

CSC3002

Introduction to Computer Science: Programming Paradigms

Project Report

A Simulated Operating System



Adriel Abraham Intan 118010485

Brandon Honata 118010484

Harry Tanamas 118010501

Steven Sugihermanto 118010499

Course Coordinator: Prof. Rui Huang

The Chinese University of Hong Kong, Shenzhen

2020

Remarks

The project that we have chosen to create is a fake Operation System. Compared to what we have proposed in our proposal, there are two improvements. **First**, we proposed that we would create a user interface similar to the IBM computers in early days. However, along the process, we realised that it would be better for us to create a modern-like user interface that is more interactive and user friendly. So that's what we made. **Second**, we proposed to deliver only 6 apps (file manager, task manager, calculator, clock, kernel, text editor, and calendar). However, we decided to walk another extra mile by creating a useful app called GPA Planner. GPA Planner is an app made exclusively for CUHKSZ students to plan for their GPA. This is different with any GPA calculators present on the internet. Are you excited to read our report? Enjoy!

Motivation

So you just got back to your room, opened your laptop and for the first time realised that something was lacking. It worked totally fine, the processing speed was fine, and not a single virus was found. It worked like any typical PC with a standard operating system does. Ah, that's it! You did not want to have the same operating system everyone was having. You felt bored with its interface and the way everything was configured. You could not hide any files without your mom knowing that you were hiding them. Your list went on...

“Can I really implement a real operating system?”. Sure, but it might be difficult dealing with the fact that it should be the only “mediator” between your computer hardware and program applications (e.g., Chrome, MS Word, games). “Ah, I can just write a **fake operating system with the following components:**

- Kernel
- File manager
- Calculator (+, -, x, /)
- Calendar (reminder?)
- Clock (system)
- Text editor
- Task manager.

Sounds great, huh?”. That's great we would be glad to assist you along the way.

Introduction to Operating System

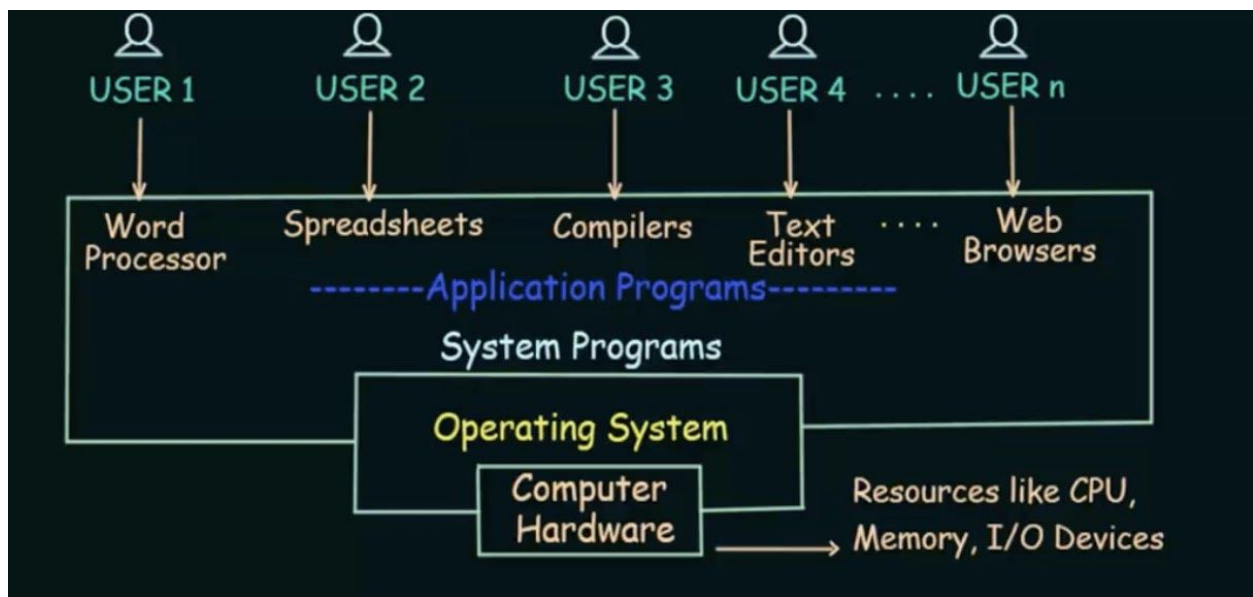
Before creating any operating systems, may it be fake or real, it would be necessary for you to understand the basics of operating systems. An operating system is a critical foundation that acts as an interface between a computer and the end-user. Every computer must have at least one OS to run other programs. It helps humans to interact with a computer directly without any difficulties. OS facilitates applications such as Chrome, Word, Games, etc. as an environment in which it will run and perform its task. The OS helps you to communicate with the computer without knowing how to speak the computer's language. Since the user can't use any computer or mobile device without having an operating system. Therefore an OS is a necessary foundation that helps humans interact comfortably with a computer.

There are other reasons why every computer needs an OS:

- The operating system lets users run and use the same software on different hardware without having any differences.
- When writing a program, developers do not know the specifications of each end-user hardware. Since there exist a lot of different computers and laptops with varying kinds of hardware. It would be impossible to create different types of software to support all sorts of different hardware configurations. Thus, the operating system isolates the application from those sorts of details.
- Creating standalone applications for computers is not impossible. However, it requires a lot more work and is also more expensive compared to using an OS. Another difficulty would be getting the program into the machine so it could be executed.
- **Consistent user interface:** Operating system is a general interface that helps various users an easy-to-work environment. Therefore the user would not need to learn a different UI each time they want to use a different computer. It would increase productivity so that users can work as quickly as possible.
- **Multitasking:** Operating systems organize memory and enable multiple programs to run in their own space and interact via shared memory with each other. Through

multitasking, users are allowed to perform many tasks at a time on a device, giving them convenience and positive experience.

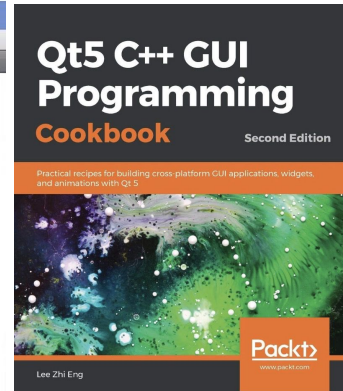
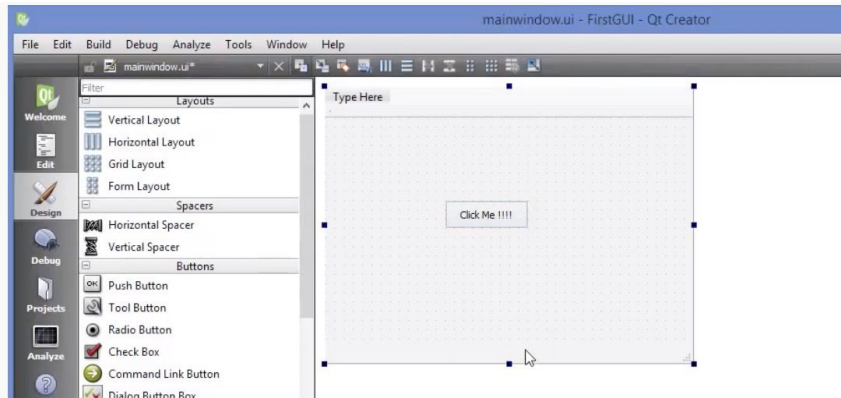
In conclusion, the goal of an Operating System is to execute user programs and to make tasks easier. The following figure summarizes how the OS acts as a medium between the hardware unit and application programs.



Related Works

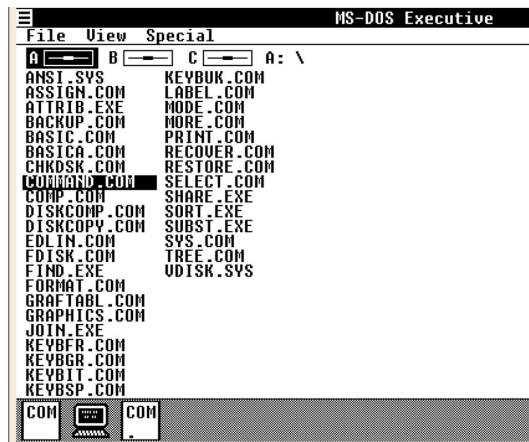
Graphical User Interface (GUI)

We found that our team can use C++ and QT to implement our GUI. Our team chose QT because it enables us to design and create an interface like we are using photoshop. The mechanism is that we design our buttons and prompts inside the QT creator app. We can then export this file into a working code of interface. C++ codes then create an “environment” to run the working code, while at the same time adding more control over the interface. Our further learning resources upon creating the GUI are the youtube **playlist QT C++ GUI Tutorial For Beginners** and the book Qt5 C++ GUI Programming Cookbook (2nd edition) by Lee Zhi Eng.



Operating System (OS)

To make a fake operation can mean to copy some functionalities of a real operating system. We found a conceptual youtube playlist about this called “operating system” by Neso Academy. This playlist taught our team about the basics of OS (computer system structure, storage structure, and input/output structure), OS structure and services, OS user interface, system programs, and many more. A demo system by IBM also helped our team to visualize the final display and guided our team about the set of programs that our fake OS should support.



```

COMMAND
TREE    COM    2831    3-07-85    1:43p
UDISK   SYS    3307    3-07-85    1:43p
      36 File(s)    61440 bytes free

A>md inspiring
A>cd inspiring
A>dir

Volume in drive A has no label
Directory of A:\INSPIRIN

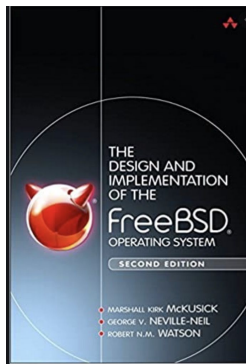
.          <DIR>        6-23-17    1:51p
..         <DIR>        6-23-17    1:51p
      2 File(s)    60416 bytes free

A>rm inspiring
Bad command or file name

```

Kernel and Command Interpreter

The same “operating system” youtube playlist by Neso Academy did a great job of mentioning that one can implement a kernel and command interpreter as one integrated program. For detailed information regarding the functions that our kernel should support, our team refers to the book Design and Implementation of the FreeBSD Operating System (2nd edition) by Marshall K.M.



Our Work

First, the outcome of the project has met all expectations of our original proposal and even better. Since, our team is able to implement more applications than what we expected. Moreover, the design or UI of this operating system is **simple yet beautiful**. Using neat visuals and easy to follow applications show that our team know how to write meaningful QT applications: both its interface and back-end logic. Using our OS, we believe that most users will say that they have a great experience as if comparable with

Mac OS, while in fact it is not. Thus, we'd say that the ability of delivering such neat visual in our work is a kind of a strength.

Second, our OS supports the **multitasking feature** that makes the user easier in "jumping" across applications. Would be satisfying to have such application, where the user needs to close the currently opened application every time the user wants to open another application? We believe that this project is our chance to mimic the real world expectation as closely as possible. It was certainly not easy to do this, but thankfully we managed to build it.

Third, rather than just creating existing apps, we put our minds together for **pioneering our own app**, the GPA Planner application. Creating this application shows that we actually put the extra mile and care for doing this course well. The GPA Planner that we created has a code base of almost 500 lines. Although many lines don't necessarily imply complexities, we'd assure you that the implementation of this application requires hard work and many hours of effort. We think it would be fair to humbly ask for a better grade.

Contributions

| Name | Student ID | Division of Work |
|----------------------|------------|--|
| Adriel Abraham Intan | 118010485 | About System, File Manager, Login Page, Task Manager |
| Brandon Honanta | 118010484 | Kernel, GPA Planner |
| Harry Tanamas | 118010501 | Calendar, Clock |
| Steven Sugihermanto | 118010499 | Notepad, Calculator |

Reflections

| Name | Reflections |
|----------------------|---|
| Adriel Abraham Intan | Problems and Solutions: <ul style="list-style-type: none">- Combining different objects into a single application. Solution: determining a master object that maintains the whole application (as the top parent). |

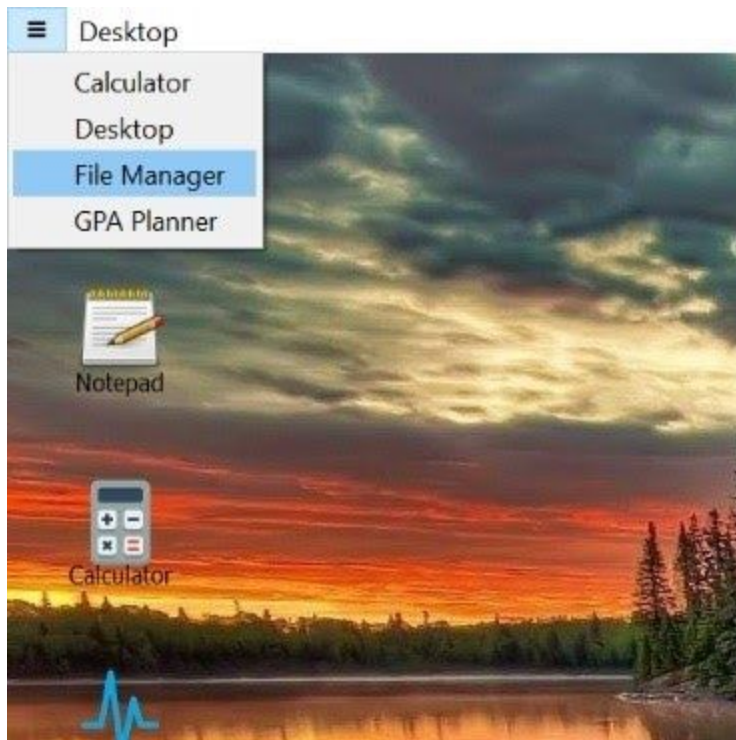
| | |
|-----------------|--|
| | <ul style="list-style-type: none"> - Copy files, including subfolders, in File Manager Solution: using recursion. - Designing GUI using Qt library Solution: read Qt documentation. - Passing informations between applications Solution: implementing a Global header as the storage for global variables <p>Lessons:</p> <ul style="list-style-type: none"> - I apply the classes and inheritance theory that is taught in the CSC3002 into this project. - I learnt that the simple tasks that are running in our computer (such as copying files) require a more complicated process (copying files require the OS to iterate through all the subfolders and copy them one by one to the target path). - I learnt that Object Oriented Programming slices a big project such as this project into smaller and easier-to-manage projects. |
| Brandon Honanta | <p>For me the main difficulty was actually the C++ language that was used. It is true that many things are taught in the class and that we had 3 assignments. But, I'd say this project has improved my fluency in writing C++ codes by one level. Moreover, this application introduced me to new data types such as QString, QDir, and some constraints that for example we cannot use any if statements outside of a function or class.</p> <p>The way I overcome these challenges are reading code documentations of QT widgets, looking for answers in stack overflow for the bugs that I made, and also asking my team members (Adriel, Steven, and Harry). I found it very helpful to organize this kind of study group as we are working for the same cause, not only for our grade, but also for improving our skills as CS students.</p> |

| | |
|---------------------|---|
| | <p>Reflecting back, I'd say that the classroom has equipped me with the critical foundational knowledge of using C++. Doing this project would be much harder without the help of the lectures and tutorials. But, I'd also say that what I've learned in this project is as much as what I learned in the class, that is a lot! For example, creating functional QT widgets are self-taught from YouTube and asking team members. All in all, I'd say this is not an easy course, but I'm glad to take it.</p> |
| Harry Tanamas | <ul style="list-style-type: none"> - Getting familiar with Qt widgets programming as it was never taught to us before. I search for information and learn about this from the internet. - Figuring out how to store the events specified by the user for each date and time on the calendar. I used the multimap class to save the data. Multimap is similar to the map class taught in the class. - Connecting different features and functions of the application so that it can run without problems and errors. |
| Steven Sugihermanto | <p>Combining the application with the main desktop is one the most difficult problems that we face in the project. Group communication is needed to solve this problem. Because in the end we all need to compile our own individual program into one. Also, by communicating we found out that most of the errors that exist have been resolved by other teammates therefore, it speeds up the whole process.</p> <p>A lot of new things are learnt through this project, such as learning the correlation between the UI and the function so that for e.g. a save button would save a file if it is pressed. A deeper understanding on C++ and Qt Widgets. And also, by combining</p> |

| | |
|--|---|
| | knowledge from other courses, I begin to comprehend the underlying foundation on how our computer interacts between hardware, software, and the operating system. |
|--|---|

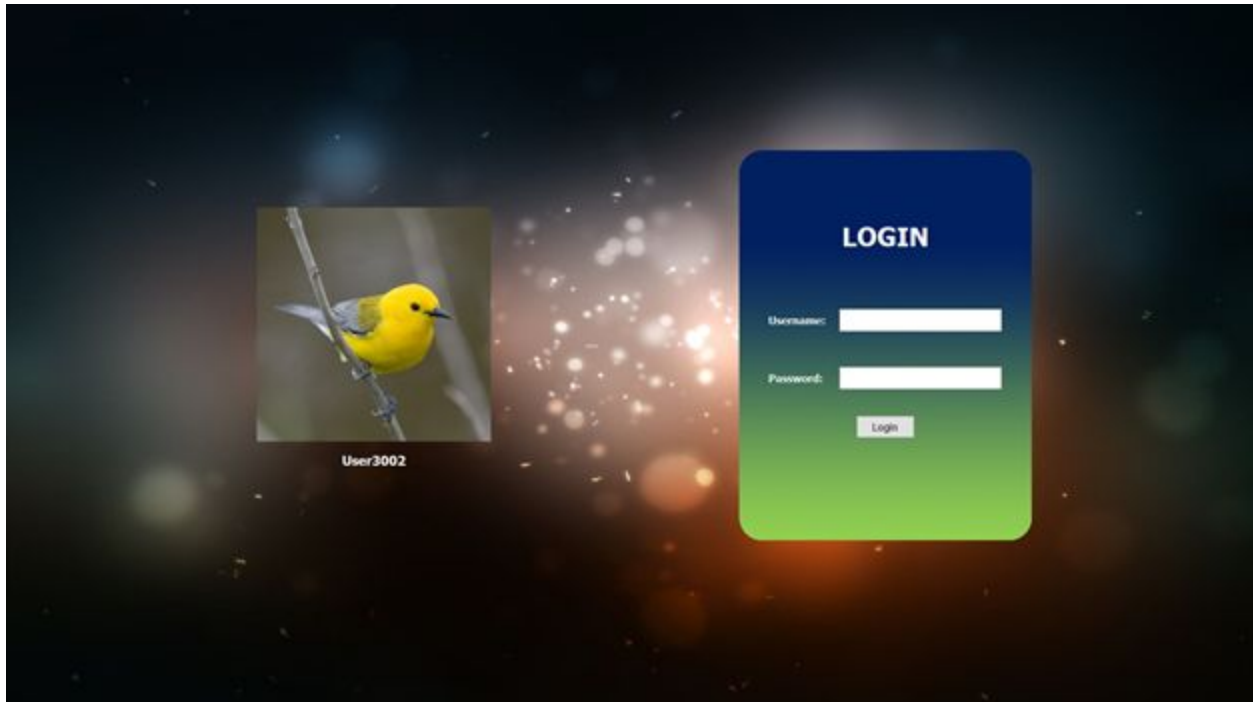
All Applications (including Desktop)

- The three lines in the top left corner of the screen will show all the currently opened applications when clicked. User can easily move to a different application by clicking the desired application.



- In the top right corner, the current time and date are shown. When the user clicks on the time, the application will open the Clock application, and when the user clicks on the date, the application will open the Calendar application.
- To close an application (except Desktop), the user only needs to click on the application name and choose close. Then, the user will be directed to the Desktop.

Login Page



When entering BASH OS, the user has to wait for some time for the OS to load, then the user will be faced with the login page. In the login page, the user has to input the correct Username and Password before accessing the OS. This feature prevents strangers from accessing the OS and improves the security of the OS.

Username: Admin

Password: CSC3002-2020

Desktop



In the Desktop, the user can directly open the desired application by left-clicking the application icon twice. When the user click on the “Desktop”, there will be three choices:

- About System

Clicking this option will open the “About System” application.

- Log Out

This option will bring the user back to the Login Page.

- Shut Down

This option will turn off the BASH Operating System (or in other word, closes the whole application).

About System

■ About System

Mon May 25 2020 21:12:10



Bash Edition: Bash 7.3.0
System Type: x86_64
Computer Name: LAPTOP-GRPIV842

Opening this application will show the current status of the user PC (including the edition of the OS, the type of PC system, and the Computer Name).

Calculator

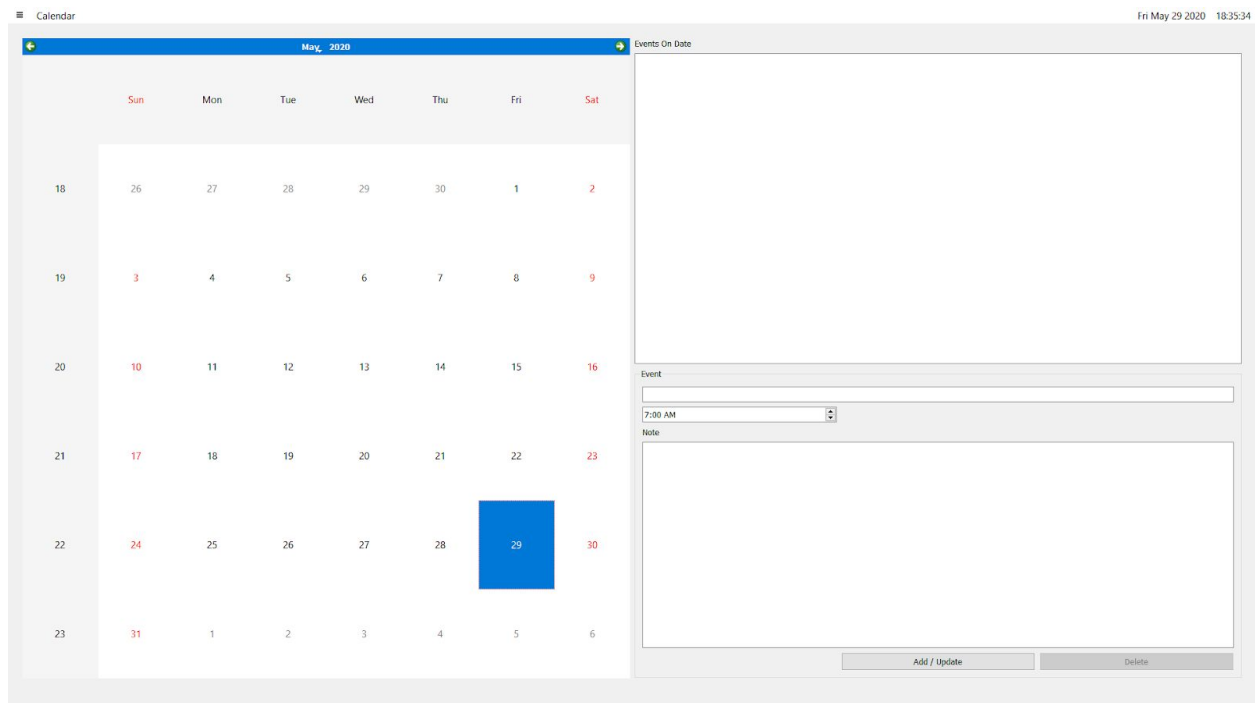
■ Calculator

Mon May 25 2020 20:50:37

| | | | | |
|----|---|-----|---|----|
| 0 | | | | |
| 7 | 8 | 9 | / | M+ |
| 4 | 5 | 6 | * | M- |
| 1 | 2 | 3 | + | M |
| AC | 0 | +/- | - | = |

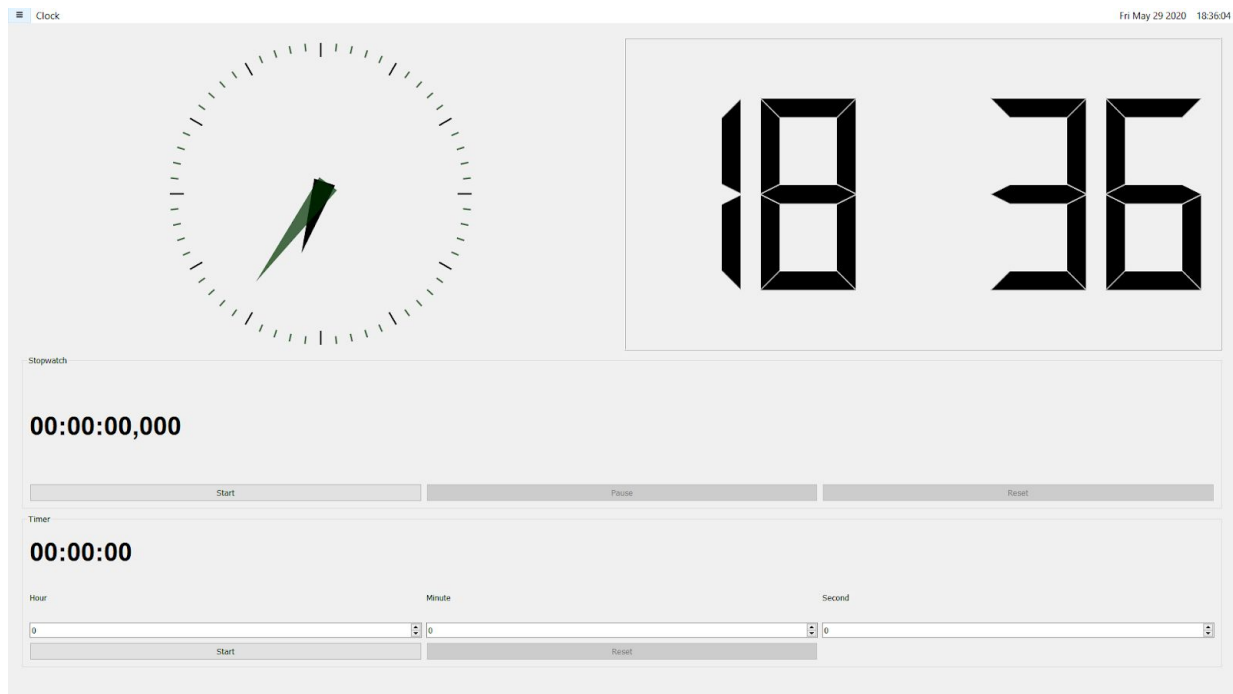
Calculator is used to perform arithmetic operations on numbers, we decided to put a calculator inside the OS because calculator is one of the most used applications in a computer. This calculator can only perform simple calculations such as addition, subtraction, multiplication. The idea of the UI is taken from ios calculator design.

Calendar



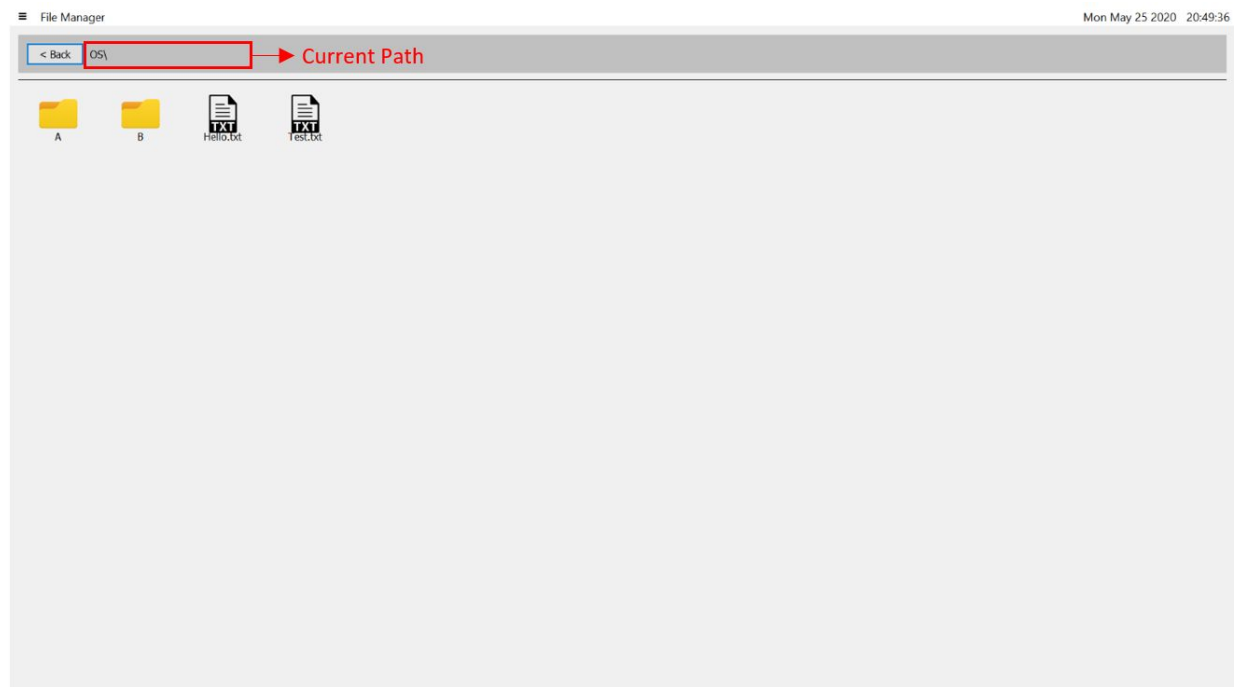
Once opened, the calendar application will show today's date. Using this application, the user can add events on any date displayed in the calendar application. After adding events to the date, when the user clicks the date, the application will display any events along with its details.

Clock



The clock application shows the current time in an analog clock and digital clock. The application is also equipped with a stopwatch and a timer function.

File Manager



The File Manager application explores all the folders and files (txt file) in the user's computer. To open a folder or file, the user only needs to double click on the icon of the desired file/folder. If the user wants to go back to the previous folder, the user only needs to click on the back button to access the previous folder. This file manager supports the creation of new folders, renaming file/folder, moving and copying files/folders, and deleting files/folders. To copy, move, or delete files/folders, the user only needs to select the files/folders in the current path, and click “File Manager”, then choose the desired action.

Kernel

The Kernel that our team has implemented works identically with command prompt in windows. The functions that we have implemented are:

| Usage | Syntax |
|---|--------------------------------|
| Peek all folders or files present inside current folder | dir |
| Change directory | cd [foldername] |
| Change to the parent folder of current folder | cd .. (with space before “..”) |
| Make directory | md [foldername] |
| Remove directory | rd [foldername] |
| Output string | echo “some string” |
| Opening text file present in current folder | filename.txt |


```
Kernel
baseFolder> dir
A
B
Hello.txt
Test.txt
baseFolder> cd A
baseFolder/A>
```

GPA Planner

This is an application that helps exclusively CUHKSZ students to find letter grade combinations to achieve the GPA target they want to achieve. It is said to be exclusive because the letter grade system is taken from CUHKSZ with A being 4.0, A- 3.7, B+ 3.3, B 3.0, and so on.

Here is how to use it:

- 1) Input the target GPA in the top left part of the program.



The screenshot shows the 'GPA Planner' application window. At the top left, there is a header 'GPA Planner'. Below it, there is a label 'Target GPA:' followed by a text input field containing the value '3.0'.

- 2) Input the course names with their corresponding credits (must be integers) and GPA.

| | Course Name | Credits | Letter Grade |
|----|-----------------------|---------|--------------|
| 1) | CSC3002 | 3 | B |
| 2) | Drawing Course | 2 | Plan |
| 3) | Conversation Speaking | 1 | B- |
| 4) | Badminton | 2 | Plan |
| 5) | | | A |
| 6) | | | A |
| 7) | | | A |
| 8) | | | A |

- 3) Press Recommend Me at the bottom most of the program.

Clear

Recommend Me

- 4) The combinations that suffice the target GPA would be shown in the top right portion of the program.

The recommendation will be displayed here:

Your max attainable gpa is 3.462500

| | Drawing | Badminto |
|----------|---------|----------|
| 3.037500 | C+ | A |
| 3.062500 | B- | A- |
| 3.137500 | B- | A |
| 3.037500 | B | B+ |
| 3.137500 | B | A- |
| 3.212500 | B | A |
| 3.037500 | B+ | B |

- 5) The Reset bottom can also be used at any time.

Notepad

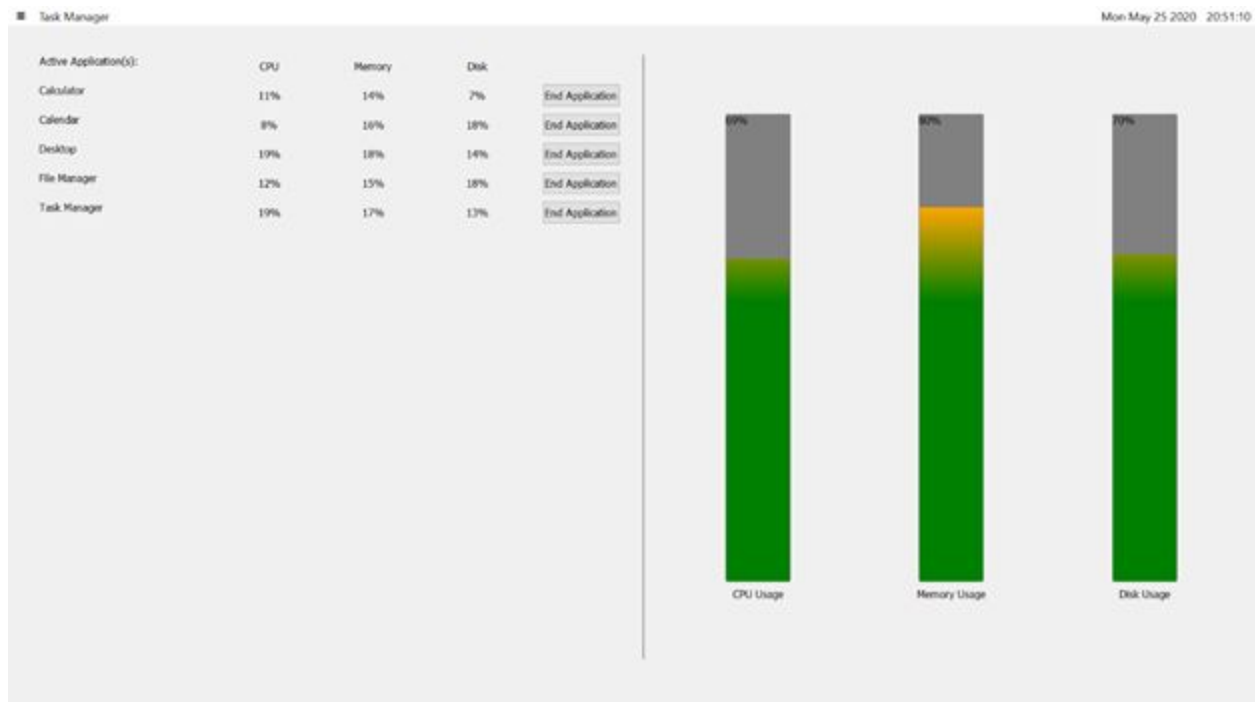
Notepad Edit

Mon May 25 2020 20:50:24

Similar to the native notepad application, our team has implemented a fully working notepad application inside the fake operating system. This notepad application supports the following command:

| Command | Usage |
|----------------|---|
| New | Create a new and empty text file |
| Save | Save or update the lastly preserved text file into the file manager using the tag (OS\...) |
| Save As | Save a new file or to an existing text file into the file manager to a new location with the same name or a different name using the tag (OS\...) |
| Close | Close the notepad application |
| Copy | Copy a selected item inside the text file and put it into a clipboard without deleting the item in the text file |
| Paste | Paste an item from the clipboard and display it into a new location inside the text file |
| Cut | Cut a selected item inside the text file and put it into a clipboard while deleting the item in the text file |
| Undo | Undo the last action the user does in the text file |
| Redo | Redo the last action the user undo in the text file |

Task Manager



The Task Manager application shows all the currently opened applications along with their CPU, Memory (RAM), and Disk (HDD) usages. From this application, the user can easily close any application to free some CPU, Memory, and/or Disk space. To close an application, the user simply needs to click the “End Application” button beside the desired application.