CSCI-352 Operating Systems
Course Syllabus

Last updated 2025/08/17 17:16:02

Update history:

```
2025/08/17: initial version
```

Name: Warren R. Carithers
Office: 3617 Golisano (70-3617)
Phone: (585) 475-5393
Email: wrc AT cs.rit.edu
My current schedule

Contents

---

1. Course Overview

1.1. Course Goal

The goal of Operating Systems is to ensure that students gain a solid understanding of the fundamental concepts of resource management and the components of a modern multitasking operating system.

1.2. Course Outcomes

The intended outcomes of this course are that students will be able to do the following:

- describe and implement basic concepts, policies, and mechanisms used by operating systems;
- compare and implement process scheduling algorithms;
- apply synchronization primitives and evaluate deadlock conditions;
- analyze and implement virtual memory management algorithms.

1.3. Prerequisites

The prerequisite for CSCI-352 is:

- One of:
  - CSCI-250, Concepts of Computer Systems
  - CMPE-240, Engineering Fundamentals of Computer Systems
  - SWEN-340, Software Design for Computing Systems

or permission of the instructor. *Please see me as soon as possible if you haven't completed one of the prerequisite courses.* If you haven't taken one of these courses (or an equivalent), there is a very good chance that you don't have sufficient experience to allow you to succeed in this course.

---

2. Resources

2.1. Textbooks

The textbook for this course is:

- Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*. It is available free online as a set of PDF chapters, or can be purchased in hardcover or softcover form from Lulu or Amazon (see links on the OS:TEP web page, and be aware of resellers on Amazon).

There are a number of other operating system textbooks available; see the course *Resources* page for information about them.

You will be reading quite a bit of C code (and writing some of it it as well), so a good C reference may be very helpful. Similarly, you will also be doing some work with the x86 and MIPS architectures and their assembly languages, so having assembly references may be useful. Again, the *Resources* page has information about many of these.

2.2. Facilities

Your work in this course will be done on the computers in the Distributed Systems Lab (70-3610). These are systems running Debian® Linux® (version 12), and the environment has been set up to be very similar to what's installed on the systems in the main CS labs.

Currently, the DSL contains the following systems:

- Nine quad-core Core i5 systems
- One Intel Xeon E5-1603 system

All of these are currently running 64-bit Debian 12. The E5-1603 system (sherlock) should be used only when all the other systems are in use, as it acts as a DHCP and PXE server for the other systems; in particular, it should not be shut down, as this may prevent the other machines from booting.

The DSL has a small library of hardcopy technical documents and textbooks, located on the West wall of the lab. These are for use in the lab only, and should not be removed from the lab.

The DSL has no lab assistant; access is controlled via our card-swipe security system. To gain access, you must have your ID entered into the security system. Early in week 2 of the semester, I will send a list of names and RIT IDs of people in the course to Mark Stamer (the CS hardware technician, `mdsics AT rit.edu`) to have them added to the access list for the lab. Mark will add everyone with a five-digit PIN; once he has done this, he'll let me know, and I'll pass the PIN on to everyone in the class. (*Note*: if you had DSL access in a previous semester for a course, your old PIN will most probably no longer work, as it is GCCIS policy to disable card-swipe access for students at the end of each term.)

The DSL door is controlled by an RFID card reader unit located on the wall next to the door handle. The reader has a set of translucent buttons (one large button at the top, with a telephone-style keypad below that). These buttons will be in one of these states:

State	Interpretation
Blinking RED	The lab is **locked** and the security system is **armed**.
Blinking ORANGE	Someone is in the middle of unlocking or relocking the lab.
Solid GREEN	The lab is **unlocked**; to enter, just push on the door.
Solid BLUE	The security system is broken, or is otherwise disabled. The lab **cannot be unlocked or entered** at this time.

The locking mechanism for the door is in the strike in the door frame. Inside the strike assembly is a moveable “keeper” When the door is unlocked, pushing the door causes the bolt to rotate the keeper out of the way so the door can be opened. When the door is locked, the keeper can only be rotated in the “closing” direction (i.e., when the door closes, the bolt pushed the keeper out of the way, after which it moves back into place and cannot be rotated to open the door). The handle on the *hallway side* of the door never turns - pushing the door moves the bolt past the keeper when the door is unlocked. However, the handle on the *lab side* of the door can always be turned to retract the bolt, ensuring that is impossible to find yourself “locked into” the lab.

If the lab is unlocked (solid GREEN buttons) when you arrive to use it, just push on the door to open it and enter the lab.

If the lab is locked (blinking RED buttons) when you arrive to use it, here is the "unlock" sequence:

- Tap your RIT ID against the card reader. You should hear the card reader beep.
- Enter your five-digit PIN, and then press the '`#`' button. (As you enter your pin, the buttons may change from RED to ORANGE; this sometimes doesn't happen, but that doesn't seem to affect the process.) You should hear a “clunk” as the latch is unlocked, and the buttons on the reader should turn GREEN.

If at any point you make a mistake, you'll need to wait for ~30 seconds for the card reader to time out; this is indicated by another beep. When that happens, you can try again.

When you leave the lab, *if you are the last person leaving,* you must re-lock the door. The locking sequence is an extended version of the unlocking sequence:

- First, pull the door toward you to make sure the door latch is engaged. (It is possible to “lock” the door without having the latch engaged, which means you can open the door *even though it's “locked”*. Among other things, this trips the security system, and sends an alarm signal to Public Safety. *Please don't do this*, as this is considered a false alarm, and we are charged money when this happens.)
- Tap your RIT ID against the card reader. You should hear the card reader beep.
- Enter your five-digit PIN, and then press the `'#'` button. (Again, the buttons may or may not change to ORANGE.) You will hear another beep from the card reader.
- Enter this character sequence: '`*`', '`5`', '`#`'. You will hear a beep after the '`*`', and a "clunk" after the '`#`' as the striker mechanism locks. The buttons may blink GREEN (or ORANGE) for a few seconds, and then

will turn RED and begin blinking, indicating that the door is now locked and the security system has been re-armed.

You will use the same account name and password to access these systems that you use on the main CS lab systems and compute servers; however, there are some things that are different.

- Unlike the main CS systems, your account isn't automatically valid for use on the DSL systems - it must be given permission to use them. I will be sending the final class list to our system admins at the same time I send it to Mark Stamer (roughly the middle of week 2). At that point, the admins will enable access for your account.

- Your home directory on the DSL network is completely separate from your main CS account's home directory. The first time you log into a DSL system, your home directory will be populated with various things (e.g., desktop environment configuration files). You can copy things like shell configuration files from your main-net CS account into your DSL account once this happens.

- Finally, the directory layout of the DSL filesystem is similar to that of the main-net CS systems, but a few things may be in different places (e.g., your home directory may have a different path on the DSL systems than it has on the main-net systems). If you make changes to your shell configuration files, please be sure to verify them to ensure things work the way you expect them to.

We also have a set of "team" accounts available for use in this course; these can serve as a central repository for holding all the materials your team members develop over the semester. These will be created, if needed, once teams have been formed. You are free to use or not use your "team" account as you prefer.

For security reasons, the DSL machines are on a separate network. The DSL server (named `mycroft` from inside the lab, but named `dsl` from the outside world) is on both networks, so you can get to your DSL account's home directory from either side.

The DSL server machine is primarily a file server, and *does not have the same set of software tools* that the lab machines have - e.g., it has no compiler - so you cannot use it to do your work. You can use SSH to log into the DSL server from the outside world, and from there you can use SSH to get to any of the lab machines. (You can also use an SSH keypair to access the server and lab machines if you have one.)

You can copy files from machines outside the lab to your DSL home directory using standard SSH_based commands such as

```
sftp dsl.cs.rit.edu
scp file(s) acct@dsl.cs.rit.edu
```

(from within a shell window) or an ftp application. (Note: depending on the level of "probing" the rest of the world does on the DSL server, we sometimes must reconfigure the server to reject connections from systems other than those in the CS domain in order to reduce the chance that the server is attacked and compromised. Should this happen, you may not be able to SSH to the server from home.)

From inside the lab, network access to the outside world is more limited. The DSL server runs a web proxy server, so you have access to the web from inside the lab, but only HTTP and HTTPS requests are forwarded. No other access from inside the lab to the outside world is available. All the DSL systems are configured to use the proxy server by default, so accessing the web *should* be completely tranparent regardless of which browser (Firefox, Chromium, lynx, links, and links2 are currently installed) you use; however, some apps (notably Firefox) sometimes ignore the system-wide settings. If you find this to be the case, you'll need to manually configure your browser to use port `80` on the server, which has IP address `10.10.10.1` and is named `mycroft`.

It should be possible to do your software development elsewhere if that is necessary. You will need a C compiler (such as GCC) that understands the "ILP32" data type model (integers, long integers, and pointers

are all 32-bit), and produces code that follows the System V ABI specification for Intel386 architectures, and an assembler that understands "AT&T" syntax for x86 assembly languages (such as the GNU assembler), as that is what all the prewritten assembly code we will be examining and using follows.

It may be necessary to install additional compiler and/or library modules to achieve this; for example, on Ubuntu systems, you will need to install the gcc-multilib package to get 32-bit compiler support, and the libc6-dev-i386 package if you want to compile code to run on a 64-bit system but have that code follow the ILP32 model.

2.3. Hardware references

In order to communication with any type of device, you must understand the structure of the control and data registers used to send and receive commands, status information, and data to and from the device. This information is found in device-specific hardware reference guides. Many of them are collected in an online hardware documentation archive I have amassed over time, but you may find it useful (or necessary) to look elsewhere for such documentation.

2.4. Other resources

Finally, see the course resource pages for this course and the *Systems Programming* (CSCI-452) course for links to numerous online sources of information.

---

3. Grades

Your final grade will be based on several deliverables, described in more detail in the following sections of this document.

Deliverables will be weighted as follows:

Component	Weight
Programming Assignments	35%
Quizzes	15%
Midterm Exam	25%
Final Exam	25%

3.1. Programming Assignments

I anticipate assigning four programming projects during the term. Project assignments will be announced in class and posted to the course web site along with their due dates. All programming projects will be weighted equally.

Some of these programming assignments are complex enough that completing them individually would be very difficult at best, so you will be working in small teams. Teams will consist of two or three people. In the event that we have an "odd person out" situation (i.e., one "extra" person in the course), I may approve a team of size four, but that is a special case rather than a general option. (This also applies to the situation where one member of a team withdraws from the course before completing it, leaving the other member(s) on their own.)

Team formation is fairly informal; once a team is formed, one person (the "team leader") should send me email to let me know that these people are working together. The message should include the names and email addresses of *all* members of the team, and should be sent to all team members as well. I will acknowledge

receipt of these selections via return email as soon as I receive them; if you do not receive this reply within a couple of days of your email, please get in touch with me directly to ensure that your request reached me.

3.2. Quizzes

In order to ensure that people are keeping up with the course material, I will give a short (~20 minutes) online quiz at the end of each odd-numbered week beginning in week 3. These will cover recent course material, and will typically consists of three to six questions; they are designed to supplement the midterm exams' coverage of the course material. There will be no make-up quizzes, but the lowest of these six quiz scores will be dropped automatically, so missing a quiz will not affect your grade.

3.3. Exams

Two exams will be scheduled for this course: one at the midpoint of the semester (week 8), and a comprehensive exam during finals week. These exams will be open-book and open-notes online exams.

The midterm exam will be given over a 48-hour time period. You can begin the exam at any time during that period; once you begin, you will have a 90-minute (1.5 hours) time limit in which to complete the exam. The tentative date for this exam is the Friday and Saturday of week 8, October 17-19 .

The final exam will be similar, but will be available over a longer time period and will have a longer time limit for completion. The due date for this exam will be determined by RIT; currently, it is December 10-14 .

---

4. RIT and CS Department Policies

The CS Department Selected Policies document has summaries of a number of department and Institute policies that you should read through. These include policies regarding regarding leaves of absence, incidents of academic dishonesty, course withdrawal, degree audits, and many other important things you should be aware of. (Some of these are described below, but not all.)

4.1. Wellness

Success in this course depends heavily on your personal health and well-being. Stress is an expected part of the college experience, and unexpected social and personal setbacks can compound this. I (along with all your other instructors) encourage you to recognize these challenges as an unavoidable path to success.

Understand that you need to take care of yourself and communicate problems before demands of assignments reach a peak. *Please feel free to reach out to me about any difficulty that may affect your performance as soon as it occurs and before it is unmanageable.* In addition to your academic advisor, I strongly encourage you to contact the multiple other support services on campus that stand ready to assist you.

4.2. Academic Conduct

Academic dishonesty will be dealt with in accordance with DCS and RIT policies.

- RIT's Honor Code (from the *RIT Student Policy Library*).
  A general statement that sets standards of behavior for all members of the RIT community.

- RIT's Academic Integrity Policy (from the *RIT Student Policy Library*).
  Defines the basic forms of academic dishonesty (cheating, duplicate submission, and plagiarism) and explains the official RIT policy regarding academic dishonesty.

Submitting individual work written by others or as an unsanctioned team is considered an act of academic dishonesty. In cases where a student is suspected of cheating or copying material, the instructor shall notify the students involved and act in accordance with RIT's Academic Integrity Policy.

The concept of "academic integrity" includes the understanding that students will take all reasonable steps to ensure that the code they write will not be accessible to other students unless that is explicitly allowed by the specifications for an assignment. You should store work only on private, personal computers or on private, password-protected, unsearchable on-line storage facilities. Hard copies, if created, must be kept secure.

Although students may discuss assignments with others, all individually submitted writings and code must be created independently by the student and not copied from others or other sources (e.g. web pages). This includes copying from public repository sites such as GitHub.

Work copied from GitHub or other, similar sources will be subject to prosecution for breach of academic integrity. You must not use a public repository such as GitHub. Be sure that any version control repository you use is private.

Similarly, LLMs to generate code or algorithms for assignment solutions is not allowed in this course. (Examples include, but are not limited to, GitHub Copilot, StarCoder, and Google Bard.) Use of an LLM to produce assignment solutions will be subject to prosecution for breach of academic integrity.

Team-developed work also must be created solely by team members and not copied from others or other sources without prior instructor approval.

4.3. Section 504 of the Rehabilitation Act

RIT is committed to providing reasonable accommodations to students with disabilities. If you would like to request accommodations such as special seating or testing modifications due to a disability, please contact the Disability Services Office. It is located in the Student Alumni Union, Room 1150; the Web site is `www.rit.edu/dso`. After you receive accommodation approval, it is imperative that you speak with me as soon as possible so that we can work out whatever arrangement is necessary.

4.4. Gender-Based Discrimination and Harassment

RIT is committed to providing a safe learning environment, free of harassment and discrimination as articulated in our university policies located on our governance website. RIT's policies require faculty to share information about incidents of gender based discrimination and harassment with RIT's Title IX coordinator or deputy coordinators, regardless whether the incidents are stated to them in person or shared by students as part of their coursework.

If you have a concern related to gender-based discrimination and/or harassment and prefer to have a confidential discussion, assistance is available from one of RIT's confidential resources on campus (listed below).

1. The Center for Women & Gender: Campus Center Room 1760; 585-475-7464; CARES (available 24 hours/7 days a week) Call or text 585-295-3533.
2. RIT Student Health Center: August Health Center, 1st floor; 585-475-2255.
3. RIT Counseling Center: August Health Center, 2nd floor, room 2100; 585-475-2261.
4. The Ombuds Office: Student Auxiliary Union/Room 1114; 585-475-7200 or 585-475-2876.
5. The Center for Religious Life: Schmitt Interfaith Center, room 1400; 585-475-2137.

6. NTID Counseling & Academic Advising Services: 2nd Floor Lynden B. Johnson; 585-475-6468 (v), 585-286-4070 (vp).

---

5. Other Course Policies

- ***Disclaimer:*** Normally, the number, type, and relative weights of assignments will not change from those specified in the syllabus and other course documents. However, I reserve the right to make changes to these or any other facet of the course, at my discretion, based upon the events of the term; if such a change must be made, you will be notified in class, via electronic mail, and on my web page for the course.

- ***Attendance and participation:*** Outside of the first week, I do not take attendance in this class. Students are, however, expected to attend and participate in class. Participation means being engaged in the course (asking and answering questions, not just sitting in the classroom).

  Please do not use cell phones, audio players, handheld gaming devices, or other similar electronics during class. If you must take a phone call, please leave the classroom and return when you have finished the call.

  Use of a laptop/notebook/netbook/tablet/etc. computer during class for taking notes is fine; other uses (web surfing, games, interacting via social media, etc.) are discourteous and can be distracting to the class, and are not allowed except in the context of the course (e.g., looking at a web site to research a question that has been brought up during class discussion).

- ***Withdrawing:*** During the add/drop period, you may drop this course and it will disappear from your transcript. After that time, you can only withdraw from the course; the course will appear on your transcript with a grade of `W`.

  Deadline to add/drop: **Tuesday, September 2, 2025**
  Deadline to withdraw: **Friday, November 7, 2025**

- ***Warning:*** This course requires a ***moderate*** to ***significant*** amount of out-of-class work. I suggest that you consider whether or not you should take any other courses that have similar or heavier workloads along with it.

- ***Coursework:*** Unless otherwise specified in the assignment, all work you submit for grading must be your own. Code or ideas (specific algorithms, optimizations, etc.) obtained from or inspired by other sources must be properly attributed.

- ***Due dates:*** The due date for an assignment is selected in order to provide adequate time to complete the assignment, while allowing sufficient remaining time in the term to complete the remaining assignments. Should it become necessary, based on the events of the term, I reserve the right to change due dates; this, in turn, may require modification of due dates for other assignments during the term, or (in some cases) elimination of some assignments.

- ***Documentation and Programming Style:*** I expect students to follow some reasonable form of programming style. I don't mandate a specific style; for the most part, it's more important that your code be neat, clear, and (above all) consistent. Here are the major things I'm looking for:

  - reasonable length functions
  - reasonable length lines of code
  - program file headers describing the contents of the file along with **your name**
  - function and class header documentation describing purpose, parameters, return values, etc.

- in-line documentation of complex sections of code
- clear and consistant indentation
- **Make sure your name is on every document and file you send to me!**

An example style standard for C from CSCI-243 is available in PDF format. A similar set of recommendations for assembly language programs from CSCI-250 is also available in PDF format.

- ***Project submission:*** Unless otherwise indicated in the assignment, all solutions are to be submitted electronically by 23:59:59 (11:59:59pm) on the specified due date. Any day of the week is a valid due date. Solutions submitted through any other method (e.g., sent via email, slipped under my office door, put in the bin on the wall outside my office, etc.) will be ignored.

  Although you may do your development elsewhere, all submissions must be made on the CS lab machines, and all code will be tested there. Code which "works on my machine at home" but doesn't work on the CS systems *doesn't work*.

  Your project submissions may be graded by a grader. If so, and you have complaints about your grade, first talk to the grader. If you are still convinced there is a problem and you are unable to convince the grader, come and see me.

- ***Grading results:*** For assignments submitted electronically, grading results (including the grade and any explanatory comments about the grade) for the assignment will be returned via email to the RIT account corresponding to the CS account that was used to submit the assignment.

  For assignments submitted in class (e.g., exams, quizzes, etc.), grading results will be returned in class. Please do not ask a friend to pick up your assignments; I will not return your graded work to anyone other than you except in extreme circumstances.

- ***Posting of grades:*** Grades for individual assignments will be posted on an irregular basis to RIT's online course management system (currently, MyCourses). Grades posted this way may not reflect the results of regrading or other adjustments.

- ***Regrading:*** Once a grade for an assignment has been returned to the class, you have one week in which to request a regrade or to bring any other questions about your grade to my attention. No regrading will be done after that time.

- ***Calculation of final grades:*** Final grade calculations are done by me using my own copies of all grade information, according to the grading scheme described above. In particular, final grade calculations are *not* done on the basis of values posted to MyCourses.

- ***Final Exam:*** RIT has an official set of *Final Examination Policies* which detail procedures related to the scheduling of final exams. Most important among these is the procedure to be followed by students who wish to request a change in date or time for an exam.

- ***Assignment of final grades:***

  I use the percentage-based grading scale shown at right in this course.

  I reserve the right to alter these division points as I see fit at the end of the term if I believe it to be necessary, based on my overall evaluation of individual or class performance and effort.

Score Range	Grade
$n \geq 93\%$	A
$90\% \leq n < 93\%$	A-
$87\% \leq n < 90\%$	B+
$83\% \leq n < 87\%$	B
$80\% \leq n < 83\%$	B-

$77\% \leq n < 80\%$	C+
$73\% \leq n < 77\%$	C
$70\% \leq n < 73\%$	C-
$60\% \leq n < 70\%$	D
$n < 60\%$	F

- ***Course evaluations:*** RIT uses SmartEvals, a web-based, Institute-wide system for student rating of teaching effectiveness. SmartEvals allows you to evaluate each of your instructors using a uniform platform and common look and feel experience across all of your courses. Access to each of your course rating forms will be open during the final few weeks of the semester; each form should take 10 or fewer minutes to complete. Information about this system will be shared with you via email and a dedicated website.

  I take your comments seriously and your responses allow me to reflect on my teaching and course delivery, I encourage you to participate in SmartEvals for this course so as to achieve as close to a 100% participation rate as possible. In keeping with Institute guidelines, I do not offer extra credit or points as incentives to complete this evaluation.

---

6. Acknowledgements

Portions of the material in this course may be based on material prepared by other faculty, and are used with their permission.
