CSED211: Lab. 4

BombLab

조성준
allencho1222@postech.ac.kr
POSTECH

2023.10.02

What We Have Learned

- GDB

- Assembly Language
  - Opcode
  - Operand
  - Registers
  - Instructions

Table of Contents

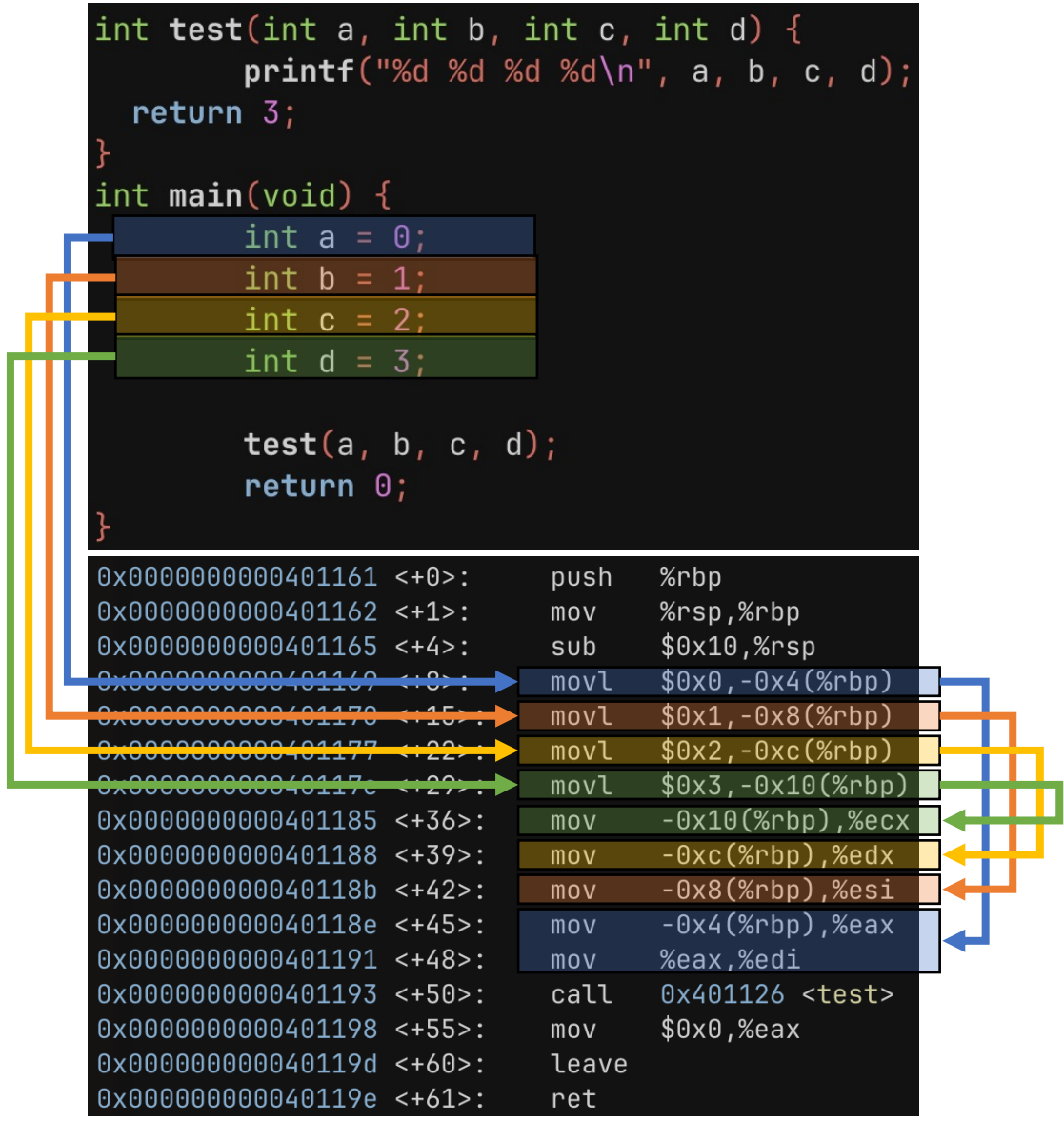
Assembly Language (Calling Convention)

- Calling convention answers
  - How does function receive parameters from the caller?
  - How does function return a value?

Assembly Language (Calling Convention)

- How does function receive parameters?
  - Some registers are used to pass parameters

Register	Purpose	Saved across calls
%rax	temp register; return value	No
%rbx	callee-saved	Yes
%rcx	used to pass 4th argument	No
%rdx	used to pass 3rd argument	No
%rsp	stack pointer	Yes
%rbp	callee-saved; base pointer	Yes
%rsi	used to pass 2nd argument	No
%rdi	used to pass 1st argument	No
%r8	used to pass 5th argument	No
%r9	used to pass 6th argument	No
%r10-r11	temporary	No
%r12-r15	callee-saved registers	Yes

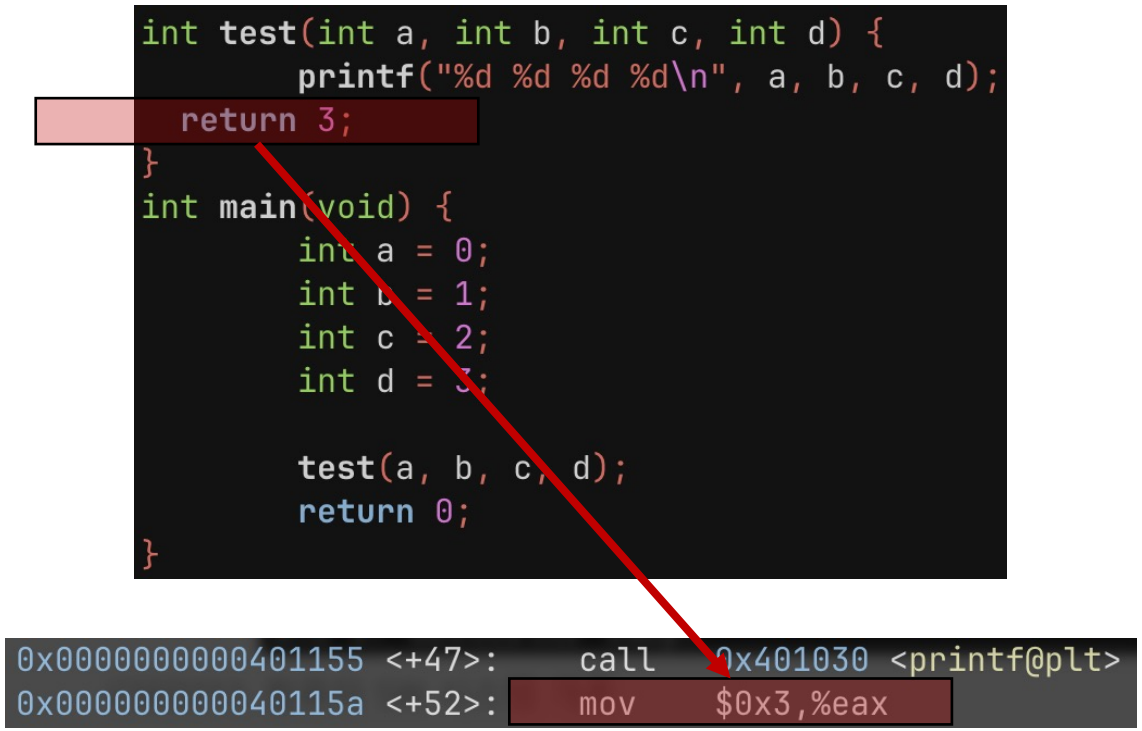


* http://6.s081.scripts.mit.edu/sp18/x86-64-architecture-guide.html

Assembly Language (Calling Convention)

- How does function return a value?
  - `rax` is used to return a value

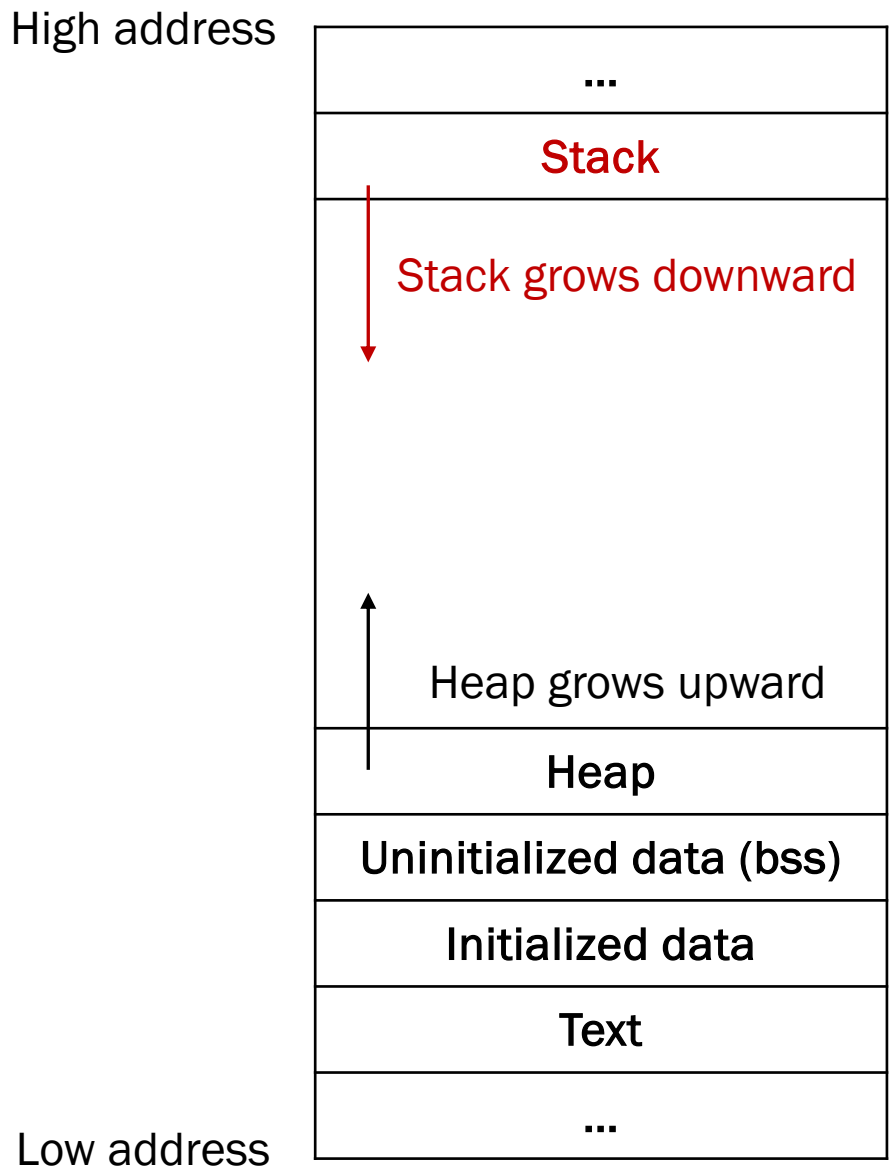
Register	Purpose	Saved across calls
%rax	temp register; return value	No
%rbx	callee-saved	Yes
%rcx	used to pass 4th argument	No
%rdx	used to pass 3rd argument	No
%rsp	stack pointer	Yes
%rbp	callee-saved; base pointer	Yes
%rsi	used to pass 2nd argument	No
%rdi	used to pass 1st argument	No
%r8	used to pass 5th argument	No
%r9	used to pass 6th argument	No
%r10-r11	temporary	No
%r12-r15	callee-saved registers	Yes



* http://6.s081.scripts.mit.edu/sp18/x86-64-architecture-guide.html

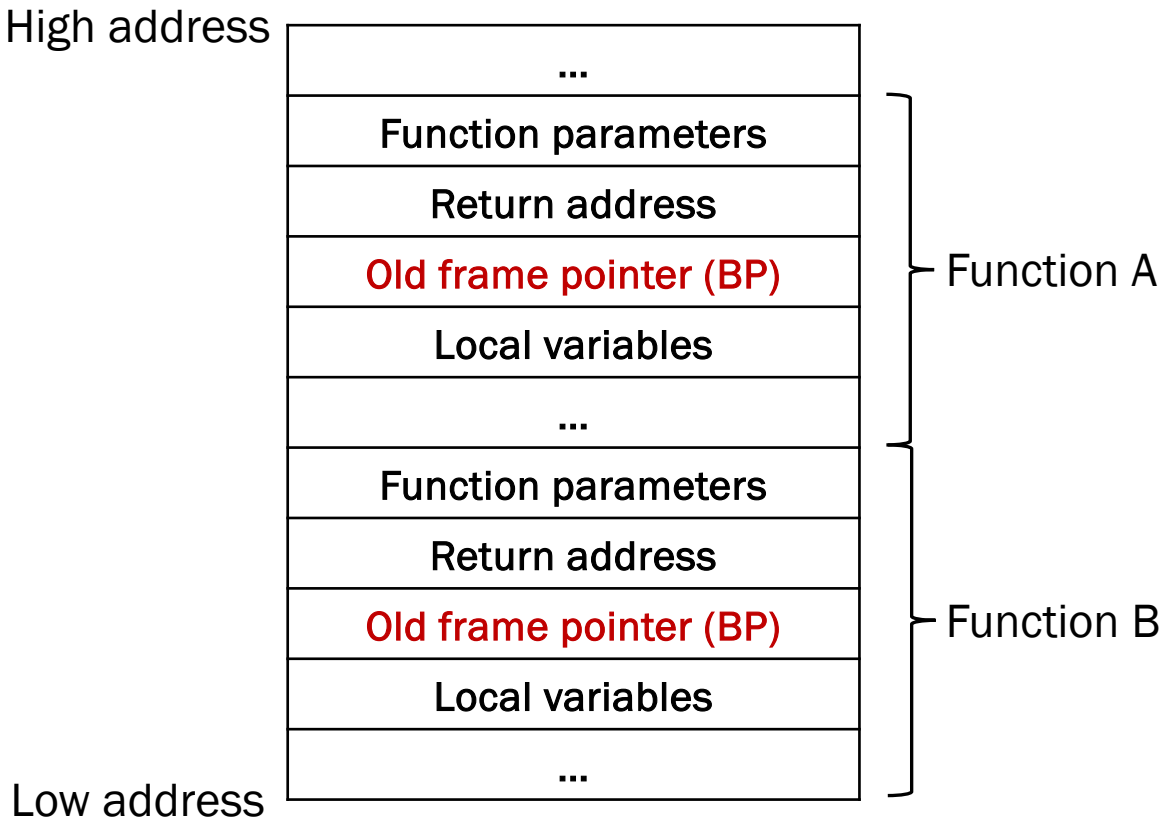
Memory Layout (Stack)

- Each program (binary) has its own address space
  - Stack is a LIFO (Last-In-First-Out) data structure
  - Local variables are stored in stack
  - Stack can grow (downward) and shrink (upward) as a program is being executed



Memory Layout (Stack)

- Stack frame
  - Stack stores information about subroutines (e.g., functions)
  - Stack is managed by
    - Pushing (prologue) and popping (epilogue) frame pointers (e.g., rbp, rsp)



Function prologue:
- push rbp
- mov rsp, rbp
- sub N, rsp

Function epilogue:
- mov rbp, rsp
- pop rbp
- ret

Homework (Bomb Lab)

- Bomb server will close at
  - 10/16 (Wed) 23:59 (midnight)

Homework (Report)

- Deadline: 10/16 (Wed) 23:59 (midnight)
- You need to
  - Explain how you defuse bomb in the report
  - Follow file name format, {student #}.pdf (No ZIP file)