C Programing

Trainer : Kiran Jaybhave

Agenda

- Introduction
- Linux File System Commands
- VIM editor
- C Programming
  - C History
    - Types of language
  - Hello World

Introduction

- Mr. Kiran G. Jaybhave
  - B.E. In Electronics and Telecommunication

- **Training**
  - CDAC Main Course Batches
  - Pre-CAT Batches

- **No Mobiles**
- **Daily Schedules**
  - **Lecture: 8.00 am to 1.00 pm**
  - **Lab: 2.00 pm to 7.00 pm**

❖DESD Course

- **C "Programming"** -- Foundation
- **Data structures and Algorithms** -- Data structure Programming Advanced
- **Micro-controllers** -- ARM Cortex-M, Interfacing, Protocols, Firmware Programming
- **Embedded Operating Systems** -- ARM Cortex-A, OS Concepts,
- **Linux Programming Linux Device Drivers** -- Driver Programming
- **Internet of Things** -- BBB, ESP8266, Python programming, Firmware Programming
- **Realtime OS** -- FreeRTOS, Realtime Task
- **Programming Circuit design/PCB design** -- Pre-requisite

- **DESD Evaluations**
  - **Each Module**
    - **Theory Exam** : MCQ : 40 marks -- CCEE
    - **Lab Exam** : Last day of the module : 40 marks -- One program
    - **Internal Eval** : On-going : Assignments, Quiz, Interview, etc.

Linux File System Commands

- **Basic Command**
- **Directory commands**
- **File commands**

Computer System

- **What is Computer ?**
  - Computer System is electronic device is integrated using Hardware and Software.
  - Hardware is tangible which helps to process data where as software is set of instructions which are executed with the help of hardware.

- **What is Program ?**
  - Program is set of instructions collected together in specific logical flow to completer particular task.

- **A program can be constructed using various program constructs like**
  - Sequence
  - Decision Control
  - Loop / Iteration

Classification of Languages

- **A Program can be written using :**
  - **Low Level Language**
    - Low level languages are the basic computer instructions or better known as machine codes.
    - The low level language is a programming language where machine instructions can be given in 0 and 1 form.
  - **High Level Language**
    - Make easy communication to computer system
    - Independent to particular type of computer
    - More close to human language than machine language.
    - Compiler helps to convert instructions to machine understandable form.
    - Takes additional time to translate the instructions to machine understandable instructions

Classification of Languages

❑ **Machine Level Language**

- The machine-level language is a language that consists of a set of instructions that are in the binary form 0 or 1.
- As we know that computers can understand only machine instructions, which are in binary digits, i.e., 0 and 1, so the instructions given to the computer can be only in binary codes.
  - Performance is good as we are directly writing the program on machine
  - Platform dependent
  - Difficult to program
  - Error prone

Assembly Language

❑Assembly Language

- The assembly language contains some human-readable instructions (consists of numbers, symbols, abbreviations), The language was introduced in 1952.
- As we know that computers can only understand the machine-level instructions, so we require a translator that converts the assembly code into machine code. The translator used for translating the code is known as an assembler.
  - Easier to understand and use and to locate errors
  - Platform dependent
  - Knowledge of hardware will help to program Machine level coding
  - Execution is slower compared to machine level language.

History

- C language was developed by Dennis Ritchie in 1972 at AT & T Bell Labs on PDP-11 machine.
- It was developed while porting UNIX from PDP-7 to PDP-11.
- Many features of C are inspired from B (Ken Thompson) and BCPL (Martin Richards).
- Initial release of C is referred as K & R C.

Standardization

- C was standardized by ANSI in 1989. This is referred as C89 .
- Standardization ensures C code to remain portable.
- Standard is revised multiple times to add new features in the language.
  - C89 – First ANSI standard
  - C90 – ANSI standard adopted by ISO
  - C99 – Added few C++ features like bool, inline, etc.
  - C11 – Added multi-threading feature.
  - C17 – Few technical corrections.

Features

- Data types
- Operators
- Control structures
- Functions
- Storage classes
- Pointers
- Arrays
- Strings
- Dynamic memory allocation
- Structure
- Union
- Enum
- File IO
- Preprocessor directives

Strengths

- Low level memory access (pointers, data structures)
- Effective memory access (bitwise operators, bit-fields, unions)
- Can access OS features (functions/commands)
- Extensive library functions (math, strings, file IO, …)
- Compilers for different platforms & architectures
- Highly Readable (macros, enum, functions, …)

Applications

❑ **System programming**

- OS development
- Device drivers
- System utilities

❑ **Embedded programming**

- ARM, AVR, PIC, etc.
- IoT development

❑ **Language development**

- Compiler development

❑ **Achievements (tiobe.com)**

- In top-2 languages in last 40 years.
- Language of year: 2019, 2017, 2008.

VI editor

- **Linux text editors for command line.**
  - VI editor
  - Emacs editor
- **VI is world's best editor (Linux CLI).**
  - VI developed by "Bill Joy".
  - In BSD(Barkeley Soft. Distrbution) UNIX OS UCB.
    - VIM = VI Improved.
- **VI modes**
    - Command mode Default mode
    - Press "**Esc**" to go in command mode.
    - Edit mode (Insert) Press **"i"** to go in Insert mode.
    - Edit the file.
- **VIM Commands**

Hello World

❑Source Code

- **Hello World program**

```
#include <stdio.h>
int main()
{
         printf("Hello World\n");
        return 0;
}
```

❑Commands

- **cmd gcc main.c -o main.out**
- **cmd> ./main.out**

Hello World

- stdio.h – header file
- printf() – library function
- main() – entry point function
  - void main() { … }
  - int main() { … }
  - int main(void) { … }
  - int main(int argc, char *argv[]) { … }
  - int main(int argc, char *argv[], char *envp[]) { … }

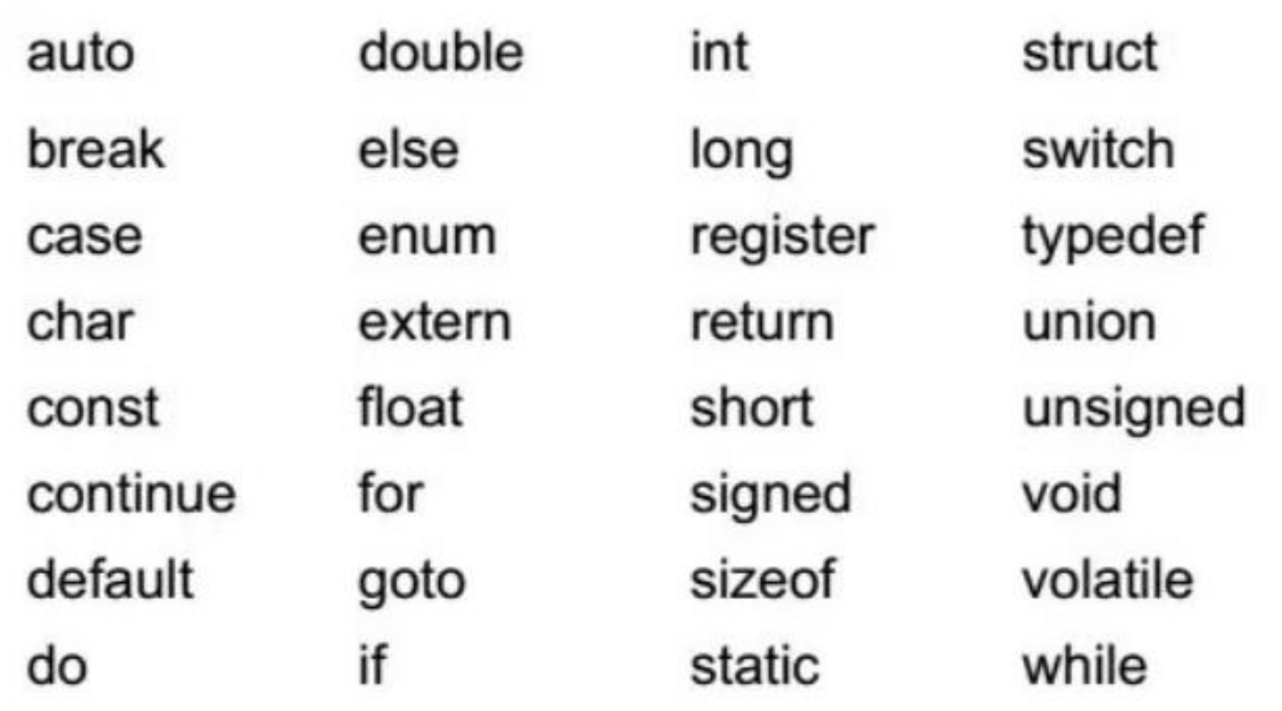
- return 0 – exit status

Tokens

- C program is made up of functions.
- Function is made up of statements.
- Statement contain multiple tokens.
    - Keywords
    - Data Types
    - Identifiers
    - Variables
    - Constants
    - Operators

Keywords

- Keywords are predefined words used in program, which have special meanings to the compiler.
- They are reserved words, so cannot be used as identifier.
- K & R C has 27 keywords. C89 added 5 keywords. C99 added 5 new keywords



auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Identifiers

- Helps to identify memory locations, functions, defined types and pre-processor macros.

- **Rules of Identifiers:**
  - Should start with alphabet or with _ (underscore)
  - Can include alphabets, _ (underscore),digits
  - Case sensitive

- **Examples:**
  - Var_1 //Valid
  - 1_var // Not Valid
  - _var //valid
  - Var-1 // invalid
  - Basic Salary //invalid

Data Types, Variables & Constants

- C allows computations to be performed on various types of data.
  - Numerical: Whole numbers, Real numbers
  - Character: Single character, Strings
- Fixed data values are said to be constants.
  - 12, -45, 0, 2.3, 76.9, 1.23456e+2, 'A', "Sunbeam", etc.
- Data is hold in memory locations identified by names called as variables.
  - Variable must be declared before its use in the program.
  - As per need, variable have some data type.
- Simple C data types are: int, double, char.
  - Data type represents amount of space assigned to the variable.
  - It also defines internal storage of the data.

Thank you!

Kiran Jaybhave

email – kiran.jaybhave@sunbeaminfo.com