Computer Practicum 1

Intro to Bash scripts - exercises

Vida Groznik

Exercise 1

Write a script that takes a filename and 3 keywords. It should grep in the file for all 3 keywords and display for each keyword the number of matches followed by the line numbers where the matches did occur.

- No other output on stdout should be produced by the script
- If the file cannot be read the script should exit with a return code 1, else with code 0 (see help exit if you do not know the exit command)
- Count the number of characters excluding comments

Exercise 2 & 3

2. Write a shell script that takes 3 arguments and prints them in reverse order If -h is entered anywhere a short description should be printed as well.

3. Write a shell script that does the following when given a path as first arg:
   - If the path is a file, print whether it is executable and print the file size
   - If the path is a directory move (cd) to it

Exercise 4

Write a script that takes two integer values as args, `I` and `J`. The script should:

- create directories named `1, 2, . . . , I`
- Use `touch` to put empty files named `1` till `J` in each of these directories
- Print an error if a negative value is provided for `I` or `J`
- If any of the files exist, the script should exit with an error.
- Provide help if one of the args is `-h`, then exit the script.
- If the third argument is a file, the script should copy this file to all locations instead of creating empty files with `touch`.

Exercise 5

Implement the `seq` command in bash:

- If called with a single argument, print all integers from 1 to this value, i.e. `1 seq 5` should give

  ```
  1
  2
  3
  4
  5
  ```

- If called with two arguments, print from the first arg to the second arg, e.g. `seq 3 5`:

  ```
  3
  4
  5
  ```

Assume that the first number is always going to be smaller or equal to the second number.

Exercise 6

Write a script that takes the following arguments:

- `-h, -q`
- `--help, --quiet`
- `-f` followed by a `filename`
- anything else should cause an error message

Once the arguments are parsed the script should do the following

- Print help if `-h` or `--help` are present, then exit
- Check that the `filename` provided is a valid file, else throw an error and exit
- Print a nice welcome message, unless `--quiet` or `-q` are given