**Chapter 10: Standard & Windows APIs**

**Topic – 1: Standard Library**

**Introduction**

* What we are going to study are from ***<stdlib.h>***.

**Commands To Terminal**

* It directly writes command to **OS terminal**.

***system("cd C:\\path\\to\\directory && filename.lnk");***

* Each time ***system()*** is used, they start from **default** environment.
* Notice that ***.lnk*** is used as an extension for the file to be opened.

**Reading Terminal Outputs**

* Uses same **file management** functions like ***fgets()*** etc.
* Some custom functions for it specifically includes ***popen()*** & ***pclose()*** etc.

***FILE \*pipe = popen("dir", "r");***

***/\* Certain codes \*/***

***pclose(pipe);***

**Environment Variables**

***// Storing names of all environment variables in a string***

***char \*path = getenv("PATH");***

***// Set a custom environment variable, "MYVAR=" is compulsory to be written***

***int ret = putenv("MYVAR=MY\_ENVIRONMENT\_VARIABLE);***

**Topic – 2: Windows Graphics**

**Introduction**

* What we are going to discuss are kept in ***<windows.h>***.

**Setting Console Colours**

***HANDLE hConsole = GetStdHandle(STD\_OUTPUT\_HANDLE);***

***// For setting foreground colour (white here)***

***SetConsoleTextAttribute(hConsole, FOREGROUND\_RED | FOREGROUND\_BLUE | FOREGROUND\_GREEN);***

***// For setting background colour (white here)***

***SetConsoleTextAttribute(hConsole, BACKGROUND\_RED | BACKGROUND \_BLUE | BACKGROUND \_GREEN);***

**Notification Boxes**

***// MessageBox(Owner window, Message, Title, Error Type);***

***MessageBox(NULL, "Can not create window!", "Error", MB\_ICONERROR);***

***MessageBox(NULL, "Loading the file!", "User", MB\_ICONQUESTION);***

***MessageBox(NULL, "Overwrite the file?", "Error", MB\_YESNO);***

**Making Decision For MB\_YESNO**

***int res = MessageBox(NULL, "Message", "Title", MB\_YESNO);***

***if (res==IDYES) { /\*Some code\*/ }***

***else if (res==IDNO) { /\*Some code\*/ }***

**Providing Icon To MB\_YESNO**

***int res = MessageBox(NULL, "Message", "Title", MB\_YESNO | MB\_ICONQUESTION);***

**Topic – 3: Standard Definitions**

**Introduction**

* Members of structures may have ***memory padding*** between them.
* **Memory padding:** Reserved spaces for **better performance**.
* We are going to use ***<stddef.h>*** library which contains **advanced memory management** functions.

**Finding Offset**

* Function ***offsetof()*** tells how members are packed within a structure by returning their respective offsets.

***struct Example {char name[10], int age; float salary;}***

***printf("%zu\n", offsetof(struct Example, name)); // 0 (bytes)***

***printf("%zu\n", offsetof(struct Example, name)); // 12 (bytes)***

***printf("%zu\n", offsetof(struct Example, name)); // 16 (bytes)***

* ***%zu*** is format specifier for objects like ***size\_t***.
* ***size\_t*** is an **unsigned integer** type.

**Packing Members Close**

***struct \_\_attribute\_\_((packed)) Example {char name[10], int age; float salary;}***

***printf("%zu\n", offsetof(struct Example, name)); // 0 (bytes)***

***printf("%zu\n", offsetof(struct Example, name)); // 10 (bytes)***

***printf("%zu\n", offsetof(struct Example, name)); // 14 (bytes)***

**Warning!**

**🡪 It must be used only when one is working on one type of hardware.**

**🡪 As *\_\_attribute\_\_* is not portable over each hardware.**