Implement a Small UNIX Shell

Emmanuel Benoist, Christian Grothoff, Pascal Mainini

Fall Term 2019/2020

This exercise must be done inside the clone of your own fork of the skeleton git repository.

1 Presentation of the files

The file `shell.c` contains a C file that must be completed. This is the main file you have to change.

The file `makefile` contains the instructions for the compilation.

2 Goal of the exercise

The goal of this exercise is to write a program in C, which emulates a Unix shell (a small one).

Your shell must:

1. Print a shell prompt (`"$ "`) for each command.

   Example:

   ```
   $
   ```

2. Read a stream of simple commands to execute from stdin.

   Example:

   ```
   $ echo Hello
   Hello
   $ expr 1 + 3
   4
   ```

3. Support setting shell variables and expansion.

   Example:

   ```
   $ A=5
   $ echo $A
   5
   $ echo ${A}BC
   5BC
   $ A=$A$A
   echo $A
   55
   ```

4. Support checking the status of the last command via $?.

   Example:

   ```
   $ true
   $ echo $?
   0
   $ false
   $ echo $?
   1
   ```

5. Support launching processes in the background with '&'.

   Example:

   ```
   $ firefox &
   $
   ```

   (here, Firefox should run and your shell should accept additional commands as inputs while Firefox is running). Without '&', the shell must only accept new commands after the previous command has terminated!

6. Support redirection of stdin, stdout and stderr with ">", "<" and "2>".

   Example:

   ```
   $ echo Hello > test.txt
   $ cat − < test.txt
   Hello
   ```

7. Support the pipeline operator "|".

   Example:

```
$ echo Hello | tr l L | tr o ?
HeLL?
```

A passing grade will be awarded for completing 6/7 of the features.

3 Upload the result to the Git repository

You must upload the file you wrote `shell.c` to the repository.

```
machine−:$ git add shell.c
....
machine−:$ git commit −m"solution for exercise 5"
....
machine−:$ git push origin master
```

4 Get feedback

You can inspect the results of the automated tests for your program inside gitlab CI (`https://gitlab.ti.bfh.ch`, go to `CI/CD > Jobs` in your forked project). There, you will also find the amount of points given.