Task 1:

Makefile: changed name of “shell2” to “shell”.

Shell.c:

- Change the prompt variable.
- Line 31-34: Exit programm if error occurs.
- Line 36-38: If no error, print the prompt and flush everything to the screen.
- Line 13: Give the prompt size 80 since a command can be max 80 chars long.
- Line 63-68: Tokenize every command from command line.
- Line 75-76: change directory function.
- Line 78-79: Print out current directory.
- Line 81-92: Prints strings and values of variables.
- Line 94-95: Terminates shell.
- Line 97-98: Prints current values of environment variables.
- Line 100-102: Set environment variables.

Task 2:

- Line 106: if command is not in our built-in commands, we fork off a child process which takes care of other commands.
- Line 107-112: If we are in the child process, we take the command and run it by using the execvp() function. If execvp() throws an eros, the program keeps running to line 110 and prints that there was an error.

Task 3:

- Line 69-72: if the last token on the command line is a “&”, we set the background_process boolean to true and delete the “&” token.
- The shell will return a new command line and return the background process as soon as its done.
- Line 113-115: If we are in a background process, the program assigns pid2 the value -1 which means that the process will be executed within the next loop.

Task 4:

- Line 145-147: function to handle signal of user and end foreground process.
- Line 26: Waits for signal of user and ends foreground process but not the whole shell.
- Line 108 and Line 122: take the user input and ignore it.

Task 5:

- Line 118-120: If we are not inside a background process, we for off another child.
- Line 122-123: we ignore the user signals and wait for 10 seconds.
- Line 124-126: if the process hasn’t completed, we kill the process by killiing its pid.