

Advanced Programming in the UNIX Environment

**Department of Computer Science
Stevens Institute of Technology**

Jan Schaumann

`jschauma@stevens.edu`

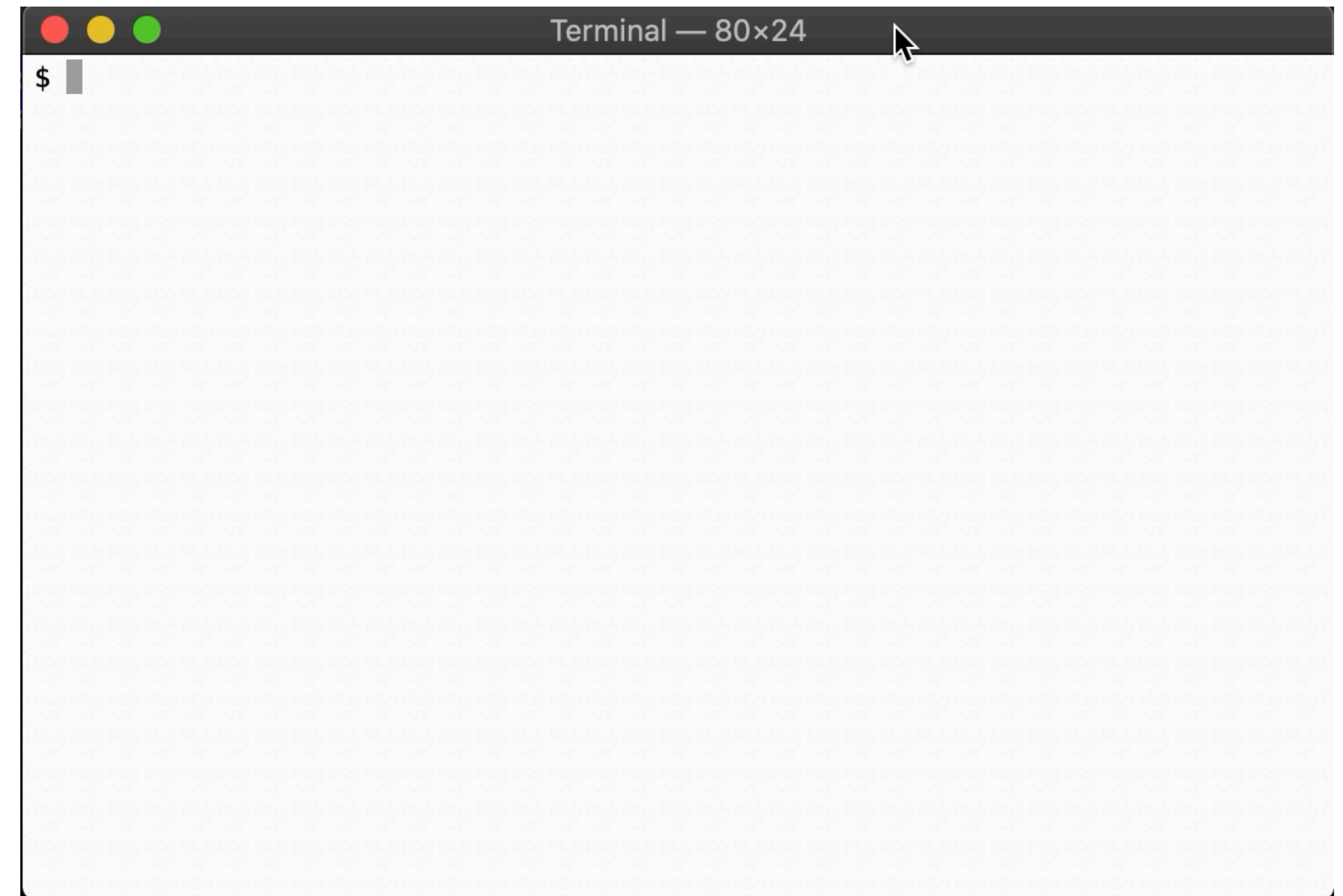
`https://stevens.netmeister.org/631/`

About this class

This class is called "*Advanced Programming in the UNIX Environment*".

It is *not* called:

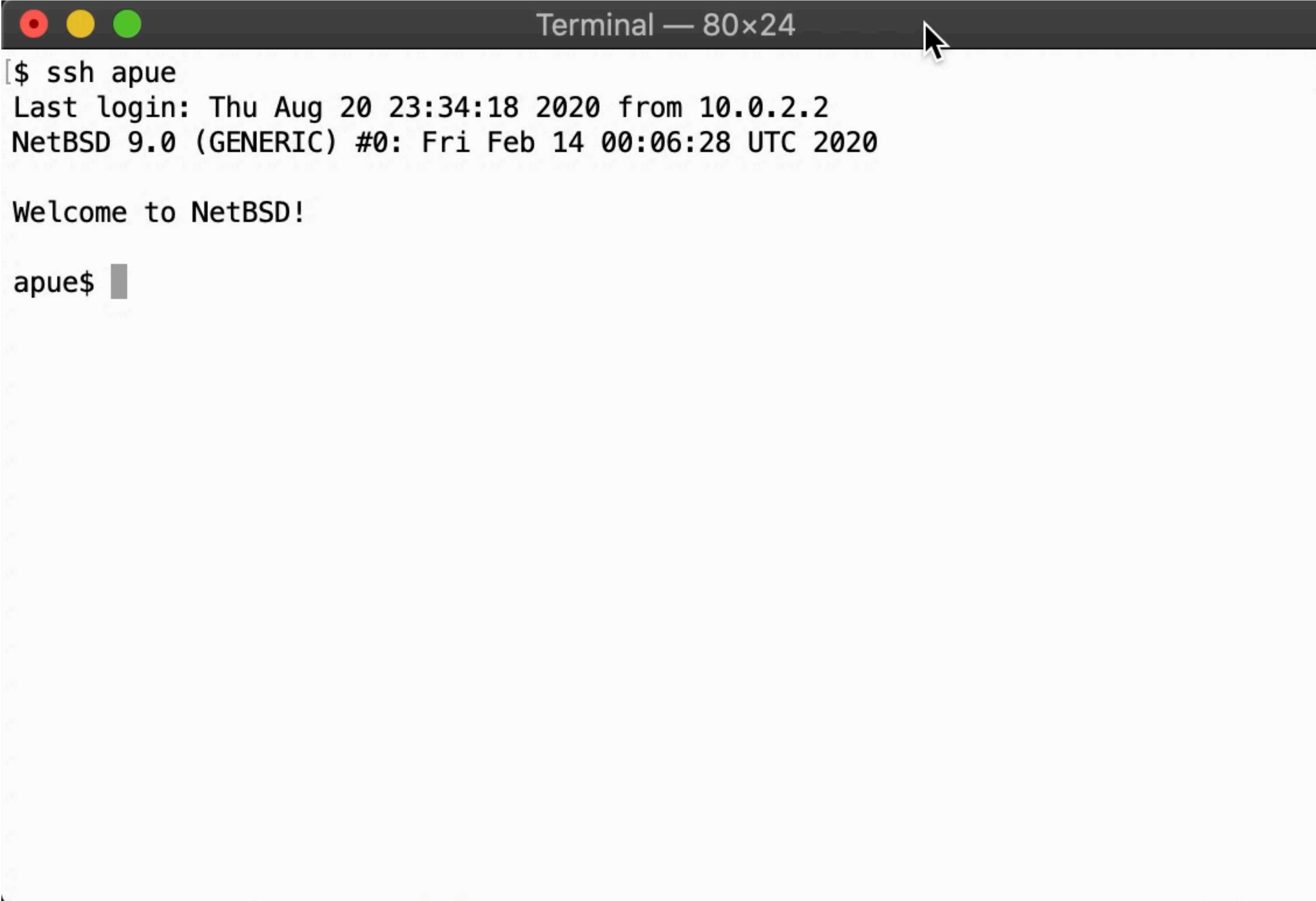
- "An Introduction to Unix"
- "What Even Is A Programming?"
- "Teach Yourself C Programming in 24 Hours!"



This class in a nutshell: the "what"



This class in a nutshell: the "what"



```
Terminal — 80x24
[$ ssh apue
Last login: Thu Aug 20 23:34:18 2020 from 10.0.2.2
NetBSD 9.0 (GENERIC) #0: Fri Feb 14 00:06:28 UTC 2020

Welcome to NetBSD!

apue$
```

This class in a nutshell: the "what"

- gain an understanding of Unix operating systems
- gain (systems) programming experience
- understand fundamental OS concepts (with focus on Unix family):
 - multi-user concepts
 - basic and advanced I/O
 - process relationships
 - interprocess communications
 - basic network programming using a client/server model

This class in a nutshell: the "why"

- understanding how Unix works gives you insights in other OS concepts
- system level programming experience is invaluable as it forms the basis for most other programming and even *use* of the system
- system level programming in C helps you understand general programming concepts
- most higher level programming languages (eventually) call (or implement themselves) standard C library functions

This class in a nutshell: the "how"

Our reference platform is NetBSD ≥ 9.3 .

You may choose to develop on e.g., your laptop running another OS, but you must make sure that your code compiles and runs flawlessly on NetBSD ≥ 9.3 using the system provided compiler (gcc $\geq 7.4.0$).

Instructions for how to install NetBSD 9.3 in a VirtualBox VM can be found here:

<https://stevens.netmeister.org/631/virtualbox/>

Instructions for how to install NetBSD-current in a UTM VM on Apple M1 hardware can be found here:

<https://stevens.netmeister.org/631/utm/>



This class in a nutshell: the "how"

- <https://stevens.netmeister.org/631/#source-code>

```
Terminal — 80x24
[apue$ cat fetch-sources.sh
#!/bin/sh
set -eu

umask 022

for set in gnusrc sharesrc src syssrc; do
    echo "Fetching ${set}..."
    ftp -V ftp.netbsd.org:/pub/NetBSD/NetBSD-9.0/source/sets/${set}.tgz
    echo "Extracting ${set}..."
    su root -c "tar xzf ${set}.tgz -C /"
    rm -f "${set}.tgz"
done
[apue$ ls -l /usr/src
ls: /usr/src: No such file or directory
[apue$ sh fetch-sources.sh
Fetching gnusrc...
Extracting gnusrc...
Fetching sharesrc...
Extracting sharesrc...
Fetching src...
Extracting src...
Fetching syssrc...
Extracting syssrc...
```

```
Terminal — 80x24
[$ ssh apue
Last login: Sun Aug 23 20:25:28 2020 from 10.0.2.2
NetBSD 9.0 (GENERIC) #0: Fri Feb 14 00:06:28 UTC 2020

Welcome to NetBSD!

apue$
```

- <https://stevens.netmeister.org/631/compare-code-exercise.html>

This class in a nutshell: the "how"

We will write a fair bit of code in this class.

Writing code is communication.

Make sure your code is:

- clearly structured
- well-formatted
- uses a consistent coding style (indentation, placement of braces, etc.)
- variables, functions etc. are sensibly named
- comments are used only when necessary, explaining the *why*, not the *how*

See also: <https://stevens.netmeister.org/631/style>

About this class

Textbook:

"Advanced Programming in the UNIX® Environment", Third Edition, by W. Richard Stevens, Stephen A. Rago

Grading:

- course participation, checkpoints: 50 points
- 2 smaller homework assignments, worth 25 points each
- 1 midterm project, worth 100 points
- 1 final project (group work), worth 200 points
- 1 final programming assignment (individual), worth 100 points

Course Notes and Participation

- create a git repository with a single text file for each lecture
- before each lecture, note:
 - what you read, what code exercises you completed
 - what questions you have
- after each lecture, note:
 - answers you've found, or especially interesting new things you learned
 - what questions remain
 - what new questions arose
- follow up on unanswered questions in class or on the mailing list
- at the end of the semester, submit all your notes

<https://stevens.netmeister.org/631/course-notes.html>

About this class

You are responsible for your work as well as your time management. If you run into challenges, contact me as soon as possible and we will work something out.

There will be no extra-credit assignments, but you may resubmit your work to address any problems identified to improve your grade.

You are responsible for your own work. You may not present as your own the ideas, code, or code samples of another, even if those are available on the internet. Any incidents of plagiarism and copyright infringement will be reported to the Dean of Graduate Academics.

<https://stevens.netmeister.org/631/#cheating>

Permitted use of (generative) AI technologies

You *may* use AI programs such as e.g., ChatGPT to help generate ideas and brainstorm.

Note that the material generated by these programs may be inaccurate, incomplete, or otherwise problematic and often stifles your own independent thinking and creativity.

You *may not* submit any work generated by an AI program as your own. If you include material generated by an AI program, it should be cited like any other reference material and must include the prompt you used to have the AI to generate the code in question.

Treat these programs like a virtual fellow student: you are allowed to "discuss" with them at a *conceptual* level, but you cannot take their code and hand it in as your own, even if you make minor changes yourself afterwards.

<https://stevens.netmeister.org/631/use-of-ai.html>

Syllabus

- Introduction, UNIX history, UNIX Programming Basics
- File I/O, File Sharing
- Files and Directories
- Filesystems, System Data Files, Time & Date
- UNIX tools: make(1), gdb(1), revision control, etc.
- Process Environment, Process Control
- Process Groups, Sessions, Signals
- Interprocess Communication
- Daemon Processes, shared libraries
- Advanced I/O: Nonblocking I/O, Polling, and Record Locking
- Encryption
- Code reading, coding style, best practices
- Review

Course Resources

Course Website: <https://stevens.netmeister.org/631/>

Course Mailinglist: <https://lists.stevens.edu/mailman/listinfo/cs631apue>

Course Slack: <https://cs631apue2022.slack.com/>

Course Videos: <https://youtube.com/c/cs631apue>

Homework

Before *every* lecture:

- review the previous week's slides and notes
- watch the video lectures and slides for that class
- follow up with questions on the course mailing list
- prepare for class by reading the assigned chapters
- do the recommended exercises

After *every* lecture:

- run all examples from the video / slides
- update your class notes

Homework

Week 1:

- make sure your class notes git repository is set up
- bookmark the course resources
- double-check that you are subscribed to the class mailing list
- customize your NetBSD VM for development
- join the course Slack channel and participate