08. Assignment #4 (System Programming)

Yuseok Jeon

ULSAN NATIONAL INSTITUTE OF SCIENCE AND TECHNOLOGY
2007

Assignment #4 (100 points)

- Assignment #4 will be released (May 10)
- Due May 23, 2023: 11:59pm

- Platform
  - We will work on Ubuntu 22.10 (latest version)
  - https://releases.ubuntu.com/kinetic/ (Desktop image)

- If you use MAC, please use Docker Desktop on Mac
  - https://docs.docker.com/desktop/install/mac-install/

- If you cannot make this environment, please contact our TA
  - 양성현 (SUNGHYUN@UNIST.AC.KR)
  - We will provide a server with account for your assignment

Goal

- Write a custom allocator that manages a heap area minimally
- Your allocator manages 64 bytes of a heap area

What you need to implement

A `program` that

- allocates 64 bytes of a heap area and manages it (suppose that there is not enough memory you can use)
- runs in *an infinite while loop* where it gets input from users through ***stdin***

Users can request a type of data (which will be allocated in the heap area) and its value to the program, then the program *allocates a room* in the heap area and assigns the value to the room

- e.g., a user can request a struct which has 2 int types and specify values for each int type in the struct

What you need to implement

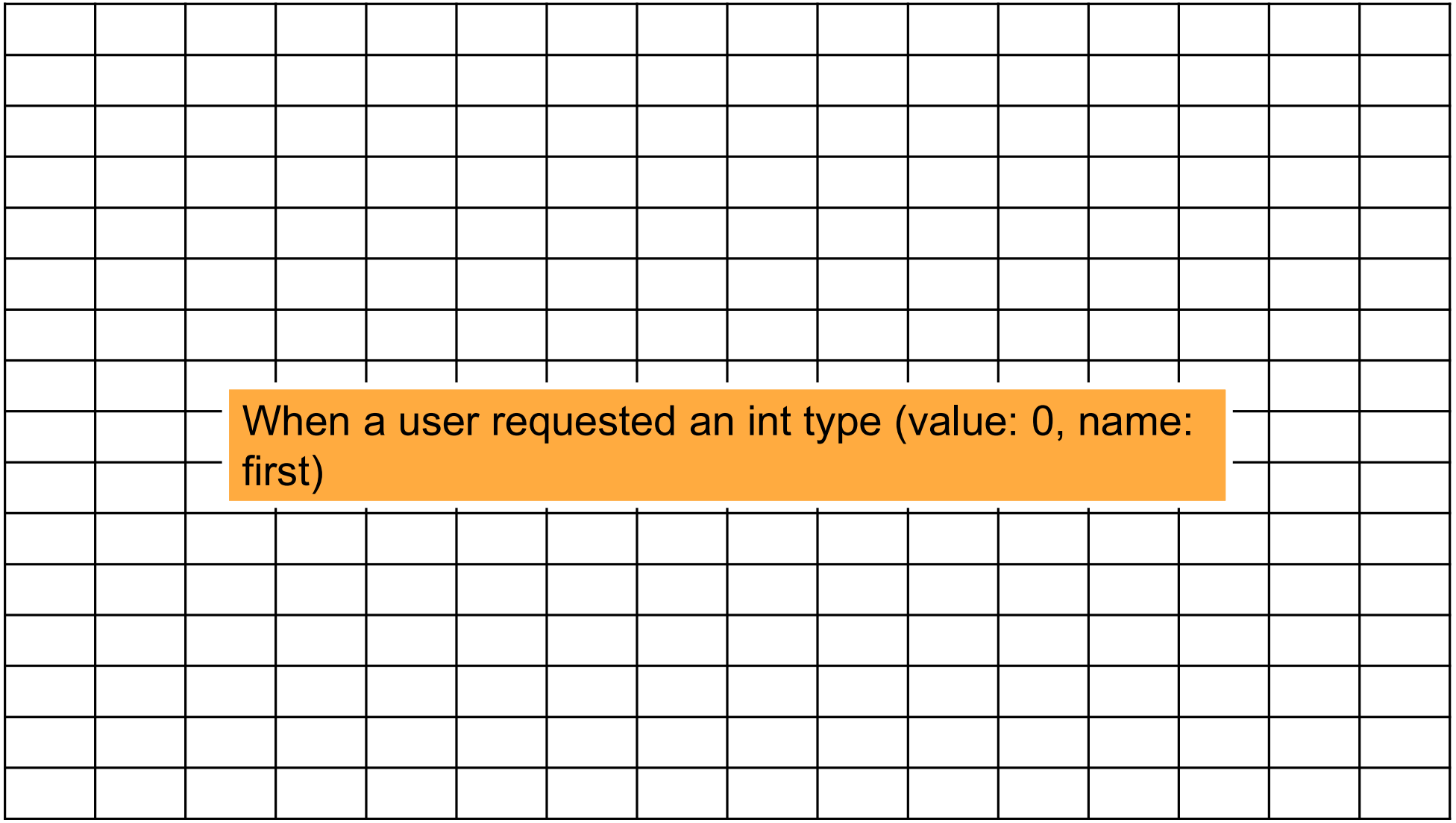
There must be no padding bytes around any data in the heap area
Also, there must be no padding bytes in any strcut type data

Also, users can request *deallocation* of any data

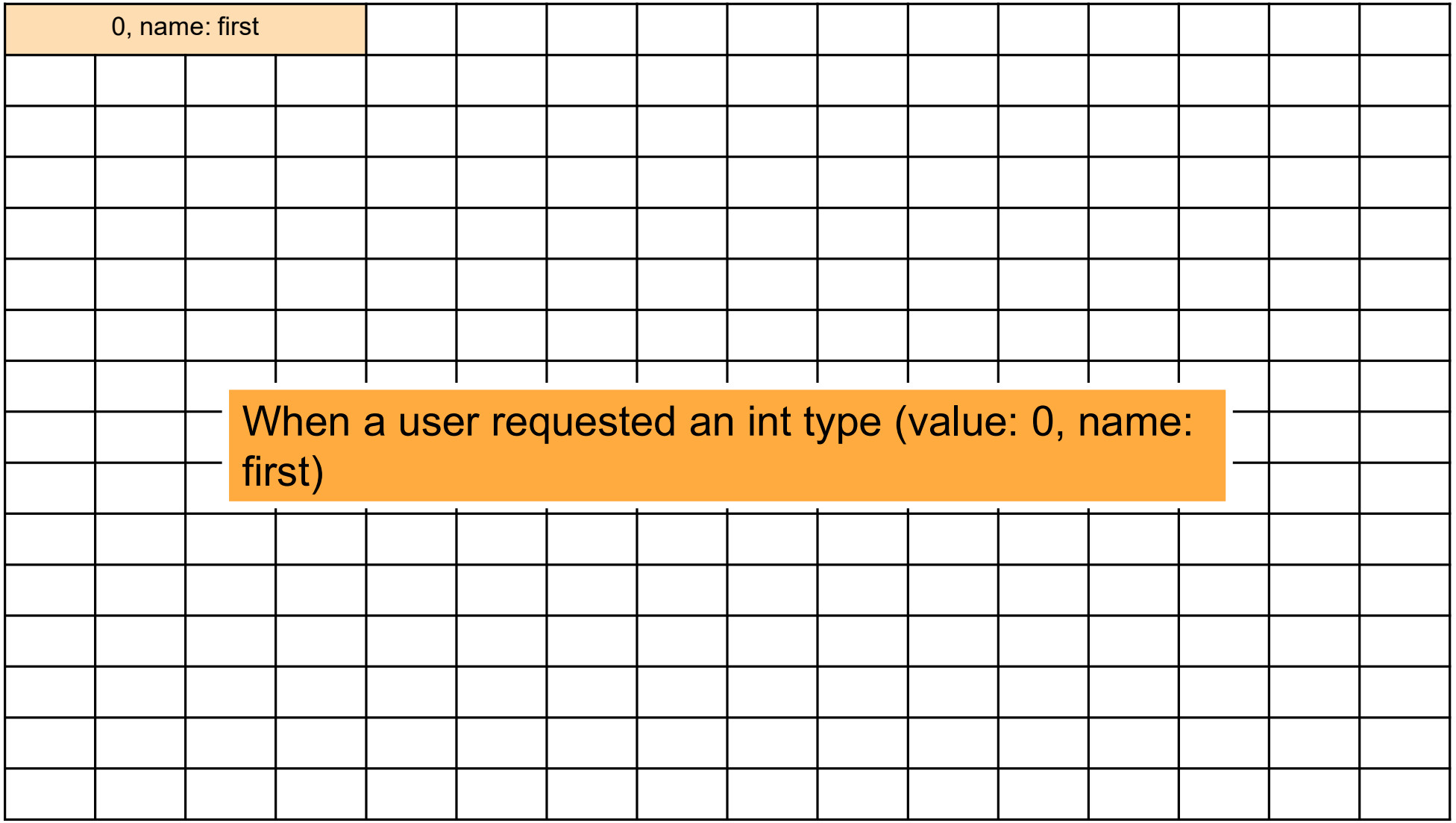
- When a deallacation of data requested, your program must move the other data to fill the hole (unless the data to be deallocated is stored at the last part of the heap area)

Your program must print out memory dump of the heap area and data list(see printing format section)

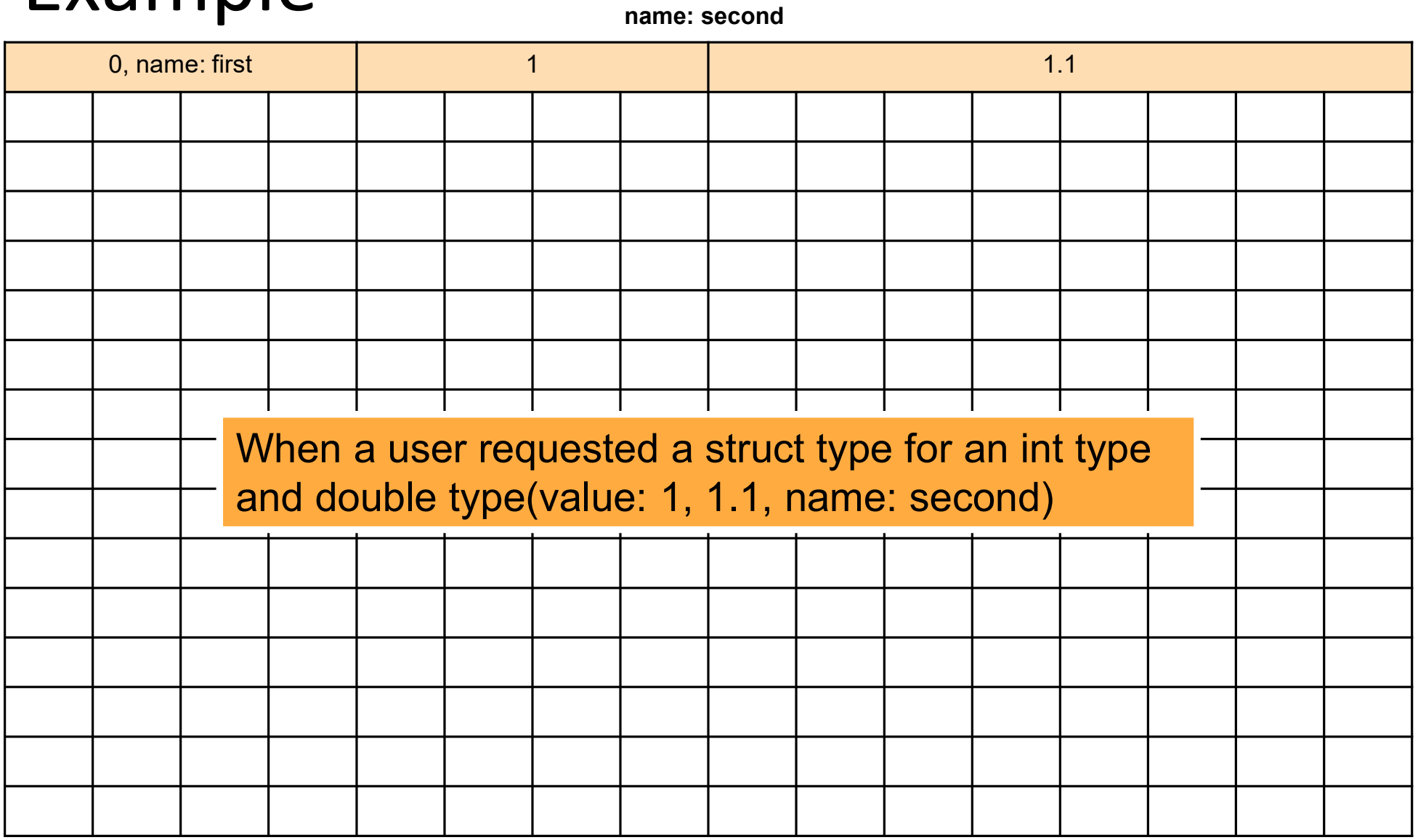
Example



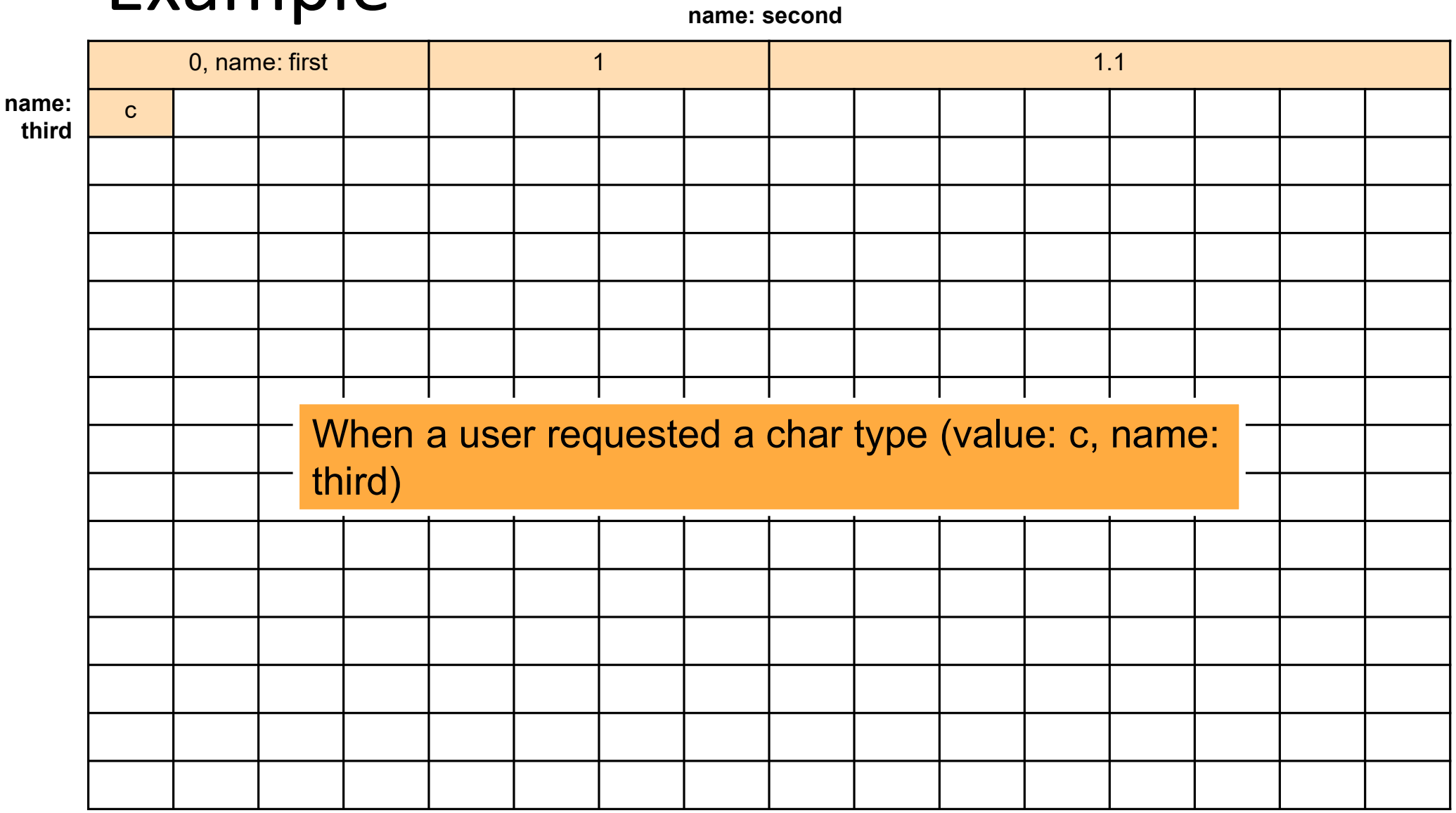
Example


0, name: first
When a user requested an int type (value: 0, name: first)

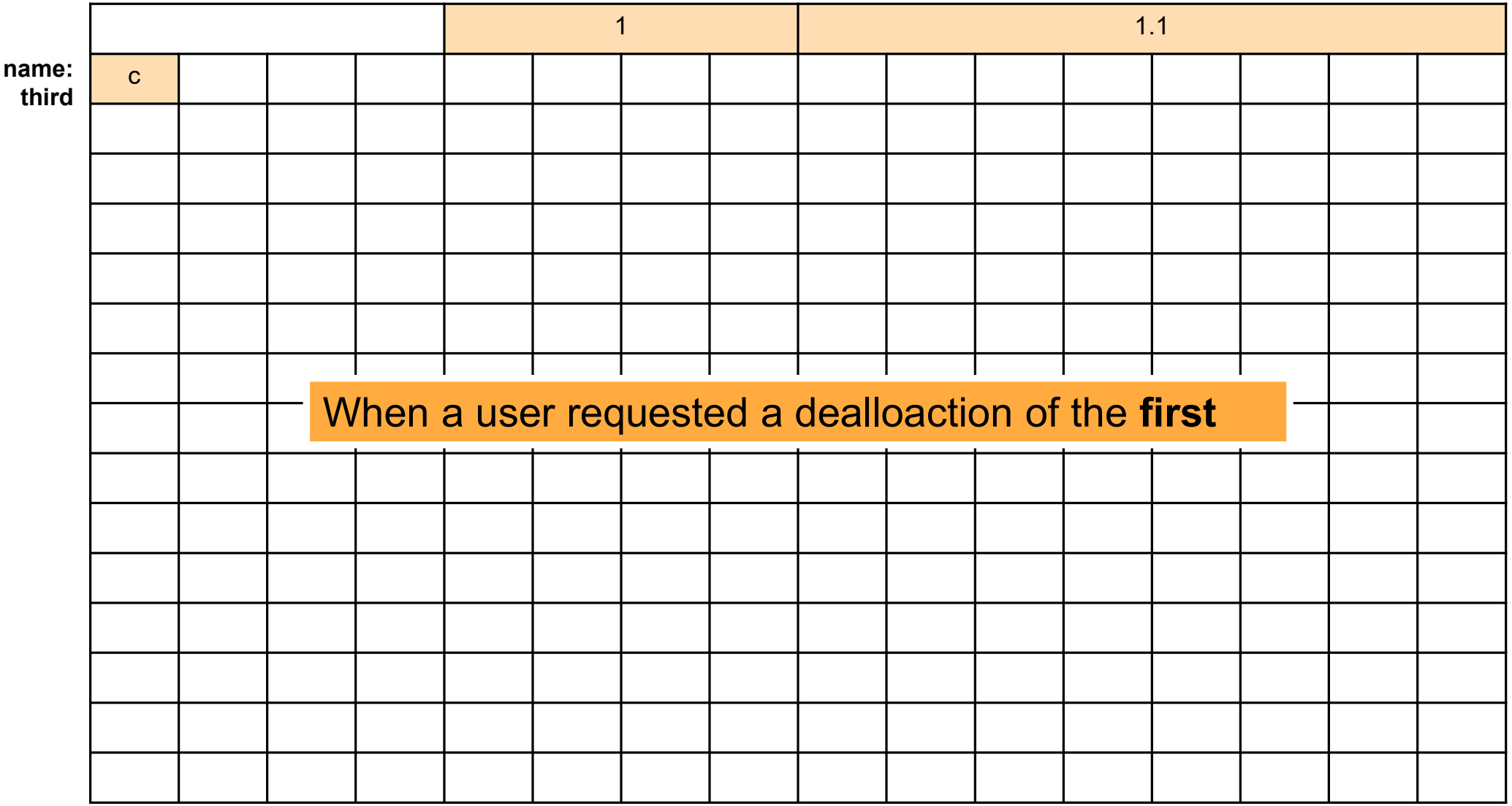
Example


name: second
0, name: first
1
1.1
When a user requested a struct type for an int type and double type(value: 1, 1.1, name: second)

Example


name: second
0, name: first
1
1.1
name: third
c
When a user requested a char type (value: c, name: third)

Example


name: second
1
1.1
name: third
c
When a user requested a dealloaction of the **first**

Example

name: second

name: third

1				1.1								c			

When a user requested a dealloaction of the **first**

If there is not enough memory for the requested data

It should not be allowed!

Your program just prints out a string
"**There is not enough memory for the data, you can only use X byte(s)**"

e.g., **There is not enough memory for the data, you can only use 1 byte(s)**

Printing format and explain again

When user requested a short type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
short short_a
Please input a value for the data type
7
There is memory dump!
07 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
```

Printing format and explain again

When user requested a float type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
float float_a
Please input a value for the data type
7.7
There is memory dump!
07 00 66 66 f6 40 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
float_a
```

Printing format and explain again

When user requested a char type

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
char char_a
Please input a value for the data type
k
There is memory dump!
07 00 66 66 f6 40 6b 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
float_a
char_a
```

Printing format and explain again

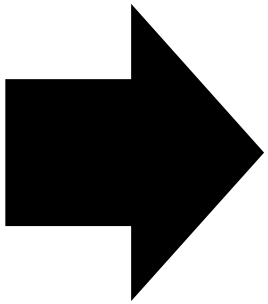
When user requested a struct type which has short and float data types

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
struct struct_a
How many data should be in the struct
2
Please input each type and its value
short 9
float 7.7
There is memory dump!
07 00 66 66 f6 40 6b 09 00 66 66 f6 40 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
float_a
char_a
struct_a
```

Printing format and explain again

When user requested a deallocation of float_a

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
struct struct_a
How many data should be in the struct
2
Please input each type and its value
short 9
float 7.7
There is memory dump!
07 00 66 66 f6 40 6b 09 00 66 66 f6 40 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
float_a
char_a
struct_a
```



```
Do you want to allocate data (1) or deallocate data (2) ?
2
Input the name of data you want to deallocate
float_a
float_a has been deallocated
There is memory dump!
07 00 6b 09 00 66 66 f6 40 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

------Data you have now------
short_a
char_a
struct_a
```

Printing format and explain again

When there is not enough memory for the data

```
Do you want to allocate data (1) or deallocate data (2) ?
1
Input the type of data you want to allocate and the name of the data
short short_b
There is not enough memory for the data which you require, you can only use 0 byte(s)
```

Types of data that your program supports

- `short`
- `char`
- `float`
- `long`
- `Struct(which has above data types and specific values)`

Use this dump_mem function!

```
void dump_mem(const void *mem, size_t len) {
    const char *buffer = mem;
    size_t i;
    for (i=0; i<len; i++){
        if (i>0 && i%16 == 0) {
            printf("\n");
        }
        printf("%02x ", buffer[i] & 0xff);
    }
    puts("");
}
```

Summary

You need to implement a program which has allocation and deallocation.

Program allocates 64 bytes of a heap area and manages it.

There must be no padding bytes around any data in the heap area and any struct type data.

Summary

- Allocation
  - User requests a type of data(short, char, float, long and struct) and its value to the program
  - Program allocates a room in the heap area and assigns the value to the room
  - If there is not enough memory, it should not be allowed
  - After allocating, program prints out memory dump of the heap area and allocated data list

Summary

- Deallocation
  - User requests deallocation of data
  - When deallocated, program must move the other data to fill the hole
  - After deallocating, program prints out memory dump of the heap area and allocated data list
- [Slide] 03.Memory representation may be helpful for you to understand the assignment

Submission

- You should submit your code with a code description that explains your code (i.e., comments in the file). In the description, your code must be well commented to explain your algorithm. Make your code .zip file with ”StudentID_YourName.zip" and submit your .zip file on blackboard.

ex)

20195147_HongjunYang.zip

- assignment4.c // assignment4 code
- assignment4.h // assignment4 header file
- Makefile // Makefile
- report.pdf // assignment4 code description

For the report, please write it in detail.