**Pointers Important Notes**

* Arrays and pointer are essentially same. Difference is sizeof(array) returns size of the array and sizeof(poinetrs) returns pointer size which is size of integer on the system
* Pointers can be assigned to another pointer but array can’t be
* Array can’t be assigned with the pointer

Int a[]={10,20,30}

Int i=10

a=&i—not allowed

int \*ptr=a is allowed

\*

\*

Types of pointers \*

1.Data pointers

2.Function pointers\*

3.Void pointers-Dereferencing the pointer is invalid

Interesting point:

Linux kernel uses something like this\*

Struct{\*

\*

Void\* ptr;

Function pointers1\*

Function pointer2\*}—to mimic object paradigm

**Function Pointers**

void (\*fn\_ptr)(int, int)

fn\_ptr=function1;🡪no () required

int a,b;

fn\_ptr(a,b)

typedef void (\*fn\_ptr)(int, int)🡪makes it as data type fn\_ptr

fn\_ptr fn= function1;

**Void Pointer**

**Void\* ptr-🡪 can’t deferenced without typecasting**

**When NULL pointer(means pointing to 0x00 location) will throw segmentation faults usually when there is high level OS like windows, linux or MAC. It is allowed to use NULL in case of emebedded Bare metal or RTOS with Memory Management Unit disabled**