CS271 Computer Architecture and Assembly Language

Instructor Information Joe Paris parisj@linnbenton.edu Office: MKH-116 Office Hours: M,W,F 1:00–2:00 or by appointment

Course Information Section Number: 01 CRN: 34268 Number of credits: 4 Scheduled time/days: MW 4:00–5:20 F 4:00–5:50 Classroom: MKH-101

Prerequisites

• CS 161 Introduction to Computer Science I with a grade of “C" or better.

Objectives The aim of this course is to help you become a better programmer by teaching you the basic concepts underlying all computer systems. We want you to learn what really happens when your programs run, so that when things go wrong (as they always do) you will have the intellectual tools to solve the problem.

Why do you need to understand computer systems if you do all of your programming in high level languages? In most of computer science, we’re pushed to make abstractions and stay within their frameworks. But any abstraction ignores effects that can become critical. As an analogy, Newtonian mechanics ignores relativistic effects. The Newtonian abstraction is completely appropriate for bodies moving at less than 0.1c, but higher speeds require working at a greater level of detail.

The following “realities” are some of the major areas where the abstractions you’ve learned in previous classes break down:

1. *Int’s are not integers, Float’s are not reals.* Our finite representations of numbers have

significant limitations, and because of these limitations we sometimes have to think in terms of bit-level representations. 2. *You’ve got to know assembly language.* Even if you never write programs in assembly,

The behavior of a program cannot be understood sometimes purely based on the abstraction of a high-level language. Further, understanding the effects of bugs requires familiarity with the machine-level model.

3. *Memory matters.* Computer memory is not unbounded. It must be allocated and

managed. Memory referencing errors are especially pernicious. An erroneous updating of one object can cause a change in some logically unrelated object. Also, the combination of caching and virtual memory provides the functionality of a uniform unbounded address space, but not the performance.

4. *There is more to performance than asymptotic complexity.* Constant factors also matter.

There are systematic ways to evaluate and improve program performance.

5. *Computers do more than execute instructions.* They also need to get data in and out and

they interact with other systems over networks.

By the end of the course, you will understand these “realities” in some detail. As a result, you will be prepared to take upper-level systems classes at university. Even more important, you will have learned skills and knowledge that will help you throughout your career.

In detail, we set forth the following learning objectives, as activities you should be able to do after completing the course:

1. Explain common bit-level representations of numeric values (unsigned, two’s

complement, floating point) and the consequent mathematical properties of arithmetic and bitlevel operations on them. 2. Recognize the relation between programs expressed in C and in assembly code,

including the implementation of expressions, control, procedures, and data structures.

3. Demonstrate ability to understand basic intention of a program through its binary

representation and apply these skills to debugging programs.

4. Investigate the programmer’s interaction with the underlying system through the different APIs and abstractions, including system support for process and thread control, virtual memory, and networking.

5. Analyze the consequences of imperfect system usage, such as poor memory and CPU

performance, crashes, and security vulnerabilities.

6. Apply tools, both standard and self-developed, that will aid program development,

including compilers, code analyzers, debuggers, consistency checkers, and profilers.

7. Apply these analytic and tool-use abilities to create reliable and efficient programs

exercising the different components of a modern computing system.

8. Understand the sources of conflict that can arise when multiple threads of execution

share resources, and demonstrate the ability to use synchronization constructs to mediate those conflicts.

Course Materials Required:

• Computer Systems: A Programmer’s Perspective, 3rd Edition by Bryant and O’Hallaron; ISBN 9780134092997 available from VitalSource.

• A text editor of your choice. I am partial to Visual Studio Code (https://code.visualstudio.com/).

• An account on a Linux-based computer.

• A GitHub account.

• Internet access.

Recommended:

• The C Programming Language, 2nd Edition by Kernighan and Ritchie; ISBN 978- 0131103627 available from Amazon.com.

A Note About the Text: Please make sure you have the Third Edition, which is significantly different from the Second Edition published in 2011. You can get either hardcopy or electronic editions. Electronic access, starting at $33.99, is available through Vital Source. There is currently not a copy on reserve in the library. Don’t buy the paperback version of the book! It’s ***not*** the same as the hardcover/electronic version. The practice and homework problems were rewritten and “are a total mess.”

In addition, you are expected to use reference material about the C programming language. The suggested reference is:

Brian W. Kernighan and Dennis M. Ritchie, The C Programming Language, Second Edition, Prentice Hall, 1988.

This is the classic *K & R* book, the standard against which all reference manuals are compared. This book should be in the library of anyone who programs in C.

Class Policies Behavior and Expectations You are held accountable to the Student Code of Conduct, which outlines expectations pertaining to academic honesty (including cheating and plagiarism), classroom conduct, and general conduct.

Attendance will not be taken. You will be considered responsible for all material presented during lectures.

Lectures will cover higher-level concepts and important “how-to’s”, especially in using tools that will help you do the labs.

The textbook contains both *practice problems* within the chapter text and *homework problems* at the end of each chapter. The intention is that you work on the practice problems as you are reading the book. The answers to these problems are at the end of each chapter. Our experience has been that trying out the concepts on simple examples helps make the ideas more concrete. Try out the practice problems associated with the readings for each class and ask questions about them at the next lecture. You will find that you will get much more out of the class if you have done some advance preparation.

The only graded assignments in this class will be a set of eight labs. Some of these are fairly short, requiring just one week, while others are more ambitious, requiring multiple weeks.

Guidelines for Communication The *best* way to reach your instructor is by email. I check my email regularly throughout the day and your message will receive a prompt reply. While they need not be strictly formal your emails should be concise, list necessary details, and written in a manner that would be appropriate for communicating with your boss.

Here is an example of a poorly written email:

And here is an example of a much more appropriate email:

The wikiHow page How to Email a Professor has many excellent tips on writing these kinds of emails and you are strongly encouraged to read it.

The *worst* way to reach your instructor is by phone. Please, don’t call me.

Students come from many different backgrounds and have unique life experiences which can enhance class discussions. For this reason, it is crucial for students to share their thoughts and insights on course-related topics. In this class we are all expected to show respect for each other at all times.

Use of Cell Phones While in class, please set your cell phone to vibrate. If you need to take a call, please step out of the room to do so.

Course Work Submitting Work All work, unless specifically stated otherwise, is to be submitted electronically. Details of how to do this will be covered in class.

Version Control We will be using Git and GitHub in this class. Assignments will be distributed and collected via GitHub Classroom. We will explain the proper usage of the program in class. In general, you should work as follows:

• Add all of the provided *source* files in your lab assignment upon downloading them and commit the initial version. Source files include any code (extensions ‘.c’, ‘.h’, ‘.pl’, ‘.py’, and ‘.sh’), as well as any Makefile and any program input file. It does not include any compiled libraries or reference programs.

• Commit early and often. Make it a habit to commit at least every hour you work actively on the assignment, and commit in small increments. Commit at the end of your work day.

• Make sure you commit your final version right before you submit for grading.

It is good software engineering practice to use version control, and learning it before starting Lab 1 is a good idea. We will be watching commit statistics on the server and may be reaching out to students who disregard our version control policy.

Handing in Assignments All written assignments must be word processed and both spell- and grammar-checked.

All assignments are to be submitted by 11:59 on the due date.

Assignments *may not* be submitted via email.

Late assignments *will not* be accepted. *No exceptions*.

As students, you are encouraged to discuss assignments and course materials with your peers. However, you are responsible for making sure you understand the work you turn in; i.e. you should be capable of explaining it verbally to the instructor if asked.

Work that appears to be directly copied from another student or is overly similar to that submitted by another student will not be given credit. We will be using a system called MOSS to check for plagiarism.

Exams The midterm exam will be given on Wednesday, February 5 from 4:00–5:20 in MKH-101. You will be allowed one, four-inch by six-inch notecard for notes, handwritten on one side of the card. You will be required to turn in this notecard along with your exam.

The final exam given will be given during week 11 of the term the date and time of which can be found here. You will be allowed one, four-inch by six-inch notecard for notes, handwritten on one side of the card. You will be required to turn in this notecard along with your exam.

There will be no early or late exams given without prior arrangement with your instructor. Any such arrangements are at your instructor’s discretion.

Quizzes In the middle of many lectures, we will pause a few minutes to have students answer a 2–3 multiple-choice question quiz on the material covered in class. This will help the instructor gauge which topics need further discussion. Student grades on these quizzes will be used solely as “bonus points,” as discussed next.

**Bonus points:** We will selectively consider raising individual grades for students just below the cutoffs based on factors such as attendance, class participation, improvement throughout the course, final exam performance, and special circumstances. In particular, the in-class quizzes provide a record of lecture attendance and attention, and hence are a good means for a student just below a cutoff to have his/her grade raised.

Grading

Type Weight Labs 50% Midterm Exam 20% Final Exam 30%

Letter Grade Percentage Performance A 90%–100% Excellent Work B 80%–89.9% Good Work C 70%–79.9% Average Work D 60%–69.9% Poor Work F 0%–59.9% Failing Work

Campus Resources Learning Center Tutors are generally available for this and other computer science classes. Check with the instructor and/or the Learning Center if you feel you need further assistance with this course.

Library Computers and printing available.

Basic Needs Any student who has difficulty affording groceries or accessing sufficient food to eat every day, or who lacks a safe and stable place to live, and believes this may affect their performance in the course, is urged to contact the Roadrunner Resource Center for support via email at resources@linnbenton.edu or visit us on the web at www.linnbenton.edu/RRC. Our office can help students get connected to resources to help. Furthermore, please notify the professor if you are comfortable in doing so. This will enable them to provide any resources that they may possess.

College Policies LBCC Email and Course Communications You are responsible for all communications sent via Moodle and to your LBCC email account. You are required to use your LBCC provided email account for all email communications at the College. You may access your LBCC student email account through student email and your Moodle account through Moodle.

Disability and Access Statement You should meet with your instructor during the first week of class if:

1. You have a documented disability and need accommodations. 2. Your instructor needs to know medical information about you. 3. You need special arrangements in the event of an emergency.

If you have documented your disability, remember that you must make your request for accommodations through the Center for Accessibility Resources (CFAR) Online Services webpage every term in order to receive accommodations. If you believe you may need accommodations but are not yet registered with CFAR, please visit the CFAR Website for steps on how to apply for services or call (541) 917-4789.

Important Board Policies and Administrative Rules

• BP7030 Student Conduct and Discipline

• AR7030-01 Student Rights, Responsibilities, and Conduct Code

• AR7030-02 Academic Integrity and Honesty

Statement of Inclusion To promote academic excellence and learning environments that encourage multiple perspectives and the free exchange of ideas, all courses at LBCC will provide students the opportunity to interact with values, opinions, and/or beliefs different than their own in safe, positive and nurturing learning environments. LBCC is committed to producing culturally literate individuals capable of interacting, collaborating and problem-solving in an ever-changing

community and diverse workforce.

Title IX Reporting Policy If you or another student are the victim of any form of sexual misconduct (including dating/domestic violence, stalking, sexual harassment), or any form of gender discrimination, LBCC can assist you. You can report a violation of our sexual misconduct policy directly to our Title IX Coordinator. You may also report the issue to a faculty member, who is required to notify the Coordinator, or you may make an appointment to speak confidentially to our Advising and Career Center by calling 541-917-4780.

Public Safety/Campus Security/Emergency Resources Linn-Benton Community College Safety & Loss Prevention Office web page.

Public Safety is also responsible for lost and found.

In an emergency, call 911. Also, call LBCC Campus Security/Public Safety at 541-926-6855 and 541-917-4440.

From any LBCC phone, you may alternatively dial extension 411 or 4440. LBCC has a public safety app available for free. We encourage people to download it to their cell phones. Public Safety also is the home for LBCC's Lost & Found. They provide escorts for safety when needed. Visit them to learn more.

Changes to the Syllabus I reserve the right to change the contents of this syllabus due to unforeseen circumstances. You will be given notice of relevant changes in class, through a Moodle Announcement, or through LBCC email.