CS-701 Lecture 15

March 22, 2001

Vickery's Law

"The computer will go down two days before the project is due."

- You have to plan for it.
- No extensions!

- Problem 1: If an assignment counts 20% of the course grade, and there is a 1% per day penalty for submitting the assignment late, what effect will submitting the assignment one day late have on a student's course grade?
  - Answer: not much!
- Problem 2: If an assignment is due at midnight, what is the probability that you will lose 1% by submitting it at 12:01? 12:10? 1:00? After Vickery goes to bed? The next afternoon?
  - Answer: It varies from 0.00 to 1.00

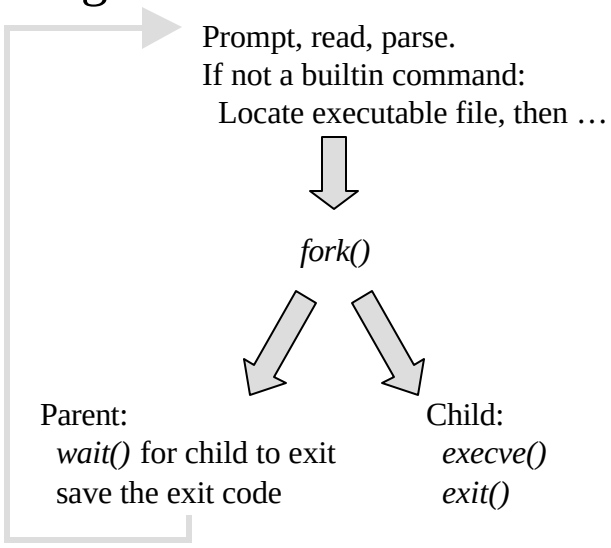
Examples of shells

- Programs that read and execute commands
  - *ash, bash, ksh, sh, csh, tcsh, zsh*
    - Allow programming by putting commands in "script" files.
    - Windows imitated this capability (sort of) with .bat files.
  - *script*
    - Copies all shell input and output to the typescript file.
  - *make*
    - Sophisticated mechanisms for deciding what commands to execute.

Executing External Commands

- Use *fork( )* to create a process to execute the command.
  - Returns process id of child to the parent; 0 to child. Otherwise, both processes are identical and continue to execute the same program.
- Child process calls *execve()* to load and execute the target file.
  - Have to find it first! Need to anticipate that PATH might change.
  - Have to set up argv[ ] and envp[ ]
  - There are library functions that take care of some of this, and then call *execve()* for you, but for this project, you are to use the system call.
- Parent process uses *wait( )* to get the child's exit code.
  - It's critical for the child process to exit even if the call to *execve()* fails.
  - And then there is the zombie problem …

Shell Algorithm for External Commands



Checking a File's Status

*char *pathname = "/usr/bin/ls;*

struct stat statBuf;

stat(char pathname, &statBuf)*

– *Returns zero if file exists. statBuf will contain lots of information about the file, such as size, time/date of last modification, etc.*
  - *But for this assignment, all we need to know is that it exists; execve() will check that it is executable, etc.*
  - *It would be better to check the permissions and skip files the user cannot execute. (Optional part of the assignment.)*