
PG – DESD

Module – Embedded C Programming

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Mobile No - 9890662093



Passing arguments: Call by value vs Call by address/reference

- Call by value

- Formal argument is of same type as of actual argument.
- Actual argument is copied into formal argument.
- Any change in formal argument does not reflect in actual argument.
- Creating copy of argument need more space as well as time (for bigger types).
- Most of data types can be passed by value – primitive & user defined types.

- Call by address

- Formal argument is of pointer type (of actual argument type).
- Address of actual argument is collected in formal argument.
- Actual argument can be modified using formal argument.
- To collect address only need pointer. Pointer size is same irrespective of data type.
- Array and Functions can be passed by address only.



Pointer - Introduction

- Pointer is a variable that stores address of some memory location.
- Internally it is unsigned integer (it is memory address).
- In C, pointer is a special data type.
- It is not compatible with unsigned int.
- Pointer is derived data type (based on primitive data type).
 - To store address of int, we have int pointer.
 - To store address of char, we have char pointer, ...
- Size of pointer variable is always same, irrespective of its data type (as it stores only the address).



Pointer - Syntax

- Pointer syntax:

- Declaration:
 - `double *p;`
- Initialization:
 - `p = &d;`
- Dereferencing:
 - `printf("%lf\n", *p);`

- Reference operator - `&`

- Also called as direction operator.
- Read as “address of”.

- Dereference operator - `*`

- Also called as indirection operator.
- Read as “value at”.

```
int main() {  
    double a = 1.2;  
    double *p = &a;  
    double **pp = &p;  
    printf("%lf\n", a);  
    printf("%lf\n", *p);  
    printf("%lf\n", **pp);  
    return 0;  
}
```

- Pointer to pointer stores address of some pointer variable.
- Level of indirection: Number of dereference operator to retrieve value.



Pointer - Scale factor

- Size of data type of pointer is known as Scale factor.
- Scale factor defines number of bytes to be read/written while dereferencing the pointer.
- Scale factor of different pointers
 - Pointer to primitive types: `char*`, `short*`, `int*`, `long*`, `float*`, `double*`
 - Pointer to pointer: `char**`, `short**`, `int**`, `long**`, `float**`, `double**`, `void**`
 - Pointer to struct/union.
 - Pointer to enum.





Thank you!

Devendra Dhande <devendra.dhande@sunbeaminfo.com>

