C Language

DMA

Saurabh Shukla (MySirG)

Agenda

1. SMA vs DMA
2. malloc()
3. Type Casting
4. calloc()
5. Memory Leak
6. free()
7. realloc()

SMA

Static Memory Allocation

```
int a;
float b;
double c;
char *p;
struct Book b1;
```

Declaration statements

DMA

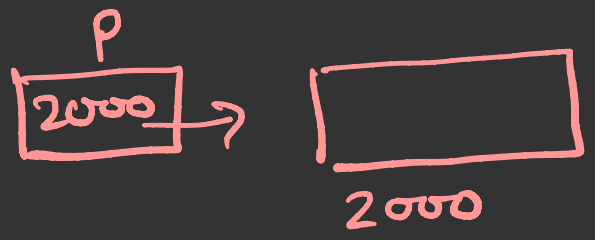
Dynamic Memory Allocation

malloc( )
calloc( )

DMA variables are not named. They only have addresses.

malloc()

```
int *p;
p = malloc(4);
```




```
void* malloc (     )
{
    ==
    ==
    return address;
}
```

int *p;

Type casting

```
p = (int *)malloc(4);
```

float *q; **calloc()**

q = (float *) calloc (5, 4);

*(q+0) q[0] = 10;

*(q+1) q[1] = 20;

*(q+2) q[2] = 30;

Malloc vs Calloc

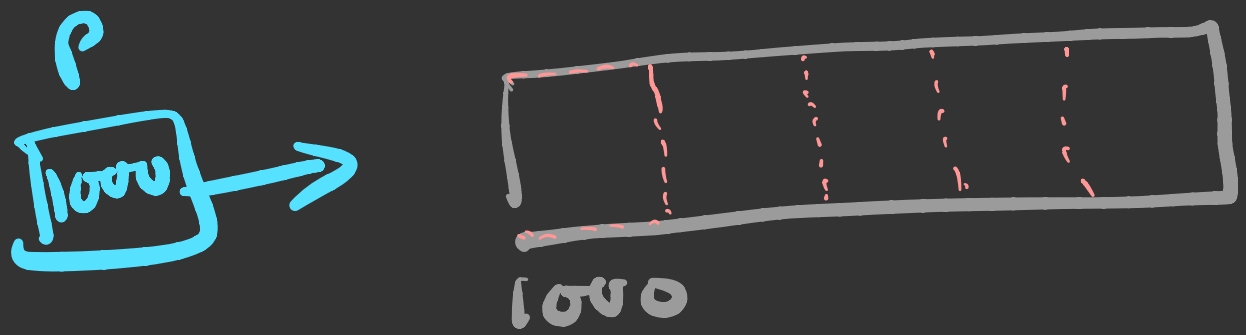
	Malloc	Calloc
1	one argument	two arguments
2	Garbage value	0 zero
3	Single block	Array of blocks

```
int *p;
p=(int*) malloc(20)
```

p[0]

*(p+0)

*(p+1)

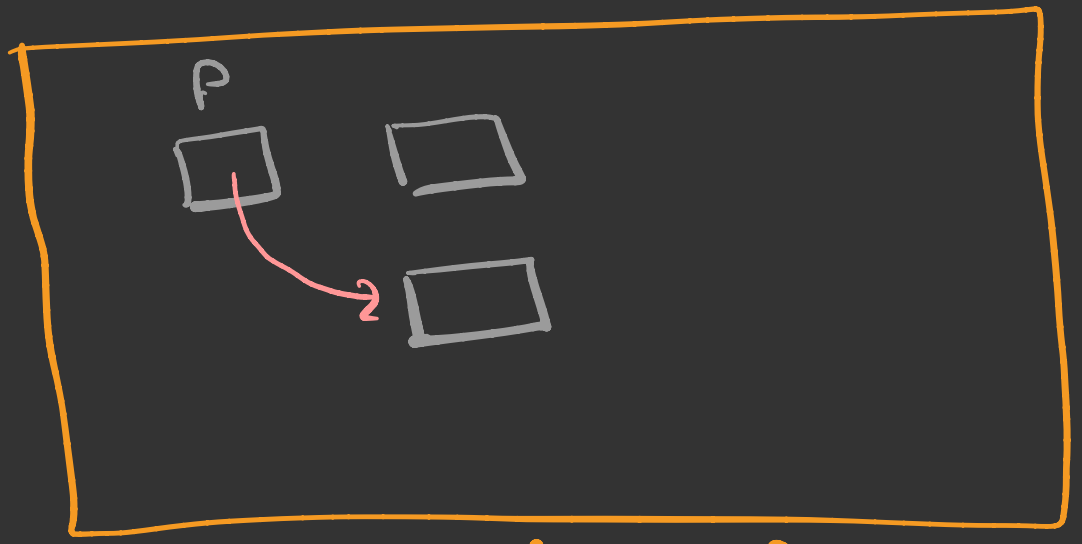


Memory Leak

```
int *p;
p = (int*)malloc(4);
```

≡

```
p = (int*) malloc(4);
```

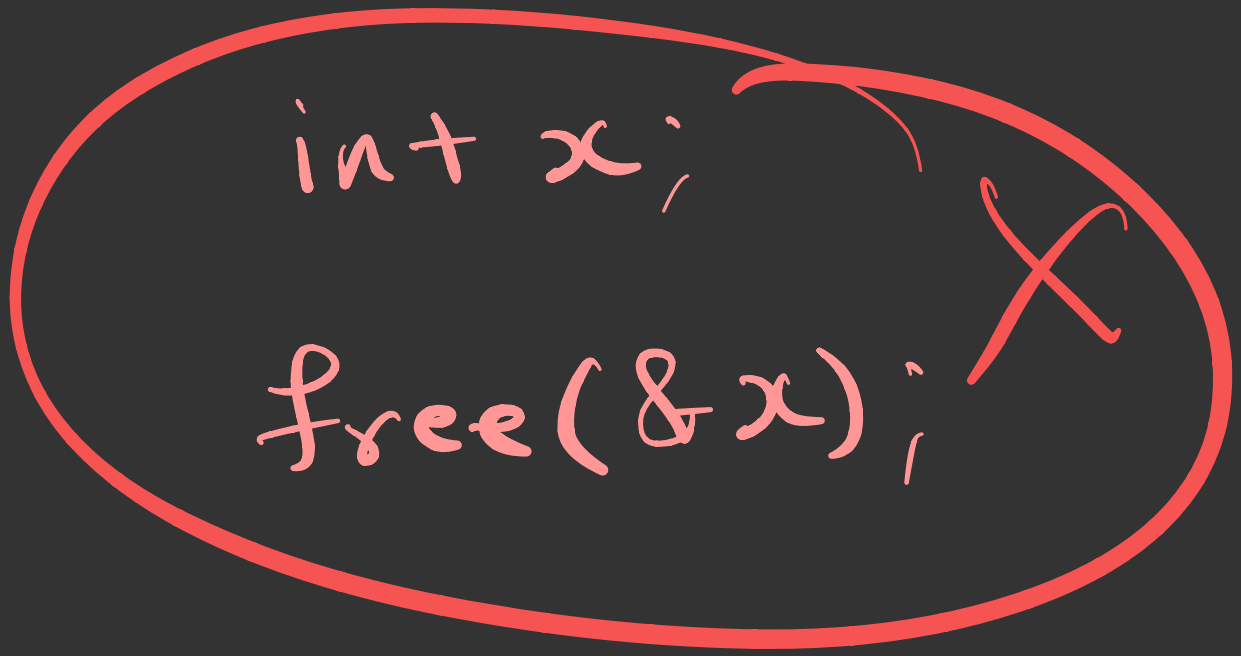


Total = Consumed + free

free( )

free function is used only to release Memory of DMA variables.

free( )



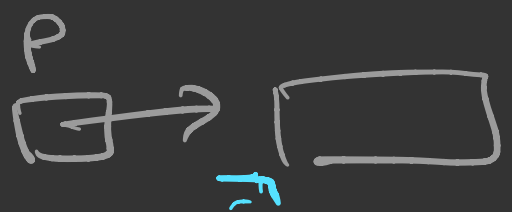
Two Options

① Return address of DMA variable

```
int *q;
q = f1();
```

free()

```
int* f1()
{
    int *p;
    p = (int*) malloc(4);

    return p;
}
```



Two Options

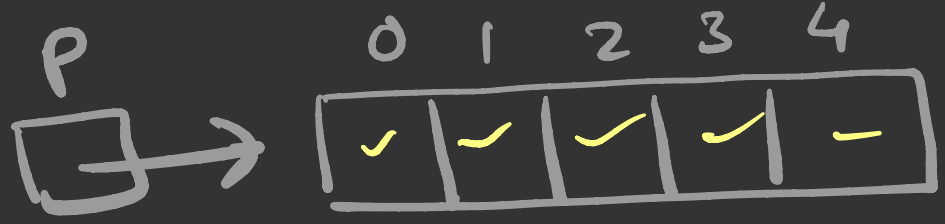
② free memory of DMA variable

```
f1()
{
    int *p;
    p=(int*)malloc(4);
    =
    =
    =
    =

    free(p);

}
```

realloc()

realloc(pointer, newsize)

```
int *p;
p = (int *) malloc(20);
```




```
p = realloc( p, 40);
```