CS 326: Operating Systems

getopt

Lab 0

Command Line Arguments

- We're all familiar with passing arguments to our programs on the command line:
  - `./my_app file.txt`
- Unix userland utilities also support command line **options**:
  - `rm` **`-r`** `my_dir`
  - `ls` **`-lsh`**

Parsing Command Line Options

- Parsing these options by hand can be a bit tricky
- Most of the command line utilities follow a similar syntax
  - Options are indicated by - or --
  - -r is often recursive
  - You can combine options: -rv
- Many of these tools use the **getopt** function to automate this process

getopt Usage

- getopt is called in a loop
  - When -1 is returned, there are no more options to parse
- A switch statement is used to determine what to do with the options
- A second loop is used to process the remaining non-option arguments

Specifying Valid Options

- The getopt call defines what options are valid:
  - `getopt(argc, argv, "cs:")`
- Options with a colon (`:`) after them accept an argument string. In our example:
  - c = just a Boolean option (was it specified or not?)
  - s = takes a string (-s “hello”)

Example

- See getopt.c on the schedule page
- HW1 enhances HW0 by adding command line options
    - See the assignments page for details!