

The image features the GlobalLogic logo in white text on a dark blue background. The background is a blurred office scene with people sitting at desks. The logo consists of the word "GlobalLogic" in a sans-serif font, with a registered trademark symbol (®) to the upper right of the "c".

GlobalLogic<sup>®</sup>

C/C++ memory organization

# Agenda

1. Using memory
2. Memory organization
3. Stack
4. Heap
5. Catching memory bugs

# Using memory



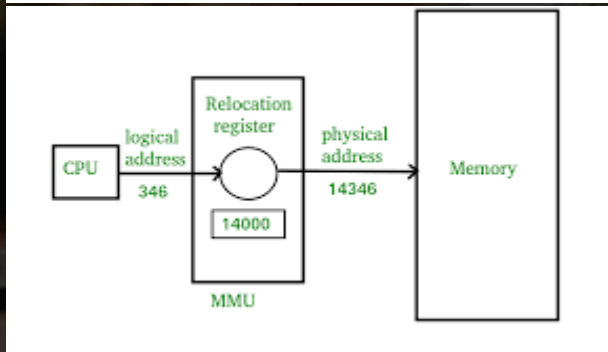
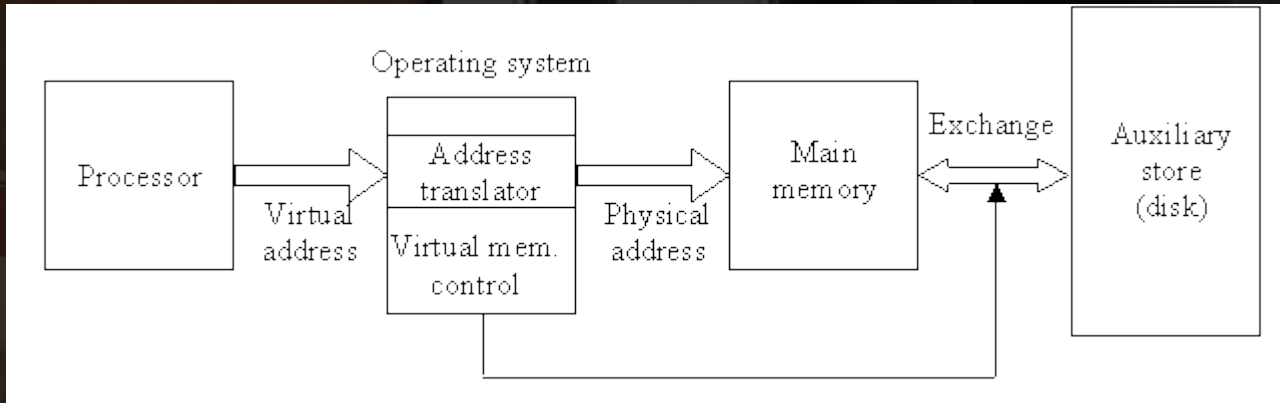
## Using memory

1. Every line of code uses memory (rather ROM or RAM) unless it is optimized by compiler
2. Every variable uses memory
3. Some code uses dynamic memory allocation

## Addressing memory

- Direct accessible memory – can be addressed by pointer, contain variables (Internal Ram, Internal Flash, mapped memory)
- Memory that cannot be directly accessed – External Ram or Rom.
- Virtual memory – can be addressed only in specific process or thread inside specific operating system

# Accessing memory

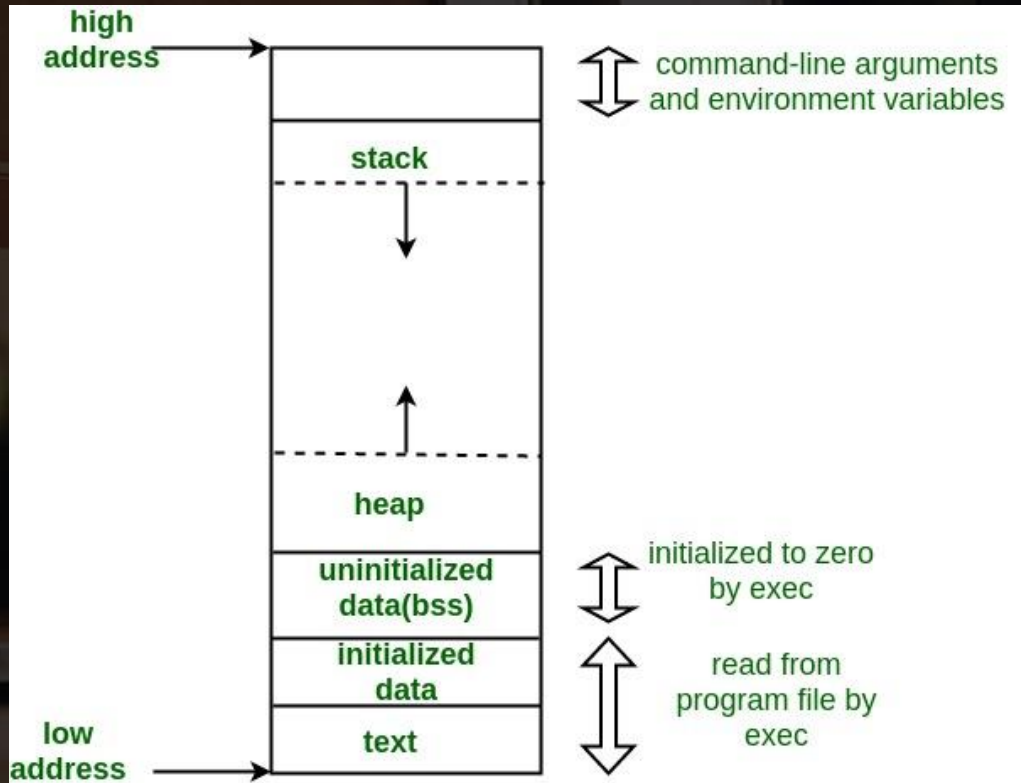


# Memory organization



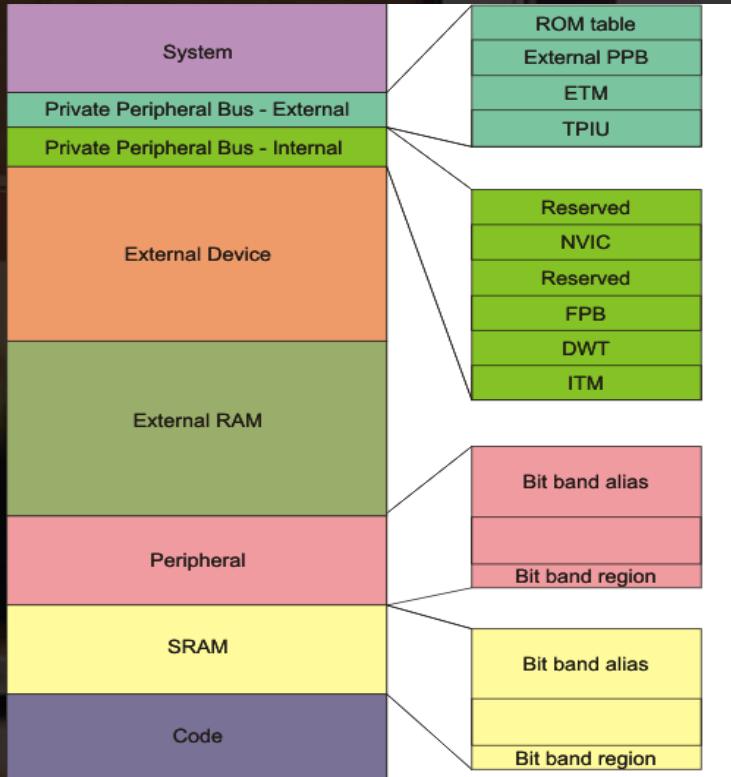


# C program memory layout





# Arm cortex memory organization



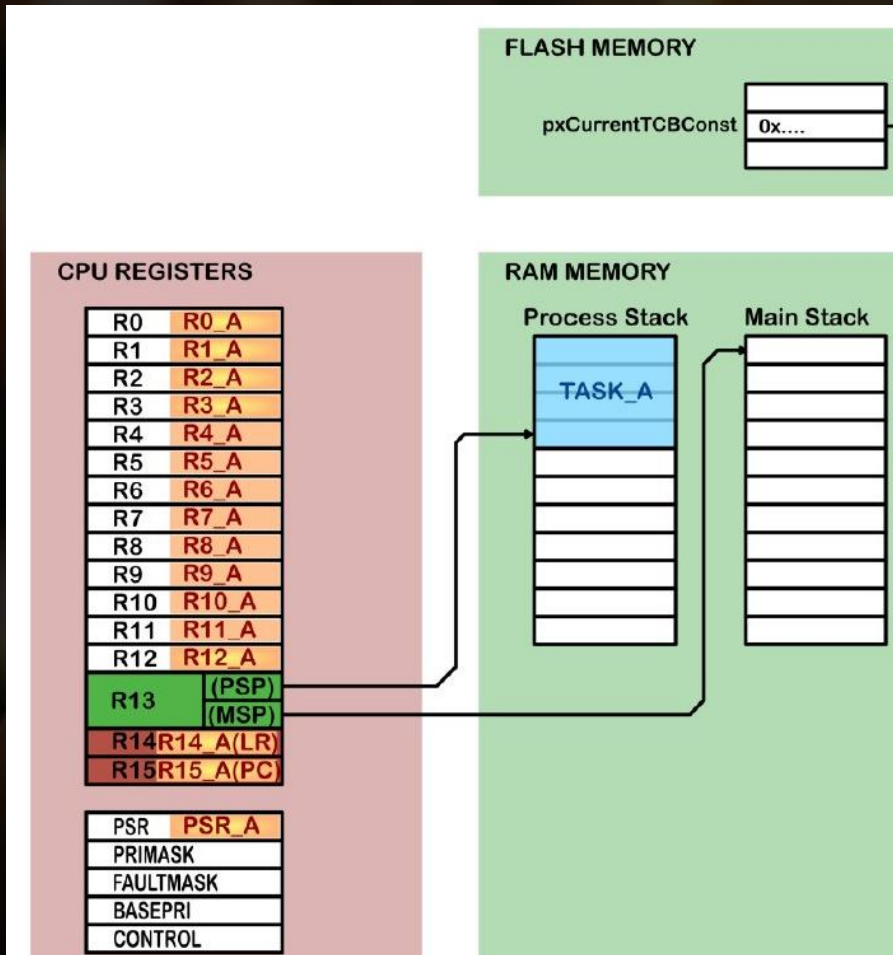
## Why data is placed to one of this sections

- .text – executable code. Should be read only. From this section code runs. May be located in RAM or ROM.
- Data may be located in Ram and Rom. Constant data belongs to ROM, dynamic belongs to RAM
- RAM contains stack and heap.

# Stack

The background image shows a person's hands interacting with a tablet computer. The tablet screen displays a data dashboard with various charts and graphs. A smartphone is placed on the desk next to the tablet. The entire scene is overlaid with a semi-transparent teal filter.





## Stack usage

- Are all variables stored on stack?
- Are function argument are always transferred via stack?
- Are function returns are always transferred via stack?

## Stack usage

- Are all variables stored on stack - no
- Are function argument are always transferred via stack - no
- Are function returns are always transferred via stack - no



# Heap

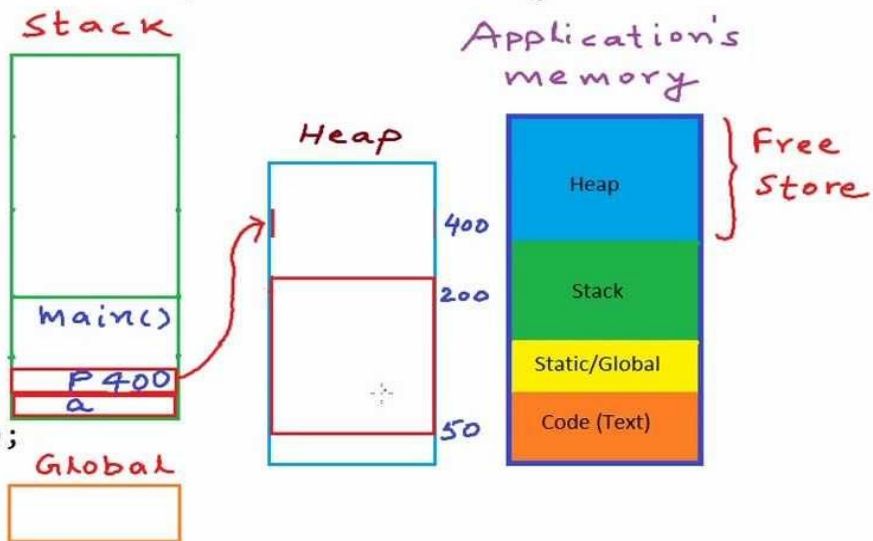


# Heap

- Only used dynamically
- Why to use heap -> to store temporary objects or when we short on memory

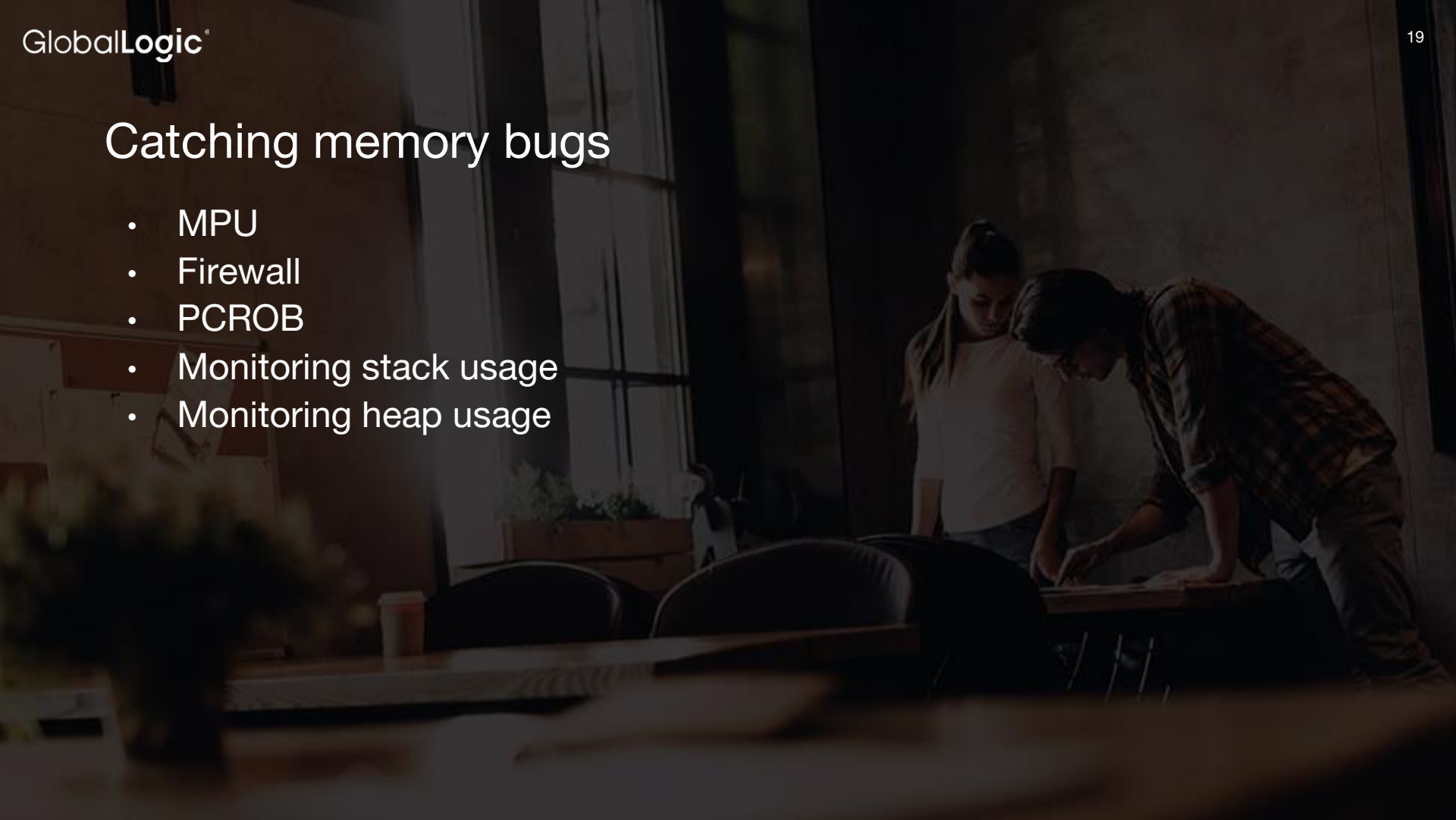
# Heap usage

```
#include<stdio.h>
#include<stdlib.h>
int main()
{
    int a; // goes on stack
    int *p;
    p = (int*)malloc(sizeof(int));
    *p = 10;
    free(p);
    p = (int*)malloc(20*sizeof(int));
}
```



## Catching memory bugs

- MPU
- Firewall
- PCROB
- Monitoring stack usage
- Monitoring heap usage



Thank you

The image features the GlobalLogic logo in white, centered over a dark blue, blurred background of an office interior. The background shows people sitting at desks with computers. A grid of light blue circular icons, each containing a different symbol, is overlaid on the entire image. The logo text is 'GlobalLogic' with a registered trademark symbol (®) to the upper right of the 'c'.

GlobalLogic®