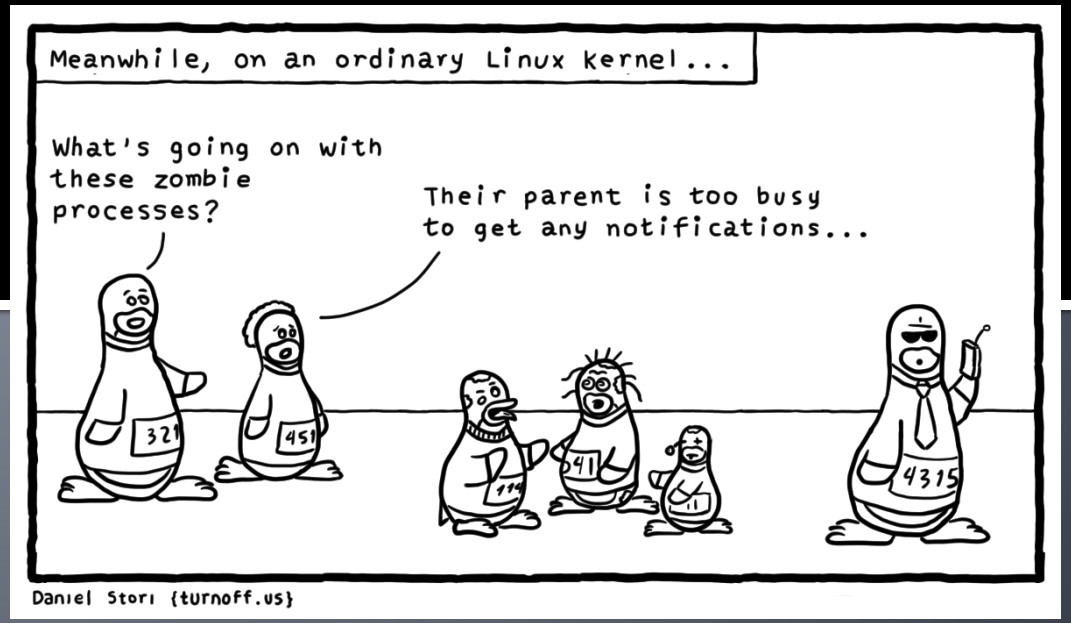
SO Project 1: BYOS
- Built Your Own Shell -



Implement a Shell

- Download [REF]
- Based on this code, implement Xsh (choose any letter/word of your choice for X)

Shell Features: Prompt

- The shell should run continuously, and display a prompt (%, similar to $) when waiting for input
- Include the full path of the current directory also in the prompt followed by the “>” sign (e.g., /usr/home/me/Desktop>)
  - The shell should read a line from input one at a time

Shell Features: Built-in Commands

- Using system call wrapper functions, implement the following **built-in** commands:
  - ic: Print directory
  - cd: w/ no arg goes to user home and w/ an argument to that directory
  - cm <file_path> <mode>: Change the *file's* permission to the *mode* bits.
  - co < file_path> <username>:Change the *file's* owner to *username*.
  - surt: exits the shell. No child processes should be alive afterwards.

- No new process will be created to execute the builtin commands

Shell Features: External Commands

- Any other command should be executed as an executable file
  - For example, typing “a.out" should execute the file a.out
  - The file can be in the current directory or in any of the directories specified by the PATH environment variable (use getenv to get the value of PATH)
  - The executable name may include a path, i.e., *path/a.out*
- The external commands should be executed after creating a new process and then executing the executable in it

Shell Features: Background Execution

- Normally when you type a command at the shell prompt, the prompt does not return until the command is finished.
- For background executions, the prompt returns immediately, the command continues execution in the background
- Typing an "&" at the end of a command (for ex., a.out&) should make it execute in the background
  - Background execution needn't be supported for builtin commands
  - You can use waitpid for fg and bg mode

Shell Features: Errors

- Send error messages to stderr
  - Check “perror” as an option
- For example:
  - The indicated executable not found
  - The directory to change to does not exist
  - File indicated for cm or co does not exist

Deliverables

- Provide the code
  - The code should not pass 1000 LoC
  - Use self-explanatory function/variable names
  - Provide comments for each function/variable and important statements
- Provide a set of slides to explain to your "jefe" how your programs **designed** and **worked**
  - Provide screen captures for all the shell features

Conclusions

- Add a conclusion section in the slides to summarize your overall experience
  - For example:
    - For which topics and how, this project helped you
    - Where did you have difficulties?
    - …

No hard coded values! :-)

"hard coded" stuff in the program ☺

Deadlines & Academic Responsibility

- Announcement: 15/11/2021
- Deadline: 5/12/2021
- Individual project, i.e., no code sharing
- Prj2 Announcement (tentative): 13/12/2021
- Prj2 Deadline (tentative): 12/01/2022