CIS511 Semi-Major Output Created By: Gregory Cu
By group

Updated as of: Sept. 17, 2025
General Instructions: An OS emulator needs a command interpreter and a display output.

searsg@login:~

```
  GNU nano 2.3.1                    File: pytest.sh                                  Modified

#!/bin/bash

##This is called the SHEBANG. This tells the shell to interpret and run the SLURM
##script using the bash (Bourne-again shell) shell.


##SLURM Commands using #SBATCH --[options]

#SBATCH --job-name demoTest                ##give the job a name
#SBATCH --output demo.out                  ##creates a file to store the job output
#SBATCH --mail-user searsg@uhd.edu         ##sends email to user's email for start/complete of job
#SBATCH --mail-type BEGIN                  ##creates an event type to be mailed to user when event occurs
#SBATCH --mail-type END                    ##creates an event type to be mailed to user when event occurs
#SBATCH --error test.e.txt                 ##creates a file to store any error output

##Load the modules needed for the job to run
module load python3


##List the executables commands (or a string of them)
##Example for code written in Python

python3 pytest.py 90



^G Get Help     ^O WriteOut     ^R Read File    ^Y Prev Page    ^K Cut Text     ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page    ^U UnCut Text   ^T To Spell
```

PowerShell

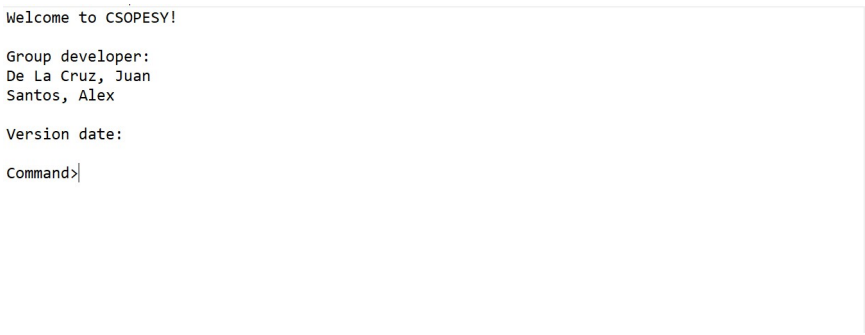
```
PowerShell 7.4.0
PS C:\Users\Tiffany> $PSVersionTable

Name                           Value
----                           -----
PSVersion                      7.4.0
PSEdition                      Core
GitCommitId                    7.4.0
OS                             Microsoft Windows 10.0.22631
Platform                       Win32NT
PSCompatibleVersions           {1.0, 2.0, 3.0, 4.0…}
PSRemotingProtocolVersion      2.3
SerializationVersion           1.1.0.1
WSManStackVersion              3.0

PS C:\Users\Tiffany> function SayHello {
>> write-host "Hello, World!"
>> }
PS C:\Users\Tiffany> SayHello
Hello, World!
PS C:\Users\Tiffany>
```

Shell Reference
Please refer to a general Linux/Windows powershell/Windows command line. This serves as a strong reference for the design of your command-line interface.

Checklist of Requirements
Your system must have ALL the following features implemented properly.

Requirement	An "OS emulator" that accepts a command input with display output
Description	  A main menu console that have the following features: 1. Text marquee or ASCII graphics marquee 2. Accept commands to change the behavior of the text marquee
Requirement	Command interpreter for the OS emulator that accepts the following commands
Description	From the main menu, the user can use the following commands: • "help" – displays the commands and its description • "start_marquee" – starts the marquee "animation" • "stop_marquee" – stops the marquee "animation" • "set_text" – accepts a text input and displays it as a marquee • "set_speed" – sets the marquee animation refresh in milliseconds • "exit" – terminates the console
Requirement	The OS emulator must have the following components
Description	The OS emulator components are: • Command interpreter – accepts command and control the marquee logic • Display handler – handles the display for the command interpreter and marquee logic • Keyboard handler – handles keyboard buffering and polling • Marquee logic – handles the animation logic for the marquee text

ASSESSMENT METHOD
Your CLI emulator will be assessed through a black box quiz system in a time-pressure format. This is to minimize drastic changes or "hacking" your CLI to ensure the test cases are met. You should only modify the parameters and no longer recompile the CLI when taking the quiz.

Test cases, parameters, and instructions are provided per question, wherein you must submit a video file (.MP4), demonstrating your CLI. Some questions will require submitting PowerPoint presentations, such as cases explaining the details of your implementation.

IMPORTANT DATES
See AnimoSpace for specific dates.

Week 4	Test case demo submission

Submission Details
Aside from video files for the quiz, you need to prepare some of the requirements in advance, such as:
- SOURCE - Contains your source code. Add a README.txt with your name and instructions on running your program. Also, indicate the entry class file where the main function is located. An alternative can be a GitHub link.
- PPT – A technical report of your system containing:
    - Command recognition
    - Console UI implementation
    - Command interpreter implementation

Grading Scheme
- You are to provide evidence for each test case, recorded through video. Each test case will have some points allocated. The test cases will be graded as follows:

Functional		
No points	Partial points	Full points
The CLI did not pass the test case. **NO WORKAROUND** is available to produce the expected output.	The CLI did not pass the test case. **A workaround** is available to produce the expected output.	The CLI passed the test case using varying inputs and produced the expected output.