HW_02_Resha.md

HW 02 Resha

CSc 139

1. Characterize high-level the original Unix Os design. State novel designs and SW design goals. List advantages. How relevant is it today? Who were original developers? Initial Development motivation?
    - Originally designed to be a convenient platform for software developers.
    - Didn't start as portable or multi-tasking capable, but grew into both as C became standardized for hardware
    - Novel Designs/Goals:
        - Meant for each function to one thing, but do it well
        - Worked to standardize several formats to make it more universal
            - Including a file system standard
    - How Relevant today?
        - Very relevant, original OS no longer in use but nearly all modern operating systems are based on UNIX
    - Original Developers:
        - AT&T Bell Labs
    - Initial Dev Motivation
        - To be used in the Bell System for computing
2. Name and briefly outline some ideal high-level OS design goals.
    - Convenient, easy to use
    - Reliable, safe
    - Fast
    - Easy to implement, design, and maintain
    - Backwards compatible for upgrades
    - Secure
    - Portable
    - Multi-user
3. Write a concise description of the C/C++ function `fflush()`.
    - Takes everything from buffer (output stream only) and throws it. Main purpose is to move to data/console/disk and clear the current output.
4. Make a concise description of the Unix/Linux OS command `whoami`. Also try `WHOAMI` on windows, describe result.
    - Returns the username of the current user. Gives a large amount of files when used on windows, normally used on Unix
5. Write a C++ program `system1` that calls the library function `system()`. Issue 3 distinct calls to `system()` with a single argument from: `ls`, `pwd`, and `whoami`. Show source and generated outputs. Describe.

```
#include <stdio.h>
#include <stdlib.h>

int main(void) {
    system("ls");
    system("pwd");
    system("whoami");
    return 0;
}
```

Output:

```
hw01.md
HW_01_Resha.pdf
HW_02_Resha.md
hw2_num5
hw2_num5.cpp
mainNotes.md
/home/alec/notesRepo/fall2021/CSc139
alec
```

6. Write the C or C++ program `system2` that reads, when being executed, OS commands via the “command line parameter” C/C++ feature, and then executes them. The command line parameters must be legal Unix/Linux commands. Print the number of commands entered. To prepare, read about `argc` , `argv` , and `envp` . Focus is only argc and argv.

```
#include <iostream>
using namespace std;

int main(int argc, char** argv)
{
    cout << "Number of arguments: " << argc << endl;

    for (int i = 1; i < argc; ++i)
        system(argv[i]);

    return 0;
}
```

CLI: ./num6 pwd ls whoami
Output:

```
Number of arguments: 4
/home/alec/notesRepo/fall2021/CSc139
hw01.md
HW_01_Resha.pdf
HW_02_Resha.md
hw2_num5.cpp
hw2_num6.cpp
mainNotes.md
num6
alec
```

7. What does API stand for? What is an API? What does the API define?
    - Application Programming Interface
    - An API allows two programs to talk to eachother, for example the OS on your phone calling a weather API to get the weather.
    - An API defines routines for sending and receiving information such as a getWeather() function to retrieve weather information.
8. What are 5 general key functions and responsibilites of an OS?
    - User Inteface
    - Resource Management
    - Task Management
    - Files Management
    - Utility Functions
9. OS commands can be purely textual, or completely graphical. Explain pros and cons. Name a sample OS for both types.
    - Textual Pro's:
        - More flexibility, can chain command together, universal use
    - Textual Con's:
        - Harder to use and steeper learning curve.

- Graphical Pros:
  - Easier to use, can be repeated easily
- Graphical Cons:
  - Much less flexibility (stuck with what has been implemented graphically), slower.

10. What is the meaning of `Information Hiding` ? How is this related to OS?
    - Splitting the data or information that should be static from areas that change often
    - Hides information from specific users if they don't have access to it
    - OS relation: OS will hide data from other users, hides critical system files, etc
11. Briefly contrast ages, origins, and uses of MS-DOS, Unix BSD, and Linux.
    - MS-Dos
      - Made in 1981
      - First made by Microsoft for personal computers
      - Mainly used in personal computes in the 80's, did not last too long since it didn't have a gui
    - Unix BSD
      - Made in 1977
      - Made at University of Berkeley
      - The major use was variants based on it in workstations and other computing
    - Linux
      - Made in 1991
      - Kernal made by Linus Torvalds
      - Made primarily because of an OS that was restricted to education use only. Created to be an open source OS kernal. It can be used on any device as long as it has a compatible license, entirely free. Used in computers, servers, etc.
12. Write a description of the C++ function `system()` .
    - Allows a C/C++ program to run terminal commands from the command (See number 5 and 6)
    - Very resource inefficient and depends on the specific OS terminal