CSA2090: Systems Programming

Introduction to C

Lecture 6: Memory Allocation and Management

Dr. Christopher Staff

Department of Computer Science & AI

University of Malta

cstaff@cs.um.edu.mt

Aims and Objectives

- Dynamic Memory
- malloc()
- free()
- realloc()

Why allocate memory?

- Whenever C encounters a variable definition (e.g., int x), then C automatically sets aside space
- But this means that you need to know exactly how much space is needed while you are designing the program!
- Give examples where it is difficult/impossible to tell in advance

Example

- Let's say we want to reverse a string, keeping a copy of the original string
- We first need to ask C for space…
- … then copy the original string into it…
- … and finally we will reverse the original

Requesting space

- `void *malloc(size_t size);`
- `void *calloc(size_t nobj, size_t size);`

malloc()

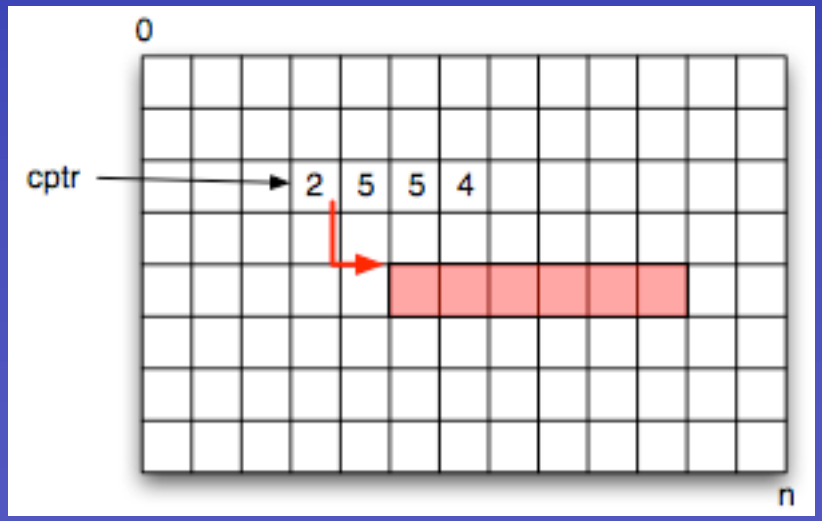
- First, how much space do we need?
- Assume `char str[] = "Hello"`
- `int len = strlen(str)+1`
- Why + 1?

malloc()

```
char *cptr = NULL;
cptr = (char *)malloc(len);
```

type conversion

- This is now a permanent area of memory



Reversing the string

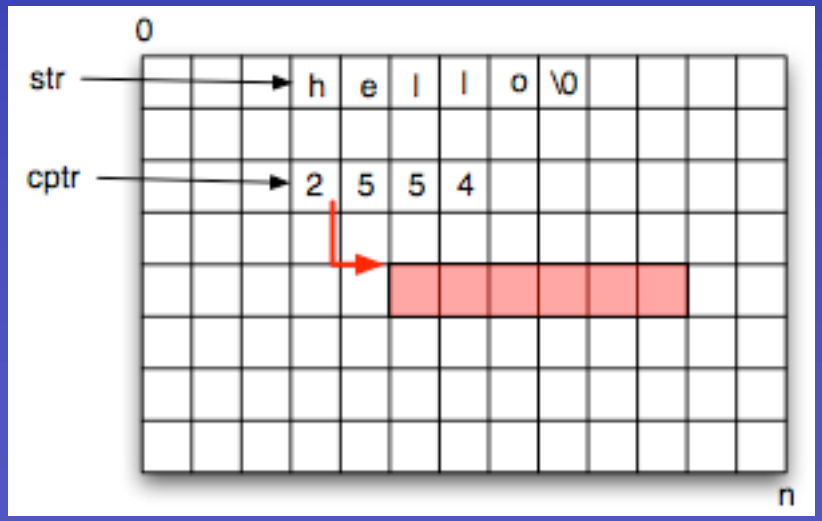
- What's the last character in `str`?

```
char str[] = "Hello"
```


Reversing the string

- What's the last character in `str`?

  `char str[] = "Hello"`



Reversing the string

```
cptr = cptr + len;
*cptr = '\0';
while (*str) {
   cptr--;
   *cptr = *str;
   str++;
} // see reverse.c
```


free()

- The memory area pointed at by cptr will persist until
  - The *program*, rather than the function, terminates
  - It is explicitly freed

```
free (c_ptr);
```

- If you "lose" the pointer, you cannot free memory

realloc()

- You can use `malloc()` or `calloc()` to allocate space for an array
- `malloc(7)` OR `calloc(7, 1)` will allocate the same amount of space
- Dynamic arrays can grow in memory, but you need to reallocate them to ensure that it's done safely

realloc()

```
void *realloc(void *p, size_t
size)

new_ptr = (int *)realloc(old_ptr,
newsize);
```


Full-up! Imma kif tghidilhom?

- You should *always* check that the `malloc(), calloc(), realloc()` operation has been successful!
- They will return NULL if they fail
  - And of course, realloc will lose the pointer if `ptr = (int *)realloc(ptr, newsize)` returns NULL!

Linked Lists, etc

- Dynamic memory is used to maintain linked lists, trees, and other data structures that cannot be efficiently pre-allocated
- See Love for examples (Appendix A and B)

Next week

- Input and Output
- Source File organisation