Wednesday 8th March, 2023

Instructions

- Group enrolment on Themis
  `https://themis.housing.rug.nl/course/2022-2023/os`
- Submit in pairs
- Use Lab 1 submission as starting point for Lab 3
- Deadline: 17th March, at 23:59am
- Programming language: C

Requirements

- Lab 1:
  - Execute a program from user's search path ($PATH).
  - Command composition (&&, ||, ;)
  - String parsing
- Lab 3:
  - Pipeline
  - I/O redirection
  - cd builtin

Pipeline

Run commands in a pipeline.

- Examples
  - `ls -l | grep "\.txt\$"`
  - `./a.out | ./b.out | ./c.out`
- with `a | b`, the output of a is directed into the input of b.
- Do not manually implement this! Use pipe(2)!
  - After forking, the child has a copy of the same pipe the parent has.
  - Make sure to let the child close the unused ends of the pipe after forking.
  - Close the write end of the pipe to send EOF.
- Start all processes of the pipeline simultaneously. Do not wait for the previous step to end.
- Exit code for a pipeline command is the exit code of the last command in the pipeline.

I/O redirections

Redirect I/O to files

- Examples
  - `./a.out < 1.in > 1.out`
  - `./a.out | ./b.out < 1.in > 1.out`
- Definitions
  - `< 1.in` redirects the content of the "1.in" file into the command's input buffer.
  - `> 1.out` redirects the content of the command's output buffer into the "1.out" file.
  - For pipelines, < redirects into the first command, > redirects out of the last command.
- Useful functions
  - open(2) - Obtain file descriptor for files
  - close(2) - Clean up opened file descriptors
  - dup(2) - Helps insert file descriptors into your pipeline

cd

Switch the working context of the shell to the provided directory

- User may provide both relative and absolute paths.
- A rather useful Linux function: chdir(2)
- Possible errors:

No path provided:	Error: cd requires folder to navigate to!
Invalid folder:	Error: cd directory not found!

Closing words

- As in lab 1, there are bonus points to be had.
  - Refer to the assignment for possible ideas
  - **NOTE:** You cannot get bonus points for ideas implemented in Lab 1.
- There is extensive documentation available on all Linux functions you need for this lab, please read the manual!
- We will continue checking your code style. Make sure your code is clean!