CptS 360 (System Programming) Unit 1: Introduction to System Programming

Bob Lewis

School of Engineering and Applied Sciences Washington State University

Spring, 2022

Bob Lewis WSU CptS 360 (Spring, 2022)

イロト イヨト イヨト イヨト

Motivation (for the whole course)

- System programming techniques form the bridge between software and hardware.
- Application programmers need to know what the OS can do for them.
- All "hackers" (in a good sense) need to know system programming.

Definitions

- "UNIX-like" OSes generally follow the principles that UNIX established in the 1970's.
- On a UNIX-like OS, there's really no such thing as a "system program".
 - Does Windows let you write your own window manager?
- "System programming" means "using operating system services to implement a program".

What is the "System"?



The Command Line Style

Most programs written for this course will use the "command line" style. The benefits of this style are:

- ease of combining into special purpose scripts
- ease of automating repetitive tasks
- remotely executable
- ease of regression testing
- standardized help mechanism

Aside: All commands have a UI, but not necessarily a GUI. cf. Stephenson, Neil, *In the Beginning was the Command Line*, 1999. (http://ace.gearnine.com/temp/command.pdf)

A B K A B K

System Programming Rules of Thumb: Input/Output

- Use Standard I/O (Unit 7) for I/O.
- Inputs are specified on the command line.
- Take defaulted input (if any) from standard input (not a default file name).
- Don't have fixed input or output file names.
- Use ASCII for input and output files.
- Filename suffixes indicate file format.
- Disregard whether stdin, stdout, or stderr are "terminals".

(There are always exceptions to these rules.)

System Programming Rules of Thumb: User Interface

- Use getopt() or getopt_long() to parse command line arguments. (We'll cover this in a lab.)
- Never prompt the user for anything (*unless* the user asks for it with an option).
- Use --help, -h and/or -H to print out a brief summary of the command and options and do nothing else.
- Allow multiple input files on the command line if it makes sense. (E.g., a compiler)
- Choose a short, mnemonic name for the program, unique in the first three characters if possible.

白 と く ヨ と く ヨ と …

System Programming Rules of Thumb: Options

- Make options optional. Make the program do something reasonable without options.
- Use option letters that are comparable to those of existing programs:
 - -a all
 - -i inquire (for confirmation)
 - -n do nothing permanent (but show what the program would do)
 - -q quietly
 - -v verbose

・ 回 ト ・ ヨ ト ・ ヨ ト

System Programming Rules of Thumb: Program Construction

- Decompose the program into functions or objects.
 - Restrict the length of a function or method to a single editor window.
 - Describing what every function or method does should fit on one line.
 - That line should be a comment at the start of the function or method.
- Indent code properly.
 - Choose an indentation style and stick to it.
 - Use indent(1) if necessary, but it shouldn't be necessary.
 - Don't insert tabs in source files (but do use the Tab key).