Assignement-1

CS-6002 Advance Operating Systems

Due Date: 15th Feburary 2019 (Except part-A)

Part A) Online course of "**Linux Basics: The Command Line Interface**" must be completed before 28th Feburary 2019. You should register this course in Audit mode and finish all of the quizzes and send me the screen shot of the final page.

https://courses.edx.org/courses/course-v1:Dartmouth_IMTx+DART.IMT.C.06+2T2018/course/

Part B) Download the latest stable kernel source and compile it. You can use following links to guide you

https://www.cyberciti.biz/tips/compiling-linux-kernel-26.html

https://www.linux.com/learn/intro-to-linux/2018/4/how-compile-linux-kernel-0

Part C) Understand Pointers, Stack & Heap Memory, malloc: When I run the following code on PC.

```
#include <stdio.h>
#include <stdlib.h>
int f(int i,int j, int *ptr);
int main (int argc, char *argv[]) {
  int a=2;
  int b=4;
  int *age = malloc(sizeof(int));
  *age =0;
  *age +=a;
  printf("Main Address of local variable a=%u,b=%u and *age=%u and
address of dynamic variable %u\n", (unsigned int)&a,(unsigned
int)&b,(unsigned int) &age, (unsigned int) age);
 f(a,b, age);

  return 0;
}

int f(int i,int j, int *ptr) {
  printf("Function f: address of local variable a=%u,b=%u and  address
of dynamic variable %u\n", (unsigned int) &i,(unsigned int)&j,(unsigned
int) ptr);

}
```

I get this output:

```
ubuntu@ubuntu-VirtualBox:~/code$ ./var0
Main Address of local variable a=3212861652,b=3212861656 and
*age=3212861660 and address of dynamic variable 158076936

Function f: address of local variable
a=3212861616,b=3212861620 and  address of dynamic variable
158076936
```

Run this code on your computer and explain this output. What these numbers mean, why they are like that and what can we learn from it about the position of the variables.

Read this article and try its example on your PC.

https://www.geeksforgeeks.org/memory-layout-of-c-program/

Part D) Compile the code in part-C with "-S" option into assembly and explain its outcome.

Example: compile var.c into var.asm assembly

```
ubuntu@ubuntu-VirtualBox:~/ gcc -S var.c -o var.asm
```