

ForNextDay() Lecture 1

Von Neumann Architecture Paragraph

The von Neumann architecture is a computer architecture in which a processing unit that contains an arithmetic logic unit (ALU) and processor registers, a control unit that contains an instruction register and program counter, a memory that stores data and instructions, an external mass storage, and an input and output are all present. This architecture allows for faster computing because it separates memory and storage into different components. Furthermore, it makes programming easier because it allows for iterative calculations to be made. When a programmer needs to repeat a calculation multiple times, he or she can use loops that iterate through different sections of code without having to copy and paste individual lines repetitively.

The Stored Program Concept Paragraph

The stored program concept, which was first introduced by Jon Von Neumann in the late 1940s, is a concept that refers to the storage of instructions in computer memory to enable it to perform a lot of different tasks in sequence or intermittently. The stored program concept allowed computers to execute instructions from a memory instead of from a hard drive. As computers became more advanced, the stored programs concept was used to store a larger amount of data, and in doing so to increase performance. In the 1990s it was adapted to support a variety of languages, including Java, C, and C++.

first.c code

```
// first.c
#include <stdio.h>
#include <stdlib.h>
int main(int argc, char * * argv)
{
    int a = 5;
    int b = 17;
    printf("main: a = %d, b = %d, argc = %d\n", a, b, argc);
    return EXIT_SUCCESS;
}
```

Terminal Activities

Signing in with sftp to upload first.c

```
C:\Users\momou\Desktop\University\CS 2263\ForNextDay\ForNextDay() 1>sftp mmoustaf@gc112m38.cs.unb.ca
mmoustaf@gc112m38.cs.unb.ca's password:
Connected to gc112m38.cs.unb.ca.
sftp> ls
1103_Lab1.txt          CS1073          CS1083
Desktop               Downloads       INFO1103
Info1103_Lab2.odt     MahmoudMoustafa_As5_Report.odt  cprogram
eclipse-workspace    untitled_0.odt
sftp> mkdir cs2263
sftp> cd cs2263
sftp> mkdir forNextDay
sftp> cd forNextDay
sftp> put first.c
Uploading first.c to /home1/ugrads/mmoustaf/cs2263/forNextDay/first.c
first.c                100% 203    4.3KB/s   00:00
sftp> █
```

Signing in with ssh to Compile first.c

```
C:\Users\momou\Desktop\University\CS 2263\ForNextDay\ForNextDay() 1>ssh mmoustaf@gc112m38.cs.unb.ca
mmoustaf@gc112m38.cs.unb.ca's password:
Last login: Thu Sep  9 12:18:27 2021 from 10.6.104.34
[mmoustaf@gc112m38 ~]$ cd cs2263/
[mmoustaf@gc112m38 ~/cs2263]$ cd forNextDay/
[mmoustaf@gc112m38 forNextDay]$ ls
first.c
[mmoustaf@gc112m38 forNextDay]$ gcc -o mine first.c
[mmoustaf@gc112m38 forNextDay]$
```

Running with multiple words

```
[mmoustaf@gc112m38 forNextDay]$ gcc -o mine first.c
[mmoustaf@gc112m38 forNextDay]$ ./mine
main: a = 5, b = 17, argc = 1
[mmoustaf@gc112m38 forNextDay]$ ./mine one two
main: a = 5, b = 17, argc = 3
[mmoustaf@gc112m38 forNextDay]$
```

Explanation of the printf statement

The printf statement is producing a terminal output saying “main : a = 5, b = 17, argc = <a changeable number>” argc is the number of arguments given to the program while running it using the “./”. a = 5 and b = 17 because their values have been explicitly stated (hard coded) in the program. argc changes based on how many arguments are present after the “./”. So, the printf statement prints the hard coded message, along with the values of variables (changing or hardcoded) thanks to the “%d”. Ps. Adding extra spaces between arguments doesn’t make any difference, when there is a space between them.