

CSE251: System Programming

1. Introduction

Seongil Wi

Who am I?

about:Seongil Wi



- Assistant professor
- Security researcher

- Office: E106, 301-8
- Office Hour: Wednesday, 2~3pm (by appointment)
 - 🏠 Homepage: <https://seongil-wi.github.io/>
 - ✉ Email: seongil.wi@unist.ac.kr



My Research



- UNIST CSE / WebSec Lab. (Web Security Lab)
 - 🏠 Homepage: <https://websec-lab.github.io/>
- Research keywords:
 - **Web and Software Security**
 - Client/Server-side Security
 - Web Vulnerability Discovery



- Finding **security bugs** in web components (applications, browsers, ...)
- Finding **emerging web threats**
- Analyzing online **criminal activities**
- Using...
 - Dynamic/static analysis
 - Clone detection
 - AI techniques
 - Etc.

Making *web ecosystems*
more *secure!*

This Course

System Programming

You will learn **how hardware + software combine to support the execution of application programs!**



Processors, Memories,
Disks, I/O devices...



Operating systems,
Compilers, Libraries, ...

You will learn **how hardware + software**
combine to support the execution of
application programs!

Web Browser,
PowerPoint, ...

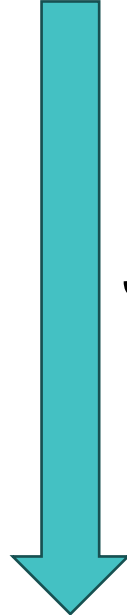




Processors, Memories,
Disks, I/O devices...



Operating systems,
Compilers, Libraries, ...



Support

Web Browser,
PowerPoint, ...





Processors, Memories,
Disks, I/O devices...

Hardware



Operating systems,
Compilers, Libraries, ...

System Software

In particular, you will learn how to use
these resources effectively as a
programmer

Why Do I Need to Know This Stuff?



- Abstraction is good, but don't forget *reality!*
 - Abstraction: removing elements that aren't relevant to a problem-solving task
- Most CSE courses emphasize abstraction
 - Abstract data types (e.g., LIST, Queue, Stack)
 - Asymptotic analysis
- These abstractions have limitation
 - Sometimes you need to understand the underlying implementation
 - Sometimes the abstract interfaces are not as flexible or performant as you need
 - Sometimes there are bugs

Useful Outcomes from Taking This Course ¹²

- Become more effective programmers
 - Able to find and eliminate bugs efficiently
 - Able to understand and tune for program performance
- Prepare for later “systems” classes in CSE
 - CSE261: Computer Architecture
 - CSE302: Building Customized Computers
 - CSE311: Operating Systems
 - CSE351: Computer Networks
 - CSE412: Parallel Computing
 - CSE467: Computer Security
 - ...

This course is one of the most fundamental courses in CSE.

System Programming Contents



- C programming and POSIX
- Machine-level Programming
- Linking
- System Level I/O
- Memory Hierarchy, Virtual Memory
- Cache
- Malloc
- Processes
- Exceptional Control Flow
- Concurrent Programming
- ...

Course Information



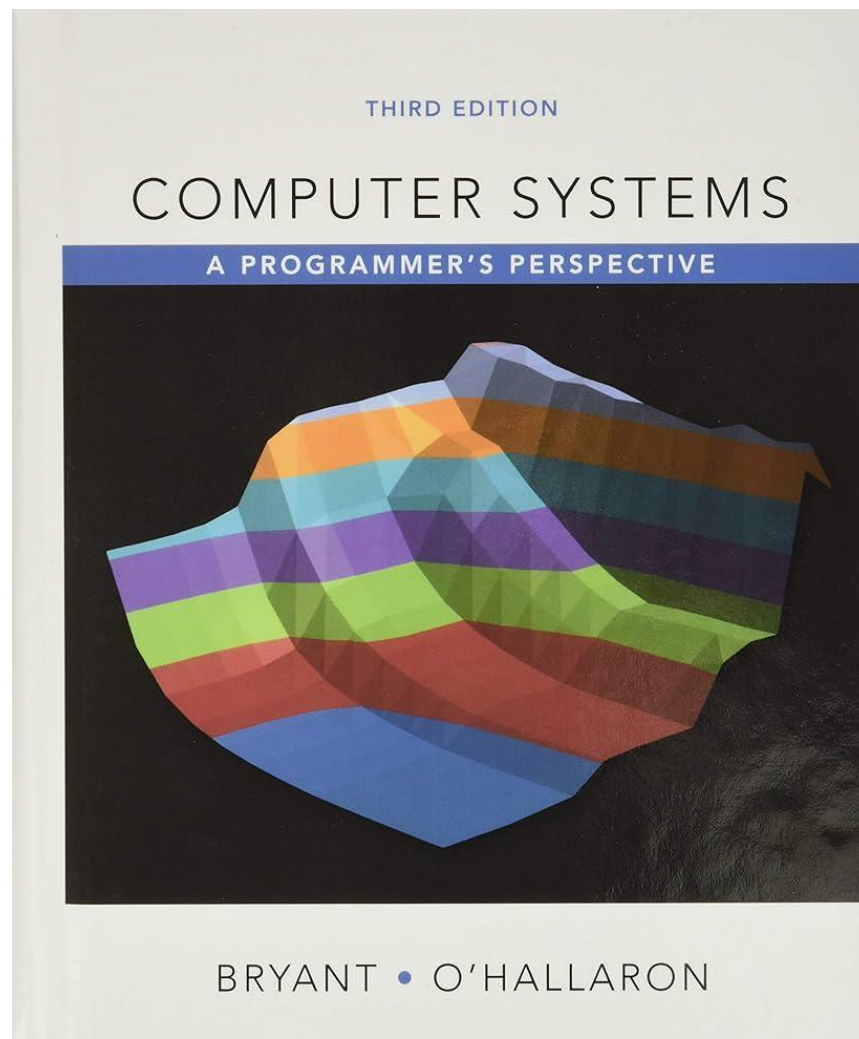
- **Course Website:**
 - <https://websec-lab.github.io/courses/2026s-cse251/>
- **Syllabus:** See the course website
- **TA:**
 - Jaeho Bae (배재호, bjho@unist.ac.kr)
 - Mingi Jung (정민기, wjdasrl4475@unist.ac.kr)
 - Jongyoung Kang (강종영, jongyoung_kang@unist.ac.kr)
 - Donggyu Kim (김동규, dgkim@unist.ac.kr)

Textbook



- **Computer Systems: A Programmer's Perspective, 3/E (CS:APP3e)**
 - Randal E. Bryant and David R. O'Hallaron
 - Pearson, 2015
- Lecture slides will be provided

Reading the textbook is highly recommended



Course Logistics*



- **Homework: 50%**
 - 5~6 Programming assignment
- **Final exam: 50%**

* Subject to change.

Attendance

Attendance is, of course, mandatory and enforce UNIST attendance rules

- **I will not include your attendance score in the grade**
 - **However**, I will drive the course in a way that rewards those who consistently participate with higher scores!
- **Also, missing more than 8 times will get an 'F'**
 - Your responsibility to check attendance online!
 - **If you attend and leave immediately (출퇴), there will be a grading penalty**
 - Show me evidence in case of an unavoidable absence, e.g., military training, illness, funeral
 - There is no excuse for absences due to your decision, e.g., interviews, competition participation

Attendance

Attendance is, of course, mandatory and enforce UNIST attendance rules

- **I will not include your attendance score in the grade**
 - **However**, I will drive the course in a way that rewards those who consistently participate with higher scores!
- **Also, missing more than 8 times will get an ‘F’**
 - Your responsibility to check attendance online!
 - **If you attend and leave immediately (출퇴), there will be a grading penalty**
 - Show me evidence in case of an unavoidable absence, e.g., military training, illness, funeral

I expect you to be here, as you expected me to be here!

Homework*



- **5~6 programming assignment**
- **Platform**
 - We will be learning about Linux on x86-64 (amd64)
 - We will work on Ubuntu 22.10
 - This is a very common and easily obtainable platform.
 - Your submissions are required to work on it!
- **A significant portion of your course grade will be programming assignments.**
 - These are individual assignments
 - All assignments must be written in C

* Subject to change.



Q. Is it okay to send an email at midnight?

– Of course! No one cares

Q. Is it ok to use ChatGPT for the programming assignment?

– Your sub-goal: Learn how to use AI ethically & constructively

Q. What happens if I submit AI-generated code?

– If unlucky, detected by our plagiarism checker. You will get **F**

– If lucky, you get a high score. But will be naturally selected soon

Important Notice (1): Academic Integrity ²¹

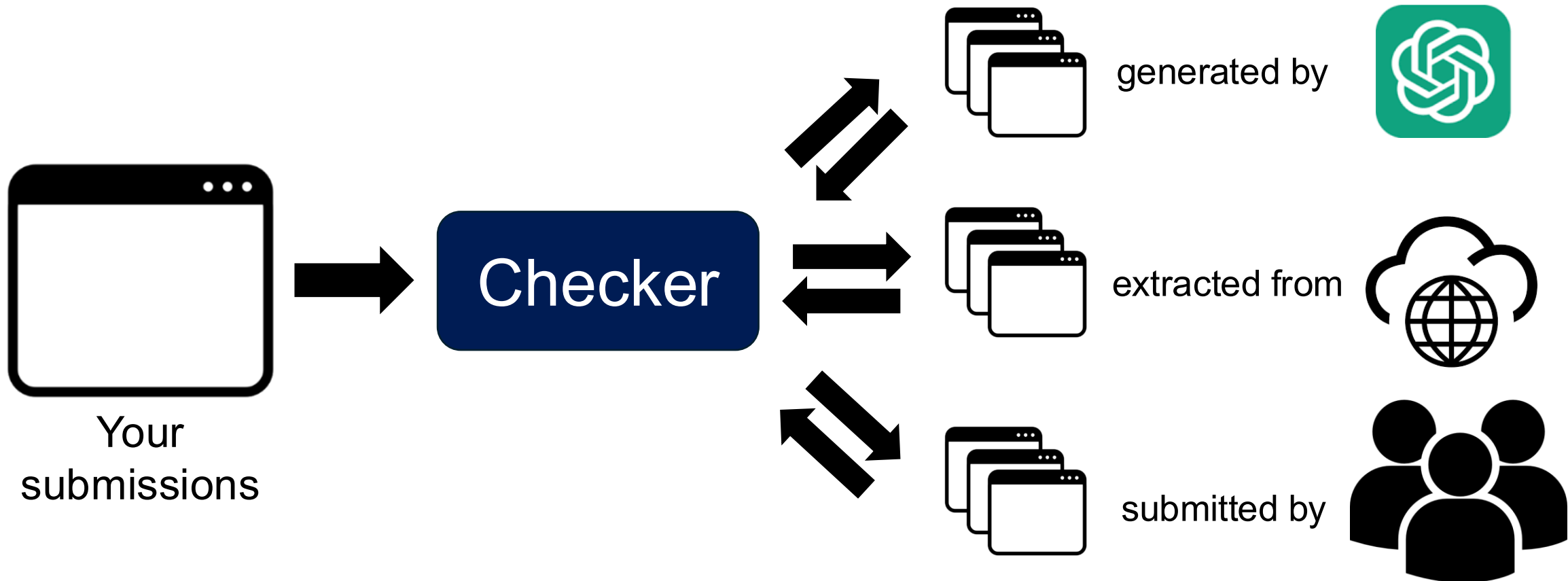


- Any solution you submit (hw, exam, etc.) must be your ***own*** work

If you violate this rule,
you will immediately get an **F**

Plagiarism Checker

- We use automated tools to strictly identify violations
- Even collaborating on several lines of code is subject to this



Important Notice (1): Academic Integrity 23

- DO NOT share the course contents (e.g., assignments or exams) with others
 - E.g., Github public repository, chegg.com, etc
- DO NOT discuss the **details of solutions** with others
- DO NOT plagiarize
 - Submit your own work
- Any integrity violation: at **LEAST F**

UNIST CSE Policy on Cheating and Plagiarism

Note: The term **solution** means program code, mathematical derivation, experimental setup, etc., for any type of deliverable, homework assignment, or projects in class.

The purpose of this document is to make our expectations in CSE as clear as possible in regard to the Honor Code at UNIST. The basic principle under which we operate is that each of you is expected to **submit your own work in your courses**. In particular, attempting to take credit for someone else's work by turning it in as your own constitutes plagiarism, which is a serious violation of fundamental academic standards. However, you are also encouraged to work as a team and collaborate with each other, and it is usually appropriate to ask others—the TA, the instructor, or other students—for direction and debugging help or to talk generally about

Criteria of Cheating



- Will rely on automated cheating detection tool
- Students are responsible for disputing the tool's report.
- How?
 - Use version control system (e.g., git) to log. i.e., create a local repository and commit your changes there. You can submit the git history.
 - If working on a Linux server, you can submit your command history also.
 - Your IDE may allow you to create the log of your edits and commands.

Important Notice (2): Class

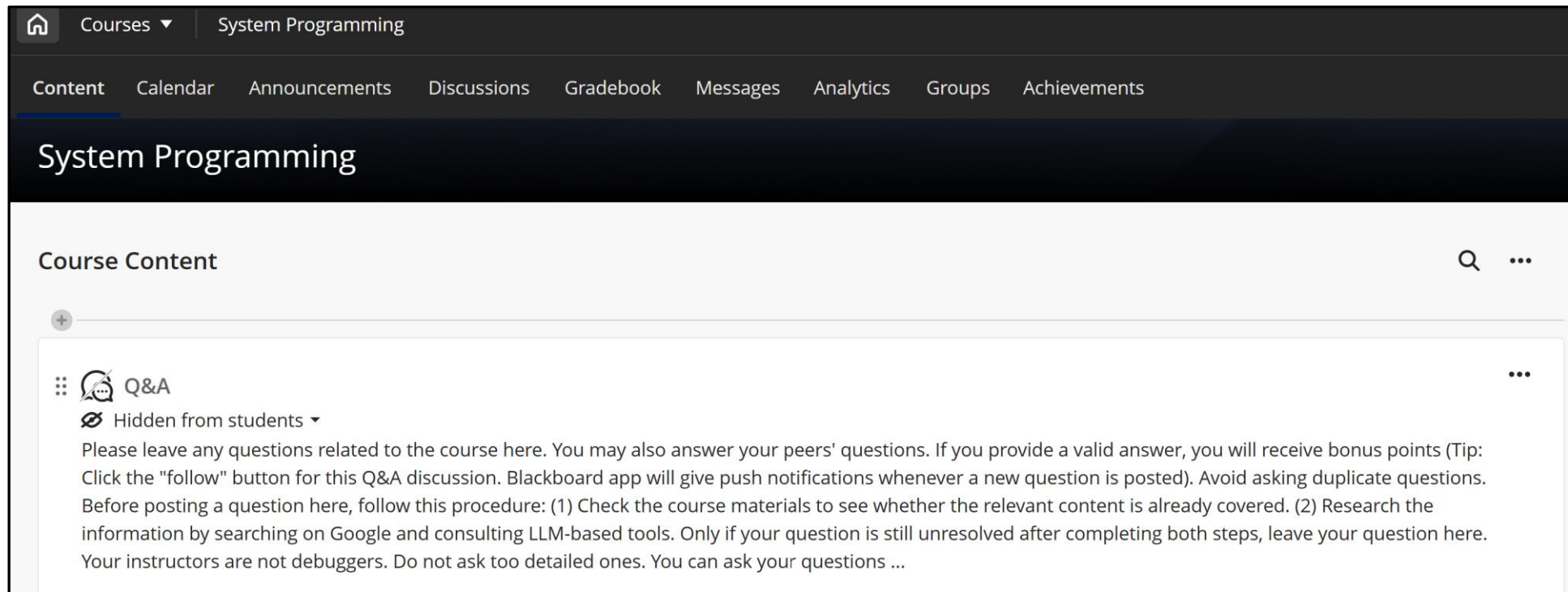


- **Language:** English (default)
 - You can use Korean for your questions, discussions, exams, and homework
- **Questions & discussion:** highly encouraged
 - (Out-of-class) If you have questions: Check the course materials → Searching on Google/LLMs → BlackBoard → TA → Prof. Wi
 - Except for
 - Too detailed ones (e.g., installation method, syntax of the program, compilation errors, ...). TA is not a debugger!
 - Directly related to the solutions
- **Actively discuss with your classmates**

Blackboard Discussion and Participation

26

- We will use Blackboard discussions for Q&A
- If you answer other students' questions, and if the answer is valid one, you will receive bonus point!
 - Only the first student who posts a valid answer will receive bonus point. I will reply to indicate who receives the participation point.



The screenshot shows the Blackboard interface for a course titled "System Programming". The navigation bar includes "Content", "Calendar", "Announcements", "Discussions", "Gradebook", "Messages", "Analytics", "Groups", and "Achievements". The "Content" section is expanded to show "Course Content". A search icon and a menu icon are visible in the top right of the content area. A "Q&A" discussion is listed, marked as "Hidden from students". The discussion text reads: "Please leave any questions related to the course here. You may also answer your peers' questions. If you provide a valid answer, you will receive bonus points (Tip: Click the 'follow' button for this Q&A discussion. Blackboard app will give push notifications whenever a new question is posted). Avoid asking duplicate questions. Before posting a question here, follow this procedure: (1) Check the course materials to see whether the relevant content is already covered. (2) Research the information by searching on Google and consulting LLM-based tools. Only if your question is still unresolved after completing both steps, leave your question here. Your instructors are not debuggers. Do not ask too detailed ones. You can ask your questions ..."

Final Remark



- Do not judge by yourself!
 - If you are confused, ask your TAs or me

Question?

Today, everyone will be acknowledged for attendance!