

Grading Rubric for Assignment 1

Implementation of shell with fork()	20%
Implementation of chdir	8%
Implementation of history	10%
Implementation of limit	10%
Piping with FIFO	25%
Implementation of signal handling	8%
Trace with ltrace and strace	4%
Code quality and readability	15%
<div><div>Error handling</div><div>Signal handling</div><div>Formatting</div><div>Comments</div><div>Misc</div></div> <div><div>[3%]</div><div>[3%]</div><div>[3%]</div><div>[3%]</div><div>[3%]</div></div>	

Notes:

- Any plagiarized content (code or report) will be heavily penalized
- Follow the following code structure to make grading easier and efficient

Code structure:

- Implement your shell in one source file and name it 'tiny_shell.c'
- Let the main() function have only following code. Implement all other functionality in **get_a_line()** and **my_system()**

```
while (1) {  
    line = get_a_line();  
    if length(line) > 1  
        my_system(line);  
}
```
- Use following signature for the **my_system()**

```
int my_system(char* line)
```

Unrestricted

- Debugging with `printf()` is not a good idea. It does buffered writes. Thus, you may not see the prints on the console even after the program had gone past a certain point unless the buffer is flushed. As a workaround follow every `printf()` statement with `fflush(stdout);` to force a buffer flush
- Use `gdb` for debugging. Will try to have a `gdb` tutorial before the deadline. If not possible will schedule one before the next one. Look up follow-fork-mode for using `gdb` with `fork()` and multiple processes. These concepts will be covered in the tutorials this week.

Extra

- **Limit** command: when running ``limit <amount_of_mem>`` in the `tshell`, this limit is placed on all child processes created
- **FIFO** piping: when running ``tshell <path_to_FIFO>``, we allow piping for any 2 commands
 - The FIFO is only used for piped commands
 - We will not be testing chain piping - eg. ``ls -l | grep ... | wc``
- **chdir** should work for both relative and absolute paths
 - eg. ``chdir ..``
- **Signal** handling: we only test this to exit out of the `tshell` (we will not test this on killing child processes)
- **Submission**: if you compile with multiple `.c` and `.h` files, use a **Makefile** AND include a **README.md**
 - Or you can stick w/ 1 `.c` file, no readme or Makefile required
- **Wildcards**: not tested