Lecture 14: ArrayList

CSE 29: Systems Programming and Software Tools

Olivia Weng

How could we implement an ArrayList in C?

How could we implement an ArrayList class in C?

```
struct list {
    uint32_t size;
    uint32_t capacity;
    String *contents; // the actual list of strings
};
```

Demo

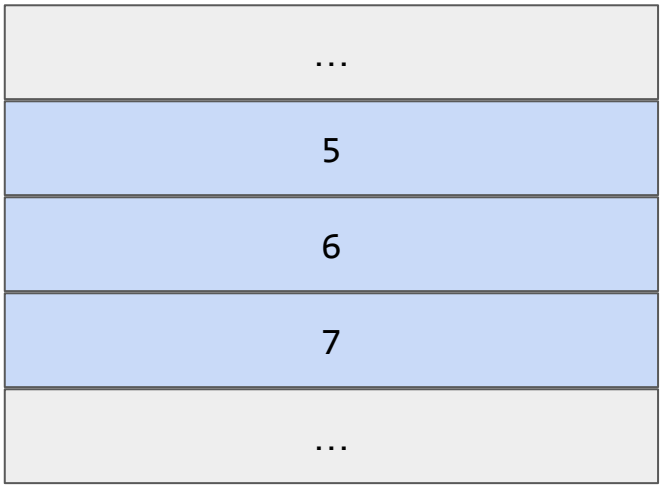
- `new_List()`
- `append1()`
- `expandCapacity()`
- `destroy_List()`

Pointer arithmetic

- We can do addition on pointers to get a new address!

```
int a[3] = {5, 6, 7};

int *pa = a;
```

```
pa = 0x100
pa + 1 = 0x104
pa + 2 = 0x108
0x10C
```



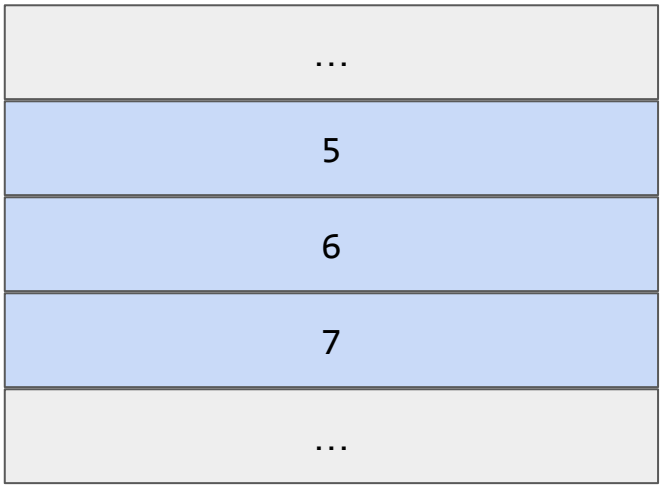
Rule: pa + n =

Pointer arithmetic

- We can do addition on pointers to get a new address!

```
int a[3] = {5, 6, 7};

int *pa = a;

int b = *(pa + 1);

printf("%d\n", b);
```

```
pa = 0x100
pa + 1 = 0x104
pa + 2 = 0x108
0x10C
```



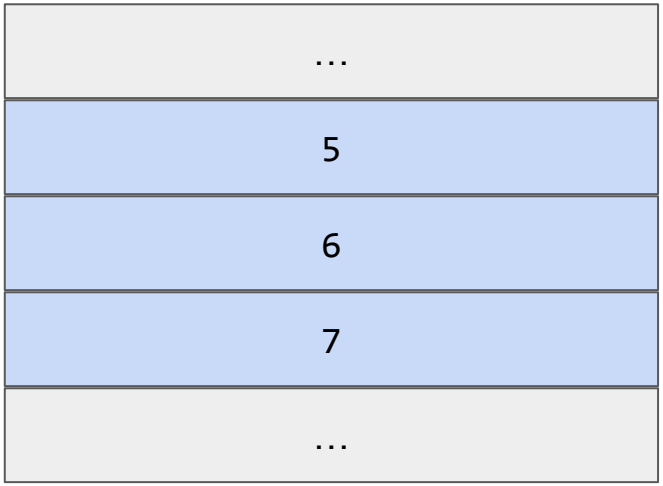

```
Rule: pa + n = pa + n * (sizeof(int))
```

Pointer arithmetic

- We can do addition on pointers to get a new address!

```
int a[3] = {5, 6, 7};

int *pa = a;

int b = *(pa + 1);

printf("%d\n", b);

printf("%d\n", *(pa + 1) + *(pa + 2));

printf("%d\n", pa[1] + pa[2]);
```

```
pa = 0x100
pa + 1 = 0x104
pa + 2 = 0x108
0x10C
```



Pointer arithmetic

- General rule: `ptr + n = ptr + n * sizeof(type)`

Pointer arithmetic

- General rule: `ptr + n = ptr + n * sizeof(type)`

```
char str[] = "Hi CSE29!";


printf("%c\n", *(str + 1));

printf("%c\n", str[1]));

printf("%s\n", str + 3);
```

Pointer arithmetic

- General rule: `ptr + n = ptr + n * sizeof(type)`

```
char str[] = "Hi CSE29!";

char str1[] = "30";

strncpy(str + 6, str1, strlen(str1));


printf("%s\n", str);
```