UiT The Arctic University of Norway

INF 3201 – Assignment Preparations

Yigit Can Dundar
yigit.c.dundar@uit.no

Logging into the cluster

- Login to the cluster with the command:
  - ssh abc123@uvcluster.cs.uit.no
- abc123 is your UiT login
- On first login, change your password using the command: passwd
- The cluster runs the Bash shell
  - Look into bash commands: traversing directories, running compiled programs etc.
- If outside UiT campus, use UiT's VPN service (link)

C Essentials

- Start with a simple program
  - Including headers
  - Assigning variables
  - Running loops
  - Output text on console

Look into exercises and tutorials online if you need to

```c
#include<stdio.h>

int main(){

    int array[10];
    int i = 0;

    printf("Assigning values 1-10 to the array\n");
    for (i = 0; i < 10; i++){
        array[i] = i + 1;
    }

    printf("Elements of the array are:\n");
    for (i = 0; i < 10; i++){
        printf("%d\n",array[i]);
    }

    return 0;
}
```

C Essentials - Pointers

- A pointers value is another variable's address in memory
- Asterisk (*) character is used to declare a pointer
- You can access a variable's address using "&" and assign it to a pointer
- You can then access the variable's value through the pointer
- Commonly used when a function needs to change the value of a variable outside its scope
- A tutorial on C pointers: link

```c
#include<stdio.h>

int main(){

    int var = 42;

    printf("Adress of var is: %x\n", &var);
    printf("Value of var is: %d\n", var);

    int *ptr;

    ptr = &var;

    printf("Address of ptr is: %x\n", ptr);
    printf("Value of ptr is %d\n", *ptr);

    return 0;
}
```

Output:

```
Adress of var is: 55dae5c
Value of var is: 42
Address of ptr is: 55dae5c
Value of ptr is 42
```

C Essentials – Memory Allocation/Deallocation

- Use malloc() to dynamically assign memory to a pointer
  - The function needs a cast type:
    - For int: (int*)
    - For float: (float*)
    - Etc…
  - The function needs a size value in bytes as its parameter
    - For example “10 * sizeof(int)” will allocate 40 bytes of memory
- You can free the allocated memory using free()

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

int main(){

    int *ptr;

    int i = 0;
    int n = 10;

    ptr = (int*)malloc(n * sizeof(int));

    for(i = 0; i < n; i++){
        ptr[i] = i + 1;
    }

    printf("Elements of the array are:\n");
    for(i = 0; i < n; i++){
        printf("%d\n",ptr[i]);
    }

    free(ptr);

    return 0;
}
```

Move files/directories to the cluster

- You can use scp to move files from your local pc to the cluster (and vice versa)
- To move a single file:
  - scp myfile abc123@uvcluster.cs.uit.no:/home/abc123/
- To move a directory:
  - scp –r Desktop/myDir abc123@uvcluster.cs.uit.no:/home/abc123/

Suggested downloads/installs

HIGHLY recommend you work with Linux! (Easy solution is to run Linux through a virtual machine like VMware Workstation Player)

- gcc
  - Linux: Run command “sudo apt-get install gcc”
  - Windows: WSL, Cygwin, mingw
  - MacOS: Guide
- Python 2.7 (and/or 3)
  - Linux: Run command “sudo apt-get install python”
  - Windows: Link
  - MacOS: Link

No need to install any of these on the cluster!