

## Linux lab 5 - Script intermediate

Assignments marked with \* are optional (implement as much as you can).

### Goal

- To be able to create small simple scripts from a specification.

**Preparations: Look through previous labs. Also look up the following.**

“runlevel” \_\_\_\_\_

\$# \_\_\_\_\_

shift \_\_\_\_\_

expr \_\_\_\_\_

test \_\_\_\_\_

seq \_\_\_\_\_

expr \_\_\_\_\_

crontab \_\_\_\_\_

### Laboration

1. Make a script that prints the numbers 1 through 100 using a for loop and seq.
2. Make a script that prints the numbers 1 through 100 using a while loop using arithmetic calculations.

3. Write a **function** that takes two numbers as arguments and adds them and prints the result.
4. Make a script that lists all files and folders in the current folder using a for loop.
5. Change the script #3 to a script that lists all folders in the current directory.
6. Create a script “multi-ls” that takes a number of folders as argument and lists each of them. Identify each listing by printing the folder name to the terminal.  
E.g. multi-ls /bin /home /boot
7. You are supposed to write a script that on invocation lists the content of your home directory and sends that listing to a file in /tmp.
8. Schedule the script in #6 to be executed once every minute using cron. How could you test that script #7 is working?

- 
9. Download the program “watchdogd” and copy it to /usr/sbin.
    - a. Write a startup script that handles the arguments start and stop by simply starting watchdog in the background or executing pkill on it. Save it under the name “watchdog” and copy it to the folder /etc/init.d .
    - b. Link to the script #8.a from the folder for the current runlevel (/etc/rc2.d) to make it start.
    - c. Link to the script #8.a from the folder for reboot (rc6.d) and halt (rc0.d) to make it stop.
    - d. To avoid problems, do not test your solution until after the lab assistant has passed you.

10. \*Implement the game described in Figure 1 (download the random generator rnd from the course page):

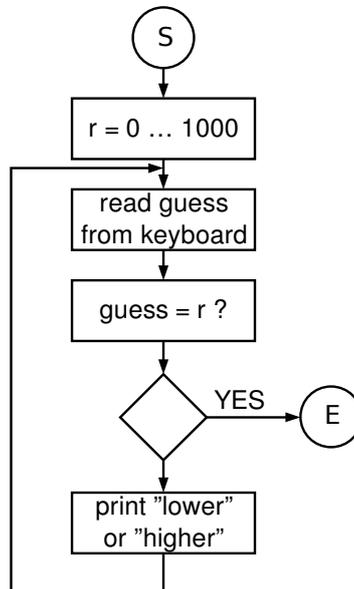


Figure 1. Flowchart of a simple guessing game.