

Operating Systems

CMPT 424 • Fall 2018

-iProject One - 100 points

Goals	To review 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	<ul style="list-style-type: none"><input type="checkbox"/> Alter the <code>ver</code> command to display your own data. [5 points]<input type="checkbox"/> Add some new shell commands: [15 points]<ul style="list-style-type: none">• <code>date</code> - displays the current date and time• <code>whereami</code> - displays the users current location (use your imagination)• something else interesting and creative; surprise me<input type="checkbox"/> Enhance the <code>host</code> display with a graphic task bar that displays ... [10 points]<ul style="list-style-type: none">• the current date and time• status messages as specified by the user with a new shell command: <code>status <string></code> example: <code>status i love operating systems</code><input type="checkbox"/> Implement scrolling in the <i>client OS</i> console/CLI. [30 points]<input type="checkbox"/> Other console/CLI enhancements: [30 points]<ul style="list-style-type: none">• 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.<input type="checkbox"/> Display a BSOD message (on the CLI) when the kernel traps an OS error. [5 points]<ul style="list-style-type: none">• Add a shell command to test this. Remember to include it in the help.<input type="checkbox"/> Add a shell command called <code>load</code> to validate the user code in the HTML5 text area (<code>id= "taProgramInput"</code>). Only hex digits and spaces are valid. [5 points]<input type="checkbox"/> [challenge] Implement line-wrap in the CLI. [+10 points]
Implementation Requirements	<ul style="list-style-type: none"><input type="checkbox"/> Your code must ... [−∞ if not]<ul style="list-style-type: none">• separate structure from presentation.• be professionally formatted.• use and demonstrate best practices.• make me proud to be your teacher.<input type="checkbox"/> Do not break GLaDOS. Don't make me flood the Enrichment Center with deadly neurotoxin. Again. (For science.)
General Hints	<p>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.</p> <p>Remember the utility of comments and how much their presence and quality effect my opinion of your work. Also, write code that is uniquely you.</p> <p>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 — if possible — entertaining.</p>

Operating Systems

CMPT 424 • Fall 2018

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.
- Be sure to add a .gitignore file so your your IDE configuration files and other messy stuff are excluded from your Git repository.

Submitting
Your Work

Add me (Labouseur) and Prof. Gildein (gildmi) as a collaborators 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. Thank you.

Note: Your project will not be accepted for grading unless and until your repo is private and you have added *Labouseur* and *gildmi* as collaborators.

```
>help
Commands:
ver - Displays the current version data
help - Lists all available commands
shutdown - Shuts down SvegOS
cls - Clears the screen
man <topic> - Displays the manual page for <topic>
trace <on | off> - Enables/disables the OS trace
rot13 <string> - Does rot13 encryption on <string>
quantum <integer> - Changes the CPU quantum
prompt <string> - Sets the prompt
date - Displays the current date and time
whereami - Displays the current location of the user
status <string> - Sets a status message
ps - Shows all active processes
kill <integer> - Terminates the specified process
load [<priority>] - Loads the specified user program
\ <regex> <function> - Filters function output
bsod - Enables the blue screen of death
run <processid> - Executes a program in memory
runall - Executes all programs
create <filename> - Creates the specified file
read <filename> - Reads the specified file
write <filename> "data" - Writes the specified file
delete <filename> - Deletes the specified file
format - Initializes disk
ls - Lists all files on disk
```