

Programming with C I

Fangtian Zhong
CSCI 112

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

Segmentation fault

- Runtime error
- Means you tried to access memory that you weren't allowed to access
- Examples of causes:
 - trying to read from a file that wasn't open
 - following a dangling pointer
 - accessing data beyond array bounds

Let's introduce a segmentation fault in nofile.c

The NULL pointer

- Uninitialized pointers point somewhere
- NULL is a pointer that points nowhere
- 0 also works for the null pointer

```
int* ptr = NULL;  
  
if (ptr == NULL) {  
    ...  
    ...  
}
```

sizeof() function

➤ Gives the number of bytes that a variable or value takes up

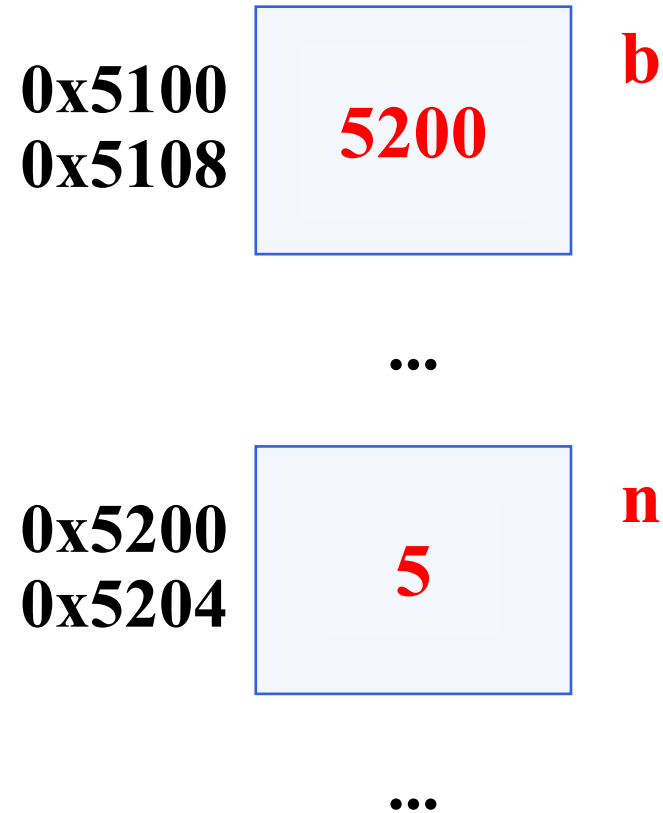
sizeof(m)

- On our server:
- Char: 1 bytes
 - Int: 4 bytes
 - Float: 4 bytes
 - Double: 8 bytes
 - Pointer: 8 bytes

Pointers

➤ Create an integer pointer variable and set it

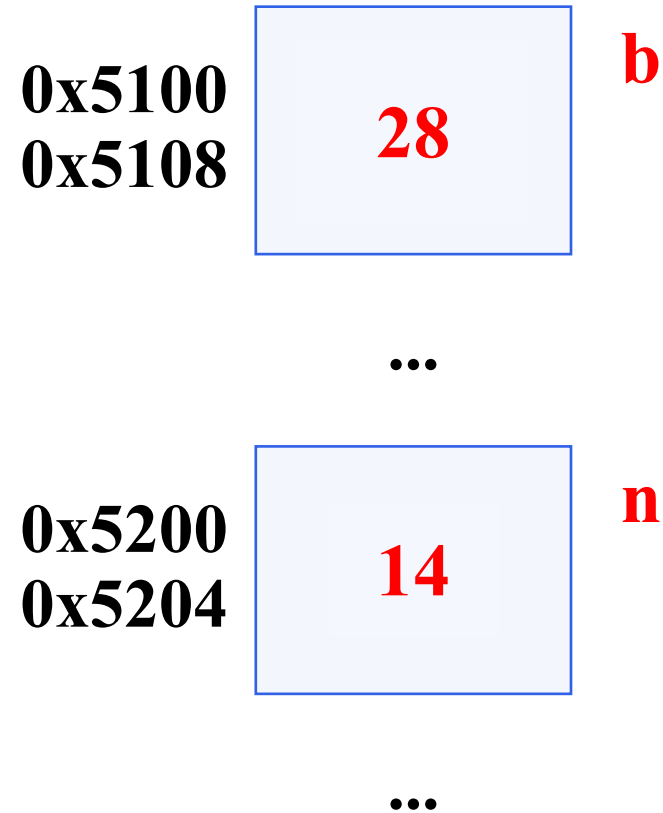
```
int main(void) {  
    int *b;  
    int n;  
    n = 5;  
    b = &n;  
}
```



Pointers

➤ Create an integer pointer variable and set it

```
int main(void) {  
    int *b;  
    int n;  
    n = 5;  
    b = &n;  
    n = 6;  
    *b += 1;  
    *b = 2 * (*b);  
    b = 2 * (*b);  
}
```



ptr0.c shows seg fault accessing *b

Functions with Output Parameters

- We've used the return statement to send back one result value from a function.
- We can also use output parameters to return multiple results from a function.

➤ *Figure Diagram of function separate with Multiple Results*

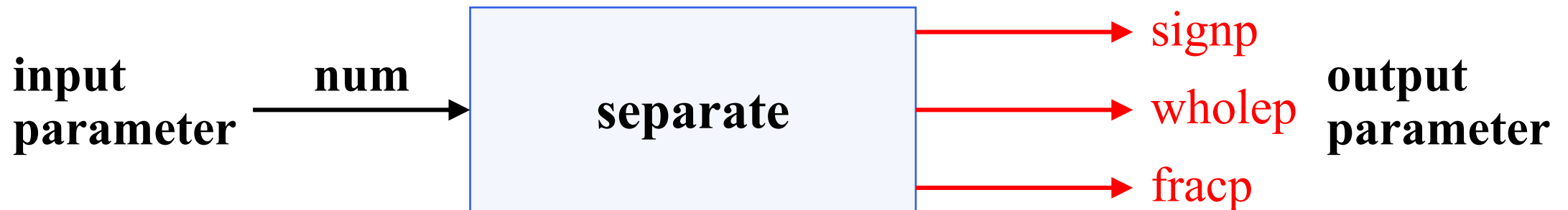


Figure Parameter Correspondence for `separate(value, &sn, &wh1, &fr);`

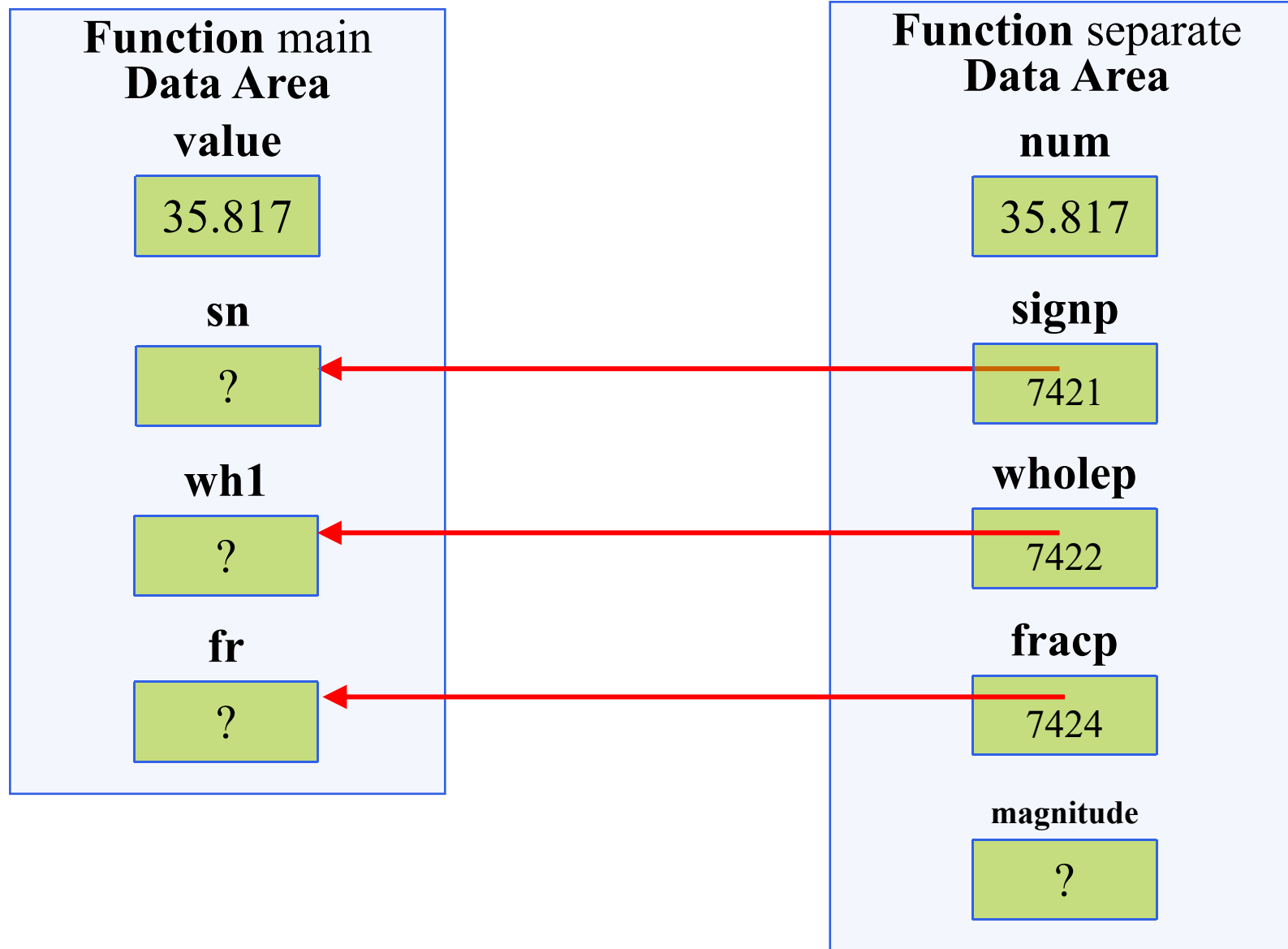



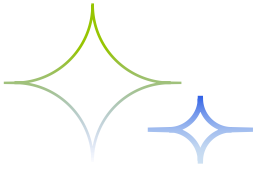


Table Effect of & Operator on the Data Type of a Reference

| Declaration | Data Type of x | Data Type of &x |
|-------------|----------------|------------------------------|
| char x | char | char * (pointer to char) |
| int x | int | int * (pointer to int) |
| double x | double | double * (pointer to double) |

Meaning of Symbol *

-  **binary operator for multiplication**
-  **“pointer to” when used when declaring a variable or a function parameters**
-  **unary indirection operator in a function body**



THE END

Fangtian Zhong
CSCI 112

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