PG – DESD

Module – Embedded C Programming

Trainer - Devendra Dhande

Email – devendra.dhande@sunbeaminfo.com

Mobile No - 9890662093

Union

- Union is user defined data-type.
- Like struct it is collection of similar or non-similar data elements.
- All members of union share same memory space i.e. modification of an member can affect others too.
- Size of union = Size of largest element
- When union is initialized at declaration, the first member is initialized.
- Application:
  - System programming: to simulate register sharing in the hardware.
  - Application programming: to use single member of union as per requirement.

```
union test {
    int num;
    char arr[2];
}u = { 65 };
printf("%d, %c, %s\n", u.num, u.arr[0],
u.arr);
```

Dynamic memory allocation

- Dynamic memory allocation allow allocation of memory at runtime as per requirement.
- This memory is allocated at runtime on Heap section of process.
- Library functions used for Dynamic memory allocation are
  - malloc() – allocated memory contains garbage values.
  - calloc() – allocated memory contains zero values.
  - realloc() – allocated memory block can be resized (grow or shrink).
- All these function returns base address of allocated block as void*.
- If function fails, it returns NULL pointer.

Memory leakage

- If memory is allocated dynamically, but not released is said to be "memory leakage".
  - Such memory is not used by OS or any other application as well, so it is wasted.
  - In modern OS, leaked memory gets auto released when program is terminated.
  - However for long running programs (like web-servers) this memory is not freed.
  - More memory leakage reduce available memory size in the system, and thus slow down whole system.
- In Linux, valgrind tool can be used to detect memory leakage.

```
int main() {
    int *p = (int*) malloc(20);
    int a = 10;
    // ...
    p = &a; // here addr of  allocated block is
lost, so this memory can never be freed.
    // this is memory leakage
    // ...
    return 0;
}
```

Dangling pointer

- Pointer keeping address of memory that is not valid for the application, is said to be "dangling pointer".
- Any read/write operation on this may abort the application. In Linux it is referred as "Segmentation Fault".
- Examples of dangling pointers
    - After releasing dynamically allocated memory, pointer still keeping the old address.
    - Uninitialized (local) pointer
    - Pointer holding address of local variable returned from the function.
- It is advised to assign NULL to the pointer instead of keeping it dangling.

```
int main() {
    int *p = (int*) malloc(20);
    // …
    free(p); // now p become dangling
    // …
    return 0;
}
```

SUNBEAM

Thank you!

Devendra Dhande <devendra.dhande@sunbeaminfo.com>