Operating Systems

CMPT 424 • Fall 2024

– *i*Project One - 50 points

Goals

To experience and come to intimately understand our initial project as well as the awesomeness that is TypeScript. Then to add all the functionality specified below.

Functional Requirements

- ☐ Alter the ver command to display your own data. [1 point]
- ☐ Add some new shell commands: [4 points]
  - date - displays the current date and time
  - whereami - displays the users current location (use your imagination)
  - something else interesting and creative; surprise me
- ☐ Enhance the *host* display with a graphic task bar that displays . . . [5 points]
  - the current date and time
  - status messages as specified by the user with a new shell command: status <string>

  example: status I love Operating Systems
- ☐ Implement scrolling in the *client OS* console/CLI. [20 points]
- ☐ Other console/CLI enhancements: [15 points]
  - Accept and display punctuation characters and symbols.
  - Handle backspace appropriately.
  - Implement command completion with the tab key.
  - Provide command history recall via the up and down arrow keys.
- ☐ Display a BSOD message (on the CLI) when the kernel traps an OS error. [2 points]
  - Add a shell command to test this. Remember to include it in the help.
- ☐ Add a shell command called load to validate the user code in the HTML5 text area (id= "taProgramInput"). Only hex digits and spaces are valid. [3 points]
- ☐ [challenge] Implement line-wrap in the CLI. [+5 points]

Implementation Requirements

- ☐ Your code must separate structure from presentation, be professionally formatted, use and demonstrate best practices, and be free of compiler errors. If there are compiler errors it's (-10 * project number) points off. [−∞ if not]
- ☐ **Do not** break GLaDOS. Don't make me flood the Enrichment Center with neurotoxin. Again. (For science!)
- ☐ **Do** cite all elements that not are entirely yours: code references; AI assistants; Hall-of-Fame projects; help from tutors, friends, strangers; etc.

General Hints

Read up on the Canvas before you mess with the console/CLI. There are some helpful links on our class web site about the HTML5 canvas. Do some Canvas experiments on your own. It's really quite amazing what you can do with it.

Remember the utility of comments and how much their presence and quality effect my opinion of your work. Also, write code that is **uniquely yours**, with comments that say something about the code that the code cannot say about itself.

Make many commits to Git. I do not want to see one massive "everything" commit when I review your code. (It's −∞ if you do that.) Commit early and often. And make sure your commit messages are descriptive, informative, and entertaining. If you've been listening to any particularly cool music while programming this assignment, let me know so I can try it. I was listening to Squeeze while I wrote this.

Specific Hints

- Keep .js and .ts files in **separate** directories.
- An empty text area means the source code is not valid.
- Scrolling is difficult. Think carefully. And **do not** scroll the entire canvas. You must scroll the text *within* the canvas, but it only has to scroll forward.
- Regarding punctuation characters: & is not the same as ↑.
- Command completion with the tab key can be tricky if there is more than a single match for the letters typed before pressing tab. Think carefully about how you handle that. I want to see elegant solutions.
- Be sure to add a .gitignore file so your your IDE configuration files and other messy stuff are excluded from your Git repository.
- Check out some Command Line Interface Guidelines.

Submitting Your Work

Add me (username *Labouseur*) as a collaborator to your **private** GitHub repository. Then e-mail me the URL. Send this to me before the beginning of the class in which this is due.

Note: Your project will not be accepted for grading unless and until your GitHub repo is private and you have added me as a collaborator.