time/Place:	M/W/F 9:10 – 10:00 AM (Section 1)		
	M/W/F 10:20 – 11:10 AM (Section 2)		
Textbooks:	*Computer Systems Fundamentals*		
Instructor:	Prof. Michael S. Kirkpatrick	Email:	kirkpams@jmu.edu
Office:	King 223	Phone:	(540) 568-3371
Office Hours:	M 2:45 – 4:30 PM, Th 2:00 – 4:00 PM, F 11:15 AM – 12:30 PM		

Course Structure and Grading Specifications

Each week begins with a pre-class reading assignment and accompanying Canvas reading quiz. Each class meeting will begin with a warm-up question, then proceed to a mix of mini-lectures, collaborative problem-solving, and multiple-choice clicker questions (ConcepTests). There will be 8 labs (small programming assignments), 3 projects (large, complex programming assignments), 4 module quizzes, a midterm, and a final.

This courses uses a specifications-based grading approach that may differ from a typical numeric scale that you may be used to. Each grading component is evaluated only as a letter grade with no plus/minus adjustments and no rounding. (E.g., an average of 84.7% on exams and module quizzes will be evaluated as a B.) Course letter grades are based on earning that letter grade in all four components. The following table describes each component's grade level requirements.

A requirements	
Exams:	Average of 85% on exams and module quizzes
Projects:	Earn 8 out of 9 project points
Labs:	Complete at least 7 of the 8 labs
Others:	Grade of 90% or higher on participation and reading quizzes
B requirements	
Exams:	Average of 75% on exams and module quizzes
Projects:	6 points total – at most one 0.5 score and no scores of 0
Labs:	Complete at least 6 labs, with at least 2 from the last 3
Others:	Grade of 80% or higher on participation and reading quizzes
C requirements	
Exams:	Average of 60% or higher on exams and module quizzes
Projects:	3 points total – at most one score of 0.5 or less
Labs:	Complete at least 4 labs, with at least 2 from the last 4
Others:	Grade of 70% or higher on participation and reading quizzes

Many students fall between these requirements, such as satisfying the A requirements for labs and projects but getting a B on the exams. Split grades will be determined as follows:

- If all are a mix of As and Bs:
  - If the exam and project grades are both As, the course grade will be A-.
  - If two of the four are an A and that includes either the exams or projects, the course grade will be B+.
  - Otherwise the course grade will be B.
- If all are a mix of As, Bs, and Cs:
  - If three of the four are As or Bs, or if both the exams and projects are As or Bs, the course grade will be B-.
  - If two of the four are a As or Bs and that includes either the exams or the projects, the course grade will be C+.
  - Otherwise the course grade will be C.
- If any component falls below the C requirements, course grades will use a traditional 4.0 scale with exams and projects weighted at 35% each and labs and participation at 15% each.

The following scenarios illustrate possible grades:

	Student 1:	Student 2:	Student 3:	Student 4:	Student 5:
Exams & quizzes:	A [91%]	A [86%]	B [78%]	C [61%]	B [75%]
Projects:	A [3, 3, 2]	B [1, 2, 3]	B [1, 3, 2]	B [2, 2, 2]	F [1, 1, 0]
Labs:	B [6]	B [6]	C [4]	C [4]	F [2]
Participation & reading:	A [95%]	A [90%]	A [90%]	B [85%]	C [65%]
Course grade:	A-	B+	B-	C+	D+

The following policies describe each component in more detail:

- **Weekly readings and quizzes** - Reading quizzes are due by Sunday at 5:00 PM. Before that time, you may retake these quizzes as many times as needed. Your lowest two quiz scores will be automatically dropped.
- **Participation** - Participation points will be earned through warm-up questions and ConcepTests in class, and Canvas metacognitive reflections, which are due each week by Thursday at 3:00 PM. ~~Late~~ Make-up submissions are not accepted.
- **Programming labs** - These labs give you practice with the skills required to complete the projects. Each lab will be due by Monday at 11:59 PM, with an opportunity to ask questions or to discuss clarifications the Friday before. All submissions will be through Git and must run on `stu.cs.jmu.edu`. Each lab will be graded on a pass/fail basis. All tests must pass in order to get credit. Late submissions will be accepted only up to one week after the due date.
- **Projects** - Both projects will be done individually or in pairs, will be submitted using Git, must run on `stu.cs.jmu.edu`, and must adhere to required coding standards. Each project has multiple phases of implementation that must be completed sequentially (e.g., all C tests must pass before beginning B requirements). Each project has up to 3 points available (A = 3, B = 2, C = 1, D = 0.5); earning the point for a grade level requires passing all of the corresponding test in the distribution. Project specifications may include other penalties that may apply.
- **Module quizzes** - Each module quiz (Friday of weeks 3, 6, 11, and 14) will have a 20-minute limit, though they should take less time than that. For the first three quizzes, you will have the opportunity to correct certain missed questions for additional credit. All module quizzes combined are worth the same amount as the midterm exam.
- **Midterm and final exam** - Both exams will include a written portion with questions and problems focused on general course concepts, and this portion will take place during the designated class time. Both exams will also include a portion focused on writing code. The final exam will be worth twice as much as the midterm.

Adjustments or extensions will be granted based on extraordinary circumstances at the instructor's discretion. If you are sick, please do not come to class; I assure you that the missed participation points will not impact your grade.

Course & University Policies

- **Classroom inclusion** - Learning environments should be built on mutual respect and support a diversity of thoughts, perspectives, experiences, and identities. Please advise me regarding any concerns or personal circumstances (including your name's proper pronunciation, any name or gender pronouns not reflected on MyMadison, or significant extracurricular commitments) that may be relevant to your full participation in class.

  As computing professionals, we adhere to the ACM Code of Ethics and Professional Conduct (`https://www.acm.org/code-of-ethics`), which forbids discrimination and harassment of all types. If you feel someone is violating these principles (including inappropriate or demeaning jokes), it is your responsibility to take action by informing me or (if you feel comfortable doing so) addressing the individual directly. I will do my best to preserve your confidentiality while addressing the issue.

- **Laptop policy** - This course is structured to use class time for discussions and other in-class activities. You may use a laptop for on-task use only. **Do not work on projects or other assignments, for this class or others, during class time.** If your laptop use becomes a distraction to your peers or to me, you will no longer be able to use it in class.

- **Attendance and grading** - Attendance is not recorded directly, but is necessary for participation credit. Please let me know if you have any specific obligations (such as athletic team commitments or military service) that may constrain your attendance.

- **Communication policy** - Communication outside of class will be primarily through office hours and Piazza (available through Canvas). My email should only be used for documenting things, such as absences or regrade requests. (I will direct all non-private course-related questions to Piazza.) I have designated times each day for responding to course communications and I am generally unavailable outside these times.

- **Academic integrity** - Students are expected to comply with the JMU Honor Code as stated in the Student Handbook and available from the Honor Council Web site at `http://www.jmu.edu/honor/code.shtml`. The Honor Code states it is a violation to "render *unauthorized* assistance to another student by knowingly permitting him or her to see or copy all or a portion of an examination or any work to be submitted for academic credit." In the context of this course, this portion of the Honor Code means:
  - You are allowed to collaborate on labs, provided that you contributed to any code that you submit; all such collaborations must be clearly identified in comments on submitted code. You are allowed to work **in pairs** on projects; no collaboration outside of pairs is allowed. Except for these two exceptions, **copying and/or sharing code is expressly forbidden**. This includes copying code from previous semesters' solutions (including your own or other students') or other external sources. It also includes posting your code publicly, such as on Github.
  - It is acceptable to consult other resources (*e.g.,* Stack Overflow) for clarifying examples but not wholesale copying of significant (more than 5-10 lines) pieces of code. All such references must be documented explicitly within code comments, including any prompts used with generative AI tools. Copying and pasting any of the

project descriptions or source code into generative AI tools (e.g., Copilot, ChatGPT) is strictly prohibited.
  - Extensive discussions with other students that are likely to lead to similar code must be disclosed *before or during submission* (either in person or documented in code comments); unintentional violations will be granted leniency, though a penalty may still apply.
  - **Project source code will be run through a code similarity tool** as an input to detecting plagiarism and unacceptable copying as described above.

- **Adding/dropping classes** - You are responsible for registering for classes and for verifying your schedule on MyMadison. Deadlines for adding or dropping classes are available from the JMU Registrar.
- **Cancellations** - JMU's cancellation policy (`http://www.jmu.edu/JMUpolicy/1309.shtml`) provides details regarding inclement weather and other emergencies.
- **Religious observance accommodations** - All faculty are required to give reasonable and appropriate accommodations to students requesting them on grounds of religious observation. If you need to request accommodations, you must let me know at least 2 weeks in advance.
- **Disability accomodations** - JMU abides by Section 504 of the Rehabilitation Act of 1973 and the Americans with Disabilities Act, which mandate reasonable accommodations be provided for students with documented disabilities. If you have a disability and may require some type of instructional and/or examination accommodations, please contact me early in the semester so that I can provide or facilitate provision of accommodations you may need. If you have not already done so, you will need to register with the Office of Disability Services, the designated office on campus to provide services for students with disabilities. The office is located in Wilson Hall, Room 107 and you may call 540-568-6705 for more information.

Course Catalog Description
Intermediate exploration of modern interrupt-driven computer systems. Explores models of computation and complex systems, techniques for communication and synchronization of parallel and concurrent software, and the protocols that make up the Internet. *Prerequisites: Grades of 'C-' or better in CS 240 and CS 261.*

Detailed Course Objectives
Following the successful completion of this course, students will be able to:

- Write safe and robust systems code that uses concurrency and synchronization.
- Compare and contrast architectures that are commonly used in concurrent systems.
- Interpret and implement UML statechart and sequence diagrams of concurrent systems.
- Summarize the relationship between signals and the process life cycle.
- Distinguish the relative merits of various forms of IPC and when to use each.
- Compare and contrast processes and threads as concurrent execution mechanisms.
- Classify network protocols according to their layer and intended purpose.

- Select appropriate mechanisms for solving synchronization problems.
- Distinguish the notions of concurrency and parallelism.
- Summarize the key challenges and foundational results of distributed systems.
- Interpret and implement systems software correctly based on technical documentation.
- Accurately assess one's mastery of course material and identify opportunities for improvement.
- Make progress toward improving one's technical and interpersonal professional skill set.