







Programming with C I

Fangtian Zhong
CSCI 112

Gianforte School of Computing
Norm Asbjornson College of Engineering
E-mail: fangtian.zhong@montana.edu

Previous uses of pointers...

-  **Reference to data**
-  **Output parameters**
-  **Arrays and strings**
-  **File pointers**

What happens when we run our executable file?

```
func1(int x) {  
    x += 1;  
    return(x);  
}
```

```
int main(void) {  
    int n = 10;  
    n = func1(n);  
    return(0);  
}
```



Stack memory



What happens when we run our executable file?



```
int main(void) {  
    int* nump;  
    nump = malloc(sizeof(int));  
    *nump =  
    free(nump);  
}
```

Stack memory



Heap memory

What happens when we run our executable file?

```
int main(void) {  
    int* nump;  
    nump = malloc(sizeof(int));  
    *nump =  
    free(nump);  
}
```



Stack memory



Heap memory

What happens when we run our executable file?

```
int main(void) {  
    int* nump;  
    nump = malloc(sizeof(int));  
    *nump =  
    free(nump);  
    *nump++;  
}
```

undefined behavior!



Stack memory



Heap memory

Dynamic Memory Allocation

➤ heap

- region of memory in which function **malloc** dynamically allocates blocks of storage

➤ stack

- region of memory in which function data areas are allocated and reclaimed

Important functions

- **malloc(<amnt of memory to reserve>)**
- **calloc(<num>, <amnt of memory to reserve>)**
- **free(pointer)**

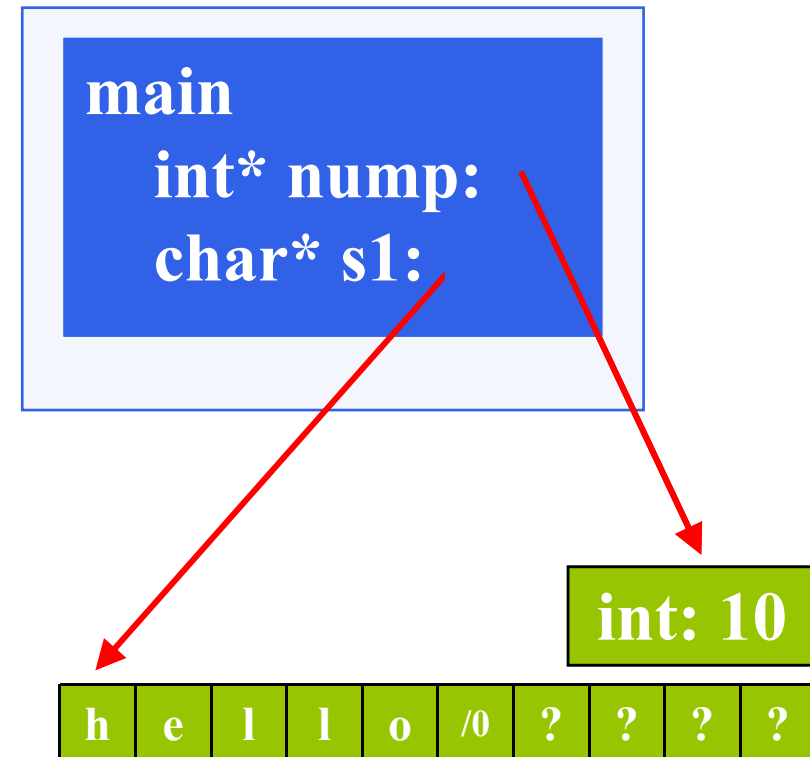
These are all from stdlib.h.

What happens when we run our executable file?

```
int main(void) {  
    int*  
    nump = malloc(sizeof(int));  
    *nump = 10;  
    char* string1;  
    string1 = calloc(10, sizeof(char));  
    strcpy(string1, "hello");  
    free(nump);  
}
```



Stack memory



Heap memory

What happens when we run our executable file?

```
int main(void) {  
    int*  
    num = malloc(sizeof(int));  
    *num = 10;  
    char* string1;  
    string1 = calloc(10, sizeof(char));  
    strcpy(string1, "hello");  
    free(num);  
}
```



Stack memory



Heap memory

Figure Multiple Pointers to a Cell in the Heap

```
double *p, *xcopyp;
```

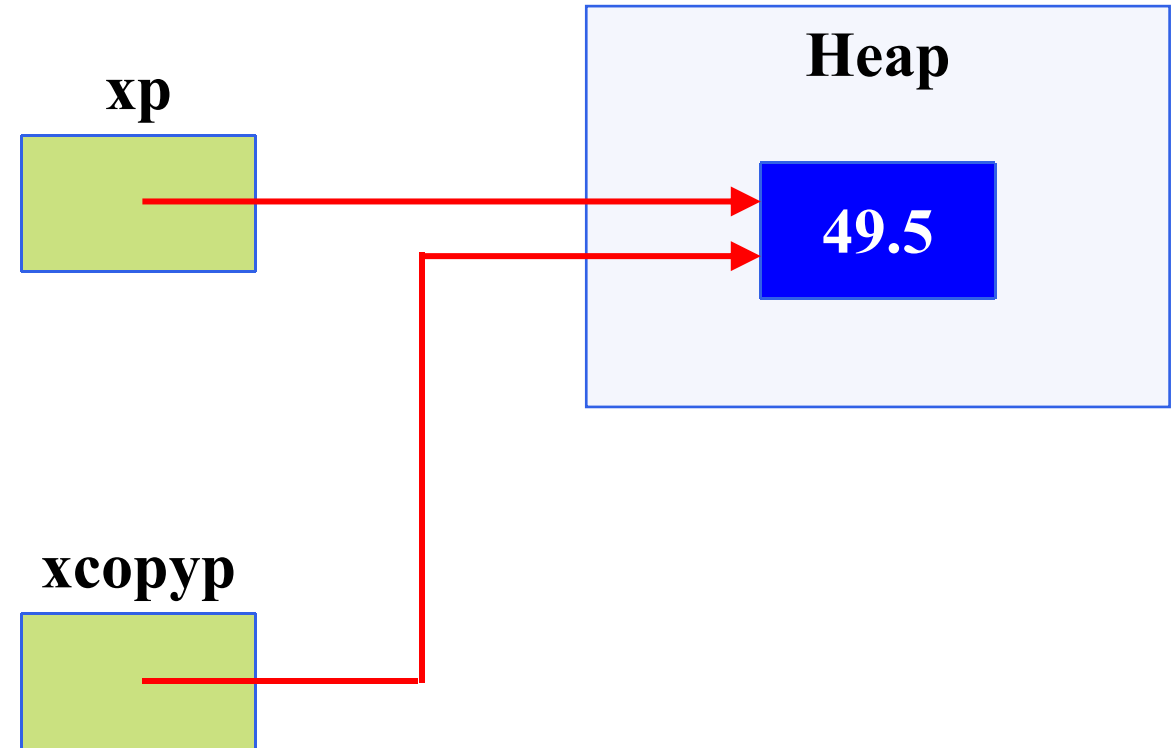
```
xp = (double *)malloc(sizeof(double));
```

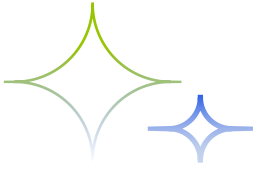
```
*xp = 49.5;
```

```
xcopyp = xp;
```

```
free(xp);
```

```
...
```





THE END

Fangtian Zhong
CSCI 112