Week 6

- Arrays
- Strings

Where do we have problems?

- Lists
- Multiple of something
- Text

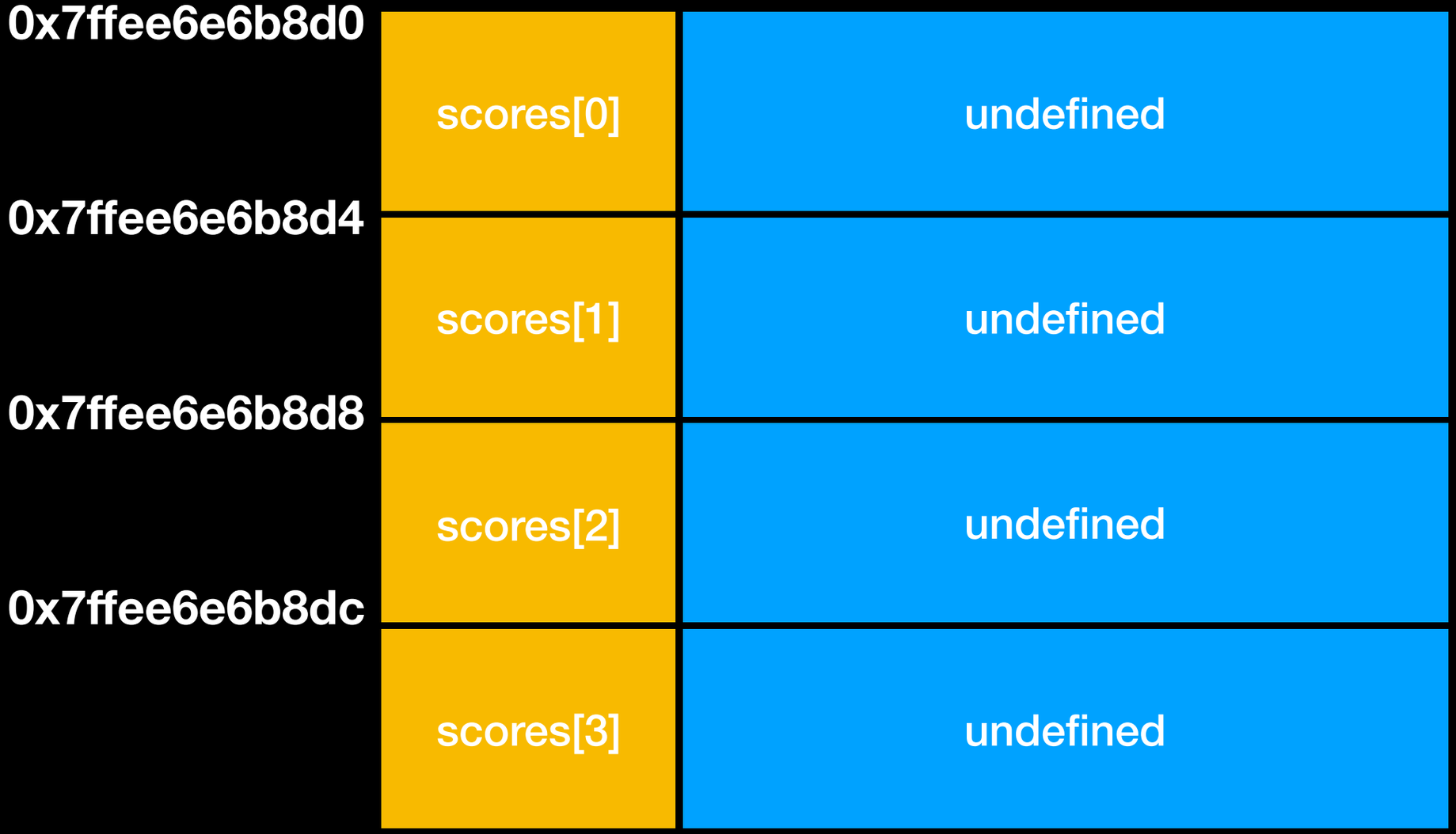
Array

- int scores[10];

- An array is a fixed size sequential collection of elements of the same type

- Addressable through e.g. scores[5] //which element?

int scores[4];


0x7ffee6e6b8d0
scores[0]
undefined
0x7ffee6e6b8d4
scores[1]
undefined
0x7ffee6e6b8d8
scores[2]
undefined
0x7ffee6e6b8dc
scores[3]
undefined

```c
int main(int argc, char **argv){
    int numbers[4];
    numbers[0] = 5;
    numbers[1] = 2;
    numbers[2] = 12;
    numbers[3] = 2;
    int sum = 0;
    for(int i = 0; i < sizeof(numbers)/sizeof(int); i++){
        sum += numbers[i];
    }
    printf("%i\n", sum);
    return 0;
}
```

How many elements?

21

Let’s extract it to a function!

```c
int sum(int numbers[]){
    int sum = 0;
    for(int i = 0; i < sizeof(numbers)/sizeof(int); i++){
        sum += numbers[i];
    }
    return sum;
}
```

Yields 7.. Why?

```
test.c:6:30: warning: sizeof on array function parameter will return size of 'int *' instead
of 'int [4]'
      [-Wsizeof-array-argument]
    for(int i = 0; i < sizeof(numbers)/sizeof(int); i++){
                             ^
test.c:4:13: note: declared here
int sum(int numbers[4]){
            ^
1 warning generated.
```

Array is converted to pointer?

Hear some words from Linus Torvalds:

```
Christ, people. Learn C, instead of just stringing random characters
together until it compiles (with warnings).

This:

  static bool rate_control_cap_mask(struct ieee80211_sub_if_data *sdata,
                                    struct ieee80211_supported_band *sband,
                                    struct ieee80211_sta *sta, u32 *mask,
                                    u8 mcs_mask[IEEE80211_HT_MCS_MASK_LEN])

is horribly broken to begin with, because array arguments in C don't
actually exist. Sadly, compilers accept it for various bad historical
reasons, and silently turn it into just a pointer argument. There are
arguments for them, but they are from weak minds.
```

When an array is passed as a parameter, it is downcasted to a pointer

```
int sum(int numbers[])
```

Becomes

```
int sum(int *numbers)
```

Pointer Arithmetic

- Addition and Subtraction on a pointer
- Ptr + 1 doesn’t increase the address by 1
- Ptr + 1 increases by the size of the value pointed to
  In a nutshell: go to next value

Pointer Arithmetic

```c
int numbers[4];
numbers[0] = 5,  numbers[1] = 2;
numbers[2] = 12, numbers[3] = 32;
int* ptr = numbers;

printf("%i\n", *ptr); //5
ptr++;
printf("%i\n", *ptr); //2
ptr+=2;
printf("%i\n", *ptr); //32
```

```c
int numbers[4];
numbers[0] = 5,  numbers[1] = 2;
numbers[2] = 12, numbers[3] = 32;
int* ptr = numbers;

printf("%i\n", ptr[0]); //5
printf("%i\n", *(ptr+1)); //2
printf("%i\n", ptr[1]); //2
printf("%i\n", *(ptr+3)); //32
printf("%i\n", ptr[3]); //32
```

```c
int sum(int *numbers){
    int sum = 0;
    for(int i = 0; i < sizeof(numbers)/sizeof(int); i++){
        sum += numbers[i];
    }
    return sum;
}
```

Still only 7

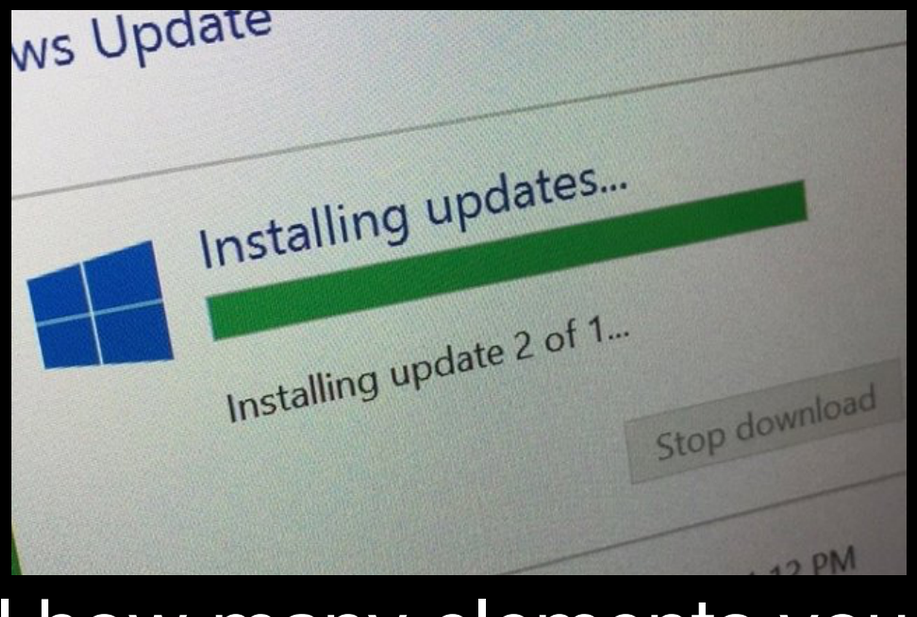
What is going on here?

- A pointer points to some address in memory
- A pointer has no size or number of elements associated
- C doesn’t save the size of some array in memory, why should it?
- Assumption: If you need the size of an array, store it YOURSELF

Pass the size of the array as parameter

```
int sum(int *numbers, int size){
    int sum = 0;
    for(int i = 0; i < size; i++){
        sum += numbers[i];
    }
    return sum;
}
```

Off by One Errors



- You need to tell how many elements you pointer may access
- One short -> one element ignored
- One too much -> Try it out!

How far can we go?

- Try accessing array elements out of bounds

- Negative numbers?
  Positive numbers?

- Will your program crash at some point?

```c
printf("%i\n", ptr[10]);
printf("%i\n", ptr[100]);
printf("%i\n", ptr[1000]);
```

1599287429

778121006

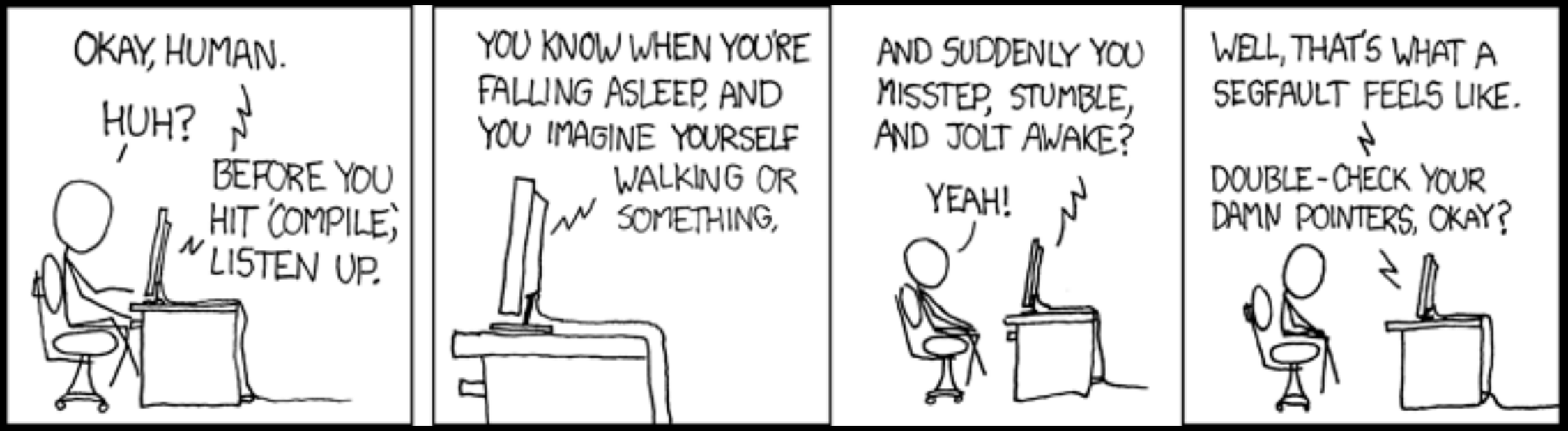
[1] 8970 segmentation fault ./a.out

What is a segmentation fault?

What is a Segmentation Fault?

- Your program gets access to specific parts of memory on your computer.
- When your program attempts to do something with memory it doesn’t have access to, it’s an unrecoverable bug and the program gets killed

What is a Segmentation Fault?



„Ich glaube Pointer und Strings sollte gesplittet werden.“

–Ich, als ich die Folien geschrieben habe

Exercises

- Double all numbers in an array and print it
- Sort 20 numbers in an array
- Write some code to sort any array, benchmark it with 1.000.000 numbers. Use random numbers for initialization
- Create a „screen“ by having a buffer where each character is a pixel.
  Write a function to draw a square by position and dimensions
- Write a function that takes an array and adds n elements together