WISDOM OYOR

215206

CSC235 ASSIGNMENT

1. Write briefly on Unix operating system especially Linux flavor

The Unix operating system is a set of programs that act as a link between the computer and the user. It allocates the system resources and coordinates all the details of the computer's internals. The operating system or the kernel, was originally developed at AT & T Bell Laboratories.

Users communicate with the kernel through a program known as the shell. The shell is a command line interpreter; it translates commands entered by the user and converts them into a language that is understood by the kernel.

UNIX is called the mother of operating systems which laid out the foundation to Linux. **Linux** is an operating system or a kernel distributed under an open-source license.

The **commands used on both the operating systems are usually the same.** There is not much difference between UNIX and Linux. Though they might seem different, at the core, they are essentially the same, since **Linux is a clone of UNIX**.

1. Write a short note on software functional requirements

A software **Functional Requirement** (FR) is a description of the service that the software must offer. It describes a software system or its component. A function is nothing but inputs to the software system, its behavior, and outputs. It can be a calculation, data manipulation, business process, user interaction, or any other specific functionality which defines what function a system is likely to perform. Functional Requirements in Software Engineering are also called **Functional Specification**.

In software engineering and systems engineering, a Functional Requirement can range from the high-level abstract statement of the sender’s necessity to detailed mathematical functional requirement specifications. Functional software requirements help you to capture the intended behavior of the system.

1. Why is Unix often preferred at some points?

UNIX is a powerful, multi-user, and multitasking operating system. It is very popular among the scientific, engineering, and academic due to its most appreciated features like flexibility, portability, network capabilities, etc.

Many developers find it's a refreshing alternative to the monolithic tools like IDEs and languages like Java.

UNIX operating system supports **Multiprocessing**, meaning many processes are executed simultaneously. In Multiprocessing, every process has a separate address space and CPUs can be added for increasing computing power.

1. Why is Unix referred to as a Scientist OS?

Unix operating systems GUIs are relatively harder to grasp and produce significant barriers for newcomers. But this complexity comes with greater level of control over files. Many programmers, developers and scientists tend to choose Unix/Linux OS over the other OSes because it allows them to work more effectively and quickly. It allows them to customize to their needs and be innovative. Linux allows programmers and scientists to freely alter, utilize, distribute, and publish their development.

1. What type of programming language is C?

C is a structured, procedural programming language that has been widely used both for operating systems and applications and that has had a wide following in the academic community.

1. Give the detailed structure of a complete C programming language

A C program is divided into six sections: Documentation, Link, Definition, Global Declaration, Main() Function, and Subprograms. While the main section is compulsory, the rest are optional in the structure of the C program.

**Documentation**

Consists of the description of the program, programmer's name, and creation date. These are generally written in the form of comments.

**Link**

All header files are included in this section which contains different functions from the libraries. A copy of these header files is inserted into your code before compilation.

**Definition**

Includes preprocessor directive, which contains symbolic constants. E.g.: #define allows us to use constants in our code. It replaces all the constants with its value in the code.

**Global** **Declaration**

Includes declaration of global variables, function declarations, static global variables, and functions.

**Main()** **Function**

For every C program, the execution starts from the main() function. It is mandatory to include a main() function in every C program.

**Subprograms**

Includes all user-defined functions (functions the user provides). They can contain the inbuilt functions and the function definitions declared in the Global Declaration section. These are called in the main() function.

1. How can I create a C programming file on the OS

On Linux:

1. Open terminal

2. Create the file using this command:

nano sample.c

(file name can be anything but it should end with dot c extension)

3. Open file with the vim editor using this command:

vim sample.c

4. To Edit the file:

Press i to go to insert mode.

Type your program.

5.To save the file:

Press Esc button and then type: wq

It will save the file.

1. Write a program in C to say Hello World to your lecturer

#include <stdio.h>

int main() {

printf("Hello world");

return 0;

}

1. Write a C program to accept the names of 10 students

#include <stdio.h>

#include <string.h>

int main()

{

//Declare Variables

char string[10][30]; //2D array for storing strings

int n = 10;

int i;

//Read the string from user

printf("Enter names one by one: \n");

for(i=0; i< n ; i++) {

scanf("%s",string[i]);

}

//Print all the names

printf("The names are: \n");

for(i=0; i< n ; i++) {

//Print the string at current index

printf("%s \n", string[i]);

}

//Return to the system

return 0;

}

1. Write a C program to count 1-N

#include <stdio.h>

#include <string.h>

int main()

{

int i, n;

//Get the max count

printf("Enter number to count to: \n");

scanf("%d", &n);

//Print the count from 1-N

printf("Counting to ");

printf("%d: \n", n);

for(i=1; i<=n; i++) {

//Print each number

printf("%d\n", i);

}

//Return to the system

return 0;

}

1. Write a C program that reverses a sentence

#include <stdio.h>

int main()

{

char sentence[1000], reverse[1000];

int begin, end, count = 0;

printf("Input a string\n");

gets(sentence);

// Calculating string length

while (sentence[count] != '\0')

count++;

end = count - 1;

for (begin = 0; begin < count; begin++) {

reverse[begin] = sentence[end];

end--;

}

reverse[begin] = '\0';

printf("%s\n", reverse);

return 0;

}

1. Write a program in C to solve a simple quadratic equation (x2 + y + c = 0). Your program you allow the use of any number as the coefficient of any of the variable.

#include <math.h>

#include <stdio.h>

int main() {

double a, b, c, discriminant, root1, root2, realPart, imagPart;

printf("Enter coefficient a: ");

scanf("%lf", &a);

printf("b: ");

scanf("%lf", &b);

printf("c: ");

scanf("%lf", &c);

discriminant = b \* b - 4 \* a \* c;

// condition for real and different roots

if (discriminant > 0) {

root1 = (-b + sqrt(discriminant)) / (2 \* a);

root2 = (-b - sqrt(discriminant)) / (2 \* a);

printf("root1 = %.2lf and root2 = %.2lf", root1, root2);

}

// condition for real and equal roots

else if (discriminant == 0) {

root1 = root2 = -b / (2 \* a);

printf("root1 = root2 = %.2lf;", root1);

}

// if roots are not real

else {

realPart = -b / (2 \* a);

imagPart = sqrt(-discriminant) / (2 \* a);

printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", realPart, imagPart, realPart, imagPart);

}

return 0;

}