Computer Security

tasi@tdohacker.org

Outline

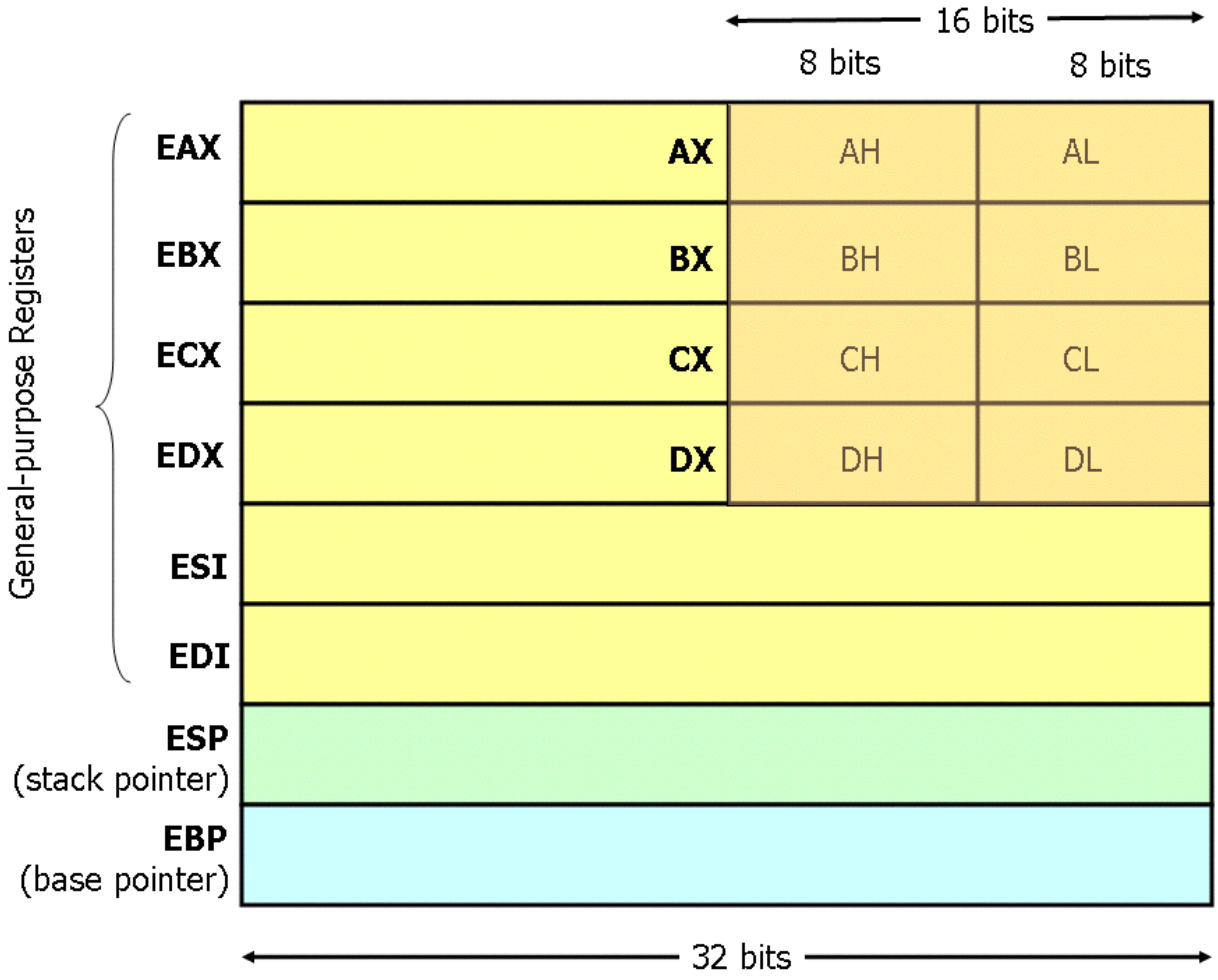
- Assembly
    - x86 Assembly
- Basic Reversing

x86 Assembly

Opcode	Operand
mov	ECX, 0x0A
push	EDX
Pop	EDX

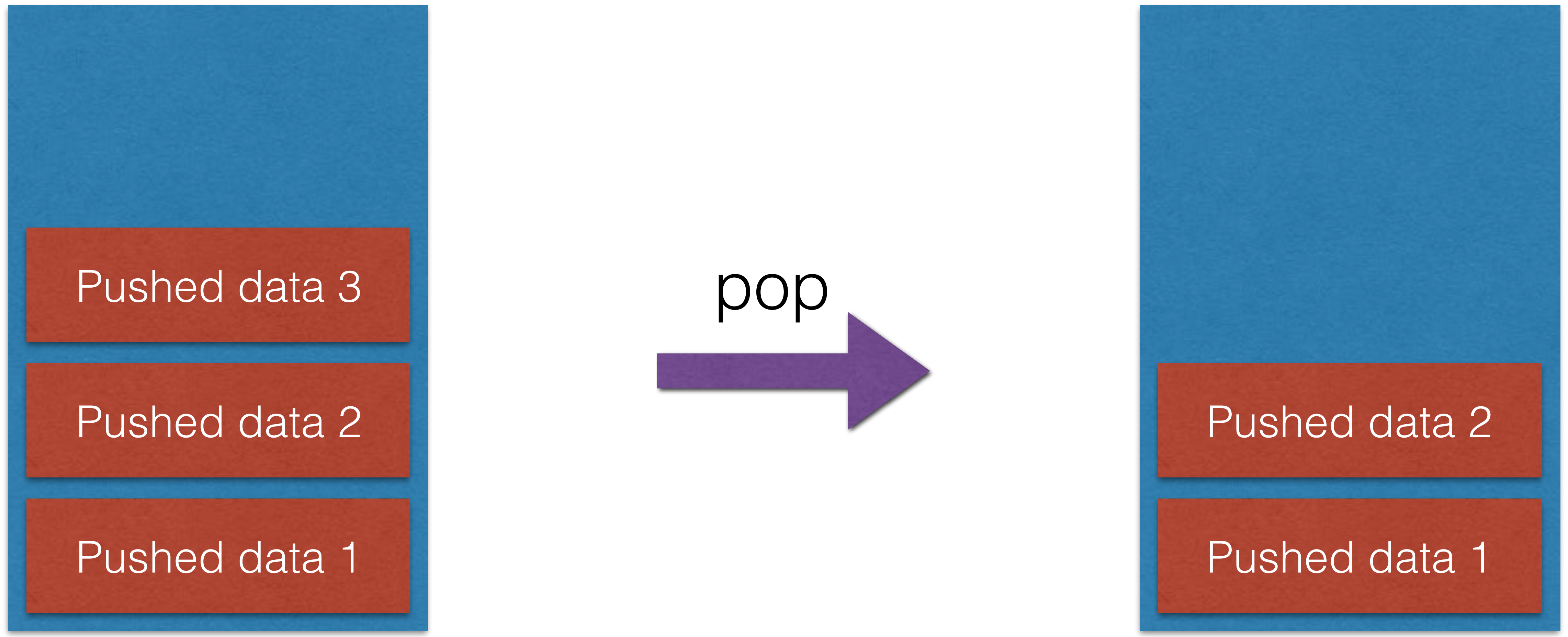
Registers

- General-Purpose Registers:
  - EAX, ECX, EDX, EBX, ESP, EBP, ESI, EDI
- Segment Registers :
  - SS, CS, DS, ES, FS, GS
- EFLAGS :
  - CF, PF, AF, ZF, …
- EIP
- SSE, MMX, FPU, debug


16 bits
8 bits
8 bits
General-purpose Registers
EAX
AX
AH
AL
EBX
BX
BH
BL
ECX
CX
CH
CL
EDX
DX
DH
DL
ESI
EDI
ESP
(stack pointer)
EBP
(base pointer)
32 bits

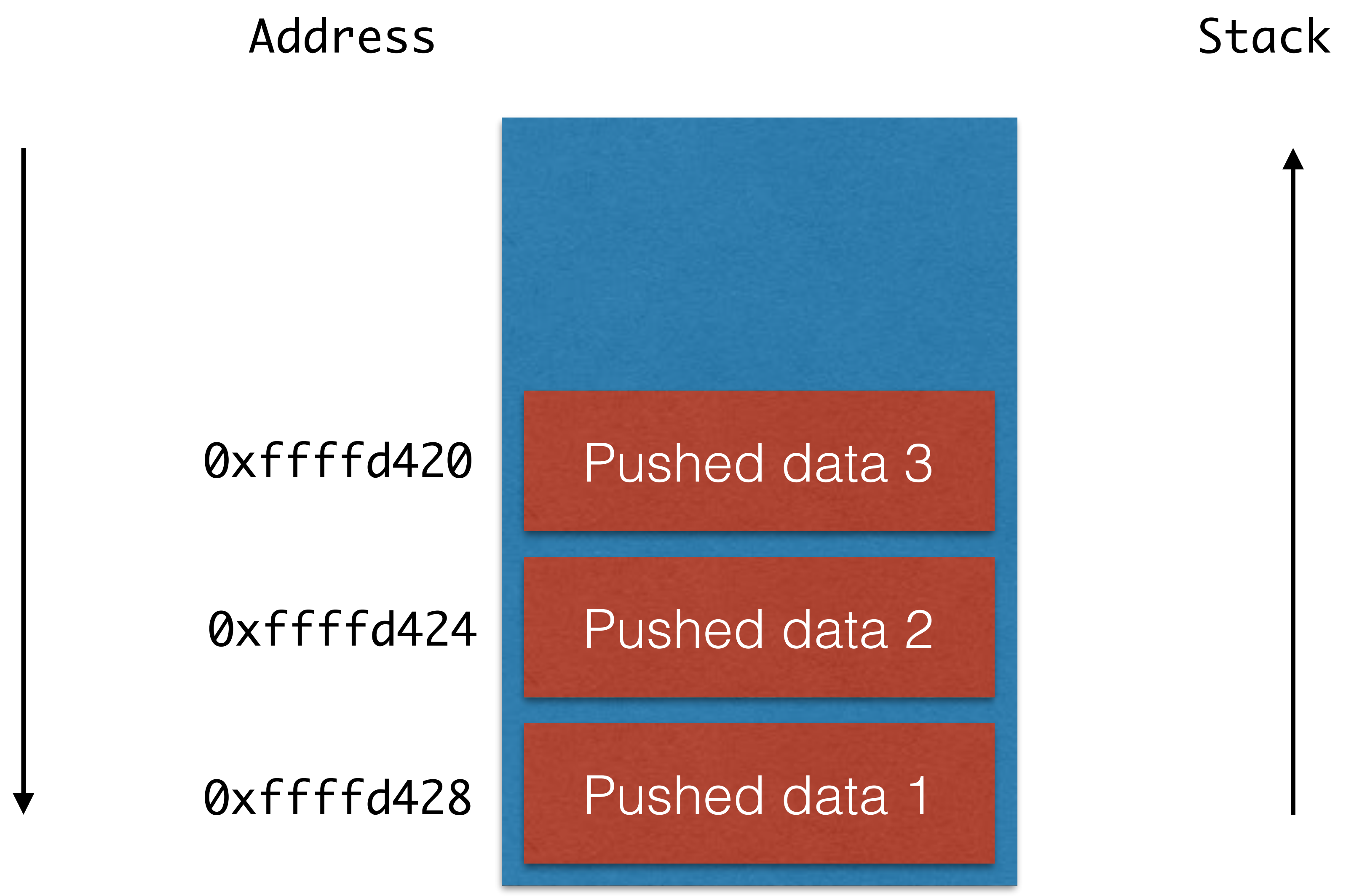
Stack

- A Region in Memory Where the Data is added / removed in last-in-first-out manner



Stack

- Stack grows towards the lower address



Calling Convention

- stdcall
    - 參數 push 入 stack 傳遞，從最後一個參數 push，Callee 負責還原 stack
- cdecl
    - 參數 push 入 stack 傳遞，從最後一個參數 push，Caller 負責還原 stack

Calling Convention

- fastcall
  - 參數放入暫存器來傳遞，Callee 負責還原 stack
    - e.g. ECX, EDX … (MSVC)
- thiscall
  - C++ 呼叫成員函數時，ECX 會放入 this 位置，Callee 負責還原 stack

Calling Convention

- cdecl

```
main:
…
…
push 0x01
push 0x02
call myfunc
add esp, 8
…
```

```
myfunc:
push ebp
…
…
…
leave
ret
```

Calling Convention

- cdecl

```
main:
…
…
push 0x01
push 0x02
call myfunc
add esp, 8
…
```

```
myfunc:
push ebp
…
…
…
leave
ret
```

Calling Convention

- fastcall

```
main:
…
…
mov edx, 1
mov ecx, 2
call myfunc
…
```

```
myfunc:
push ebp
…
…
…
leave
ret 8
```

Calling Convention

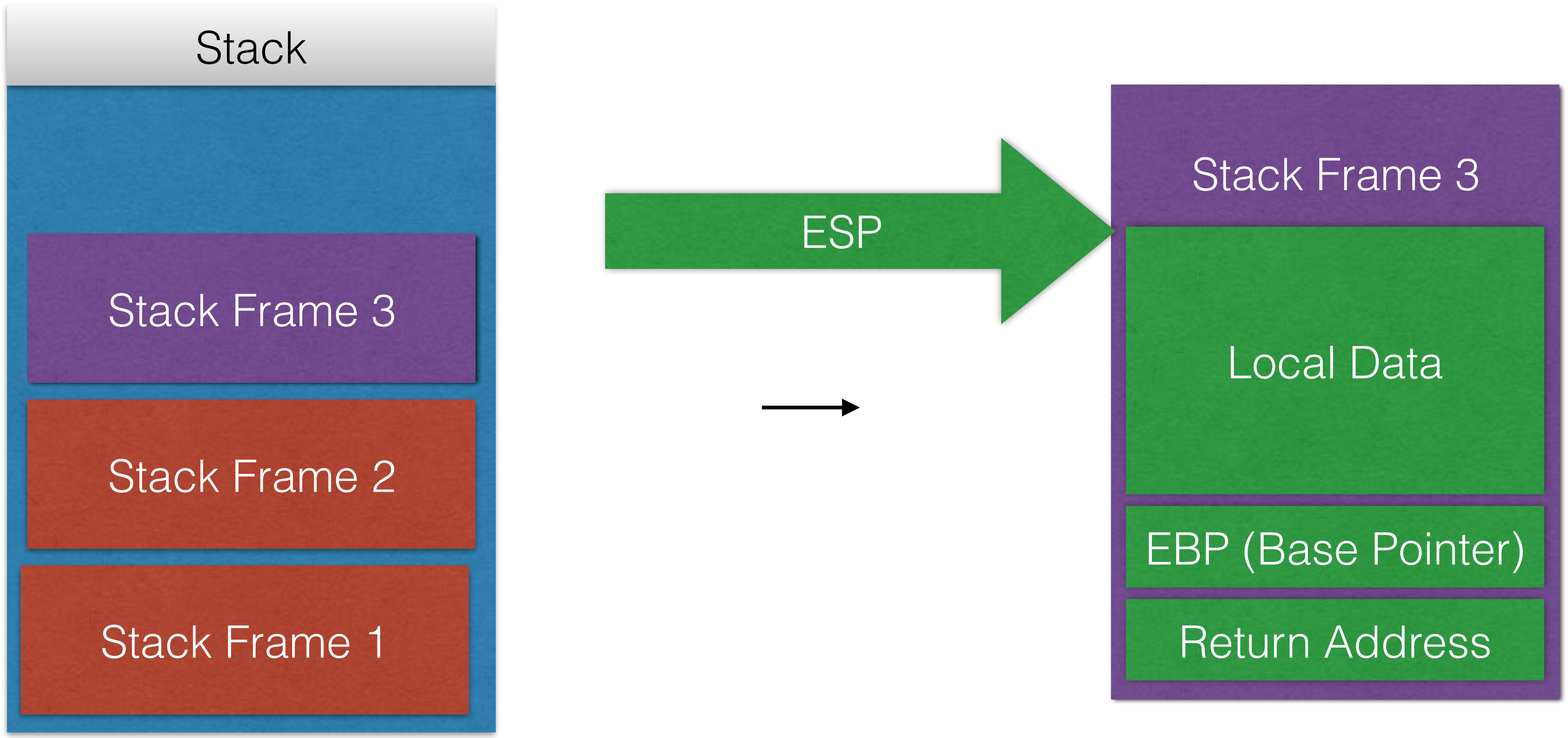
- thiscall

```
main:
…
…
push 0x01
push 0x02
mov ecx, 0x????????
call myfunc
…
```

```
myfunc:
push ebp
…
…
…
leave
ret 8
```

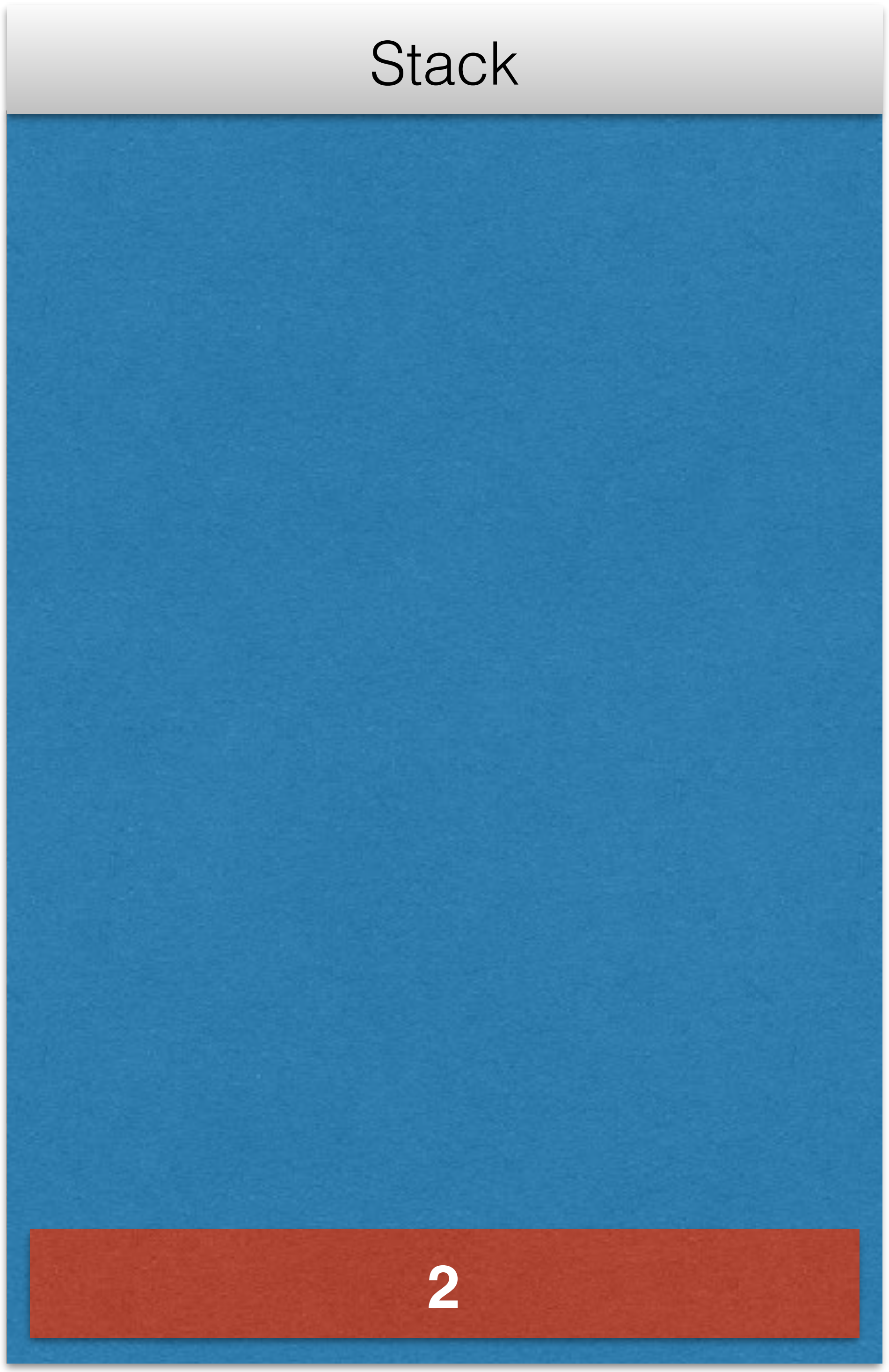
Stack Frame

- Stack 主要為一塊快的 StackFrame 所組成，當呼叫新的函數，Stack 便會push上一塊該函數的 StackFrame ，離開函數時則 pop 掉。



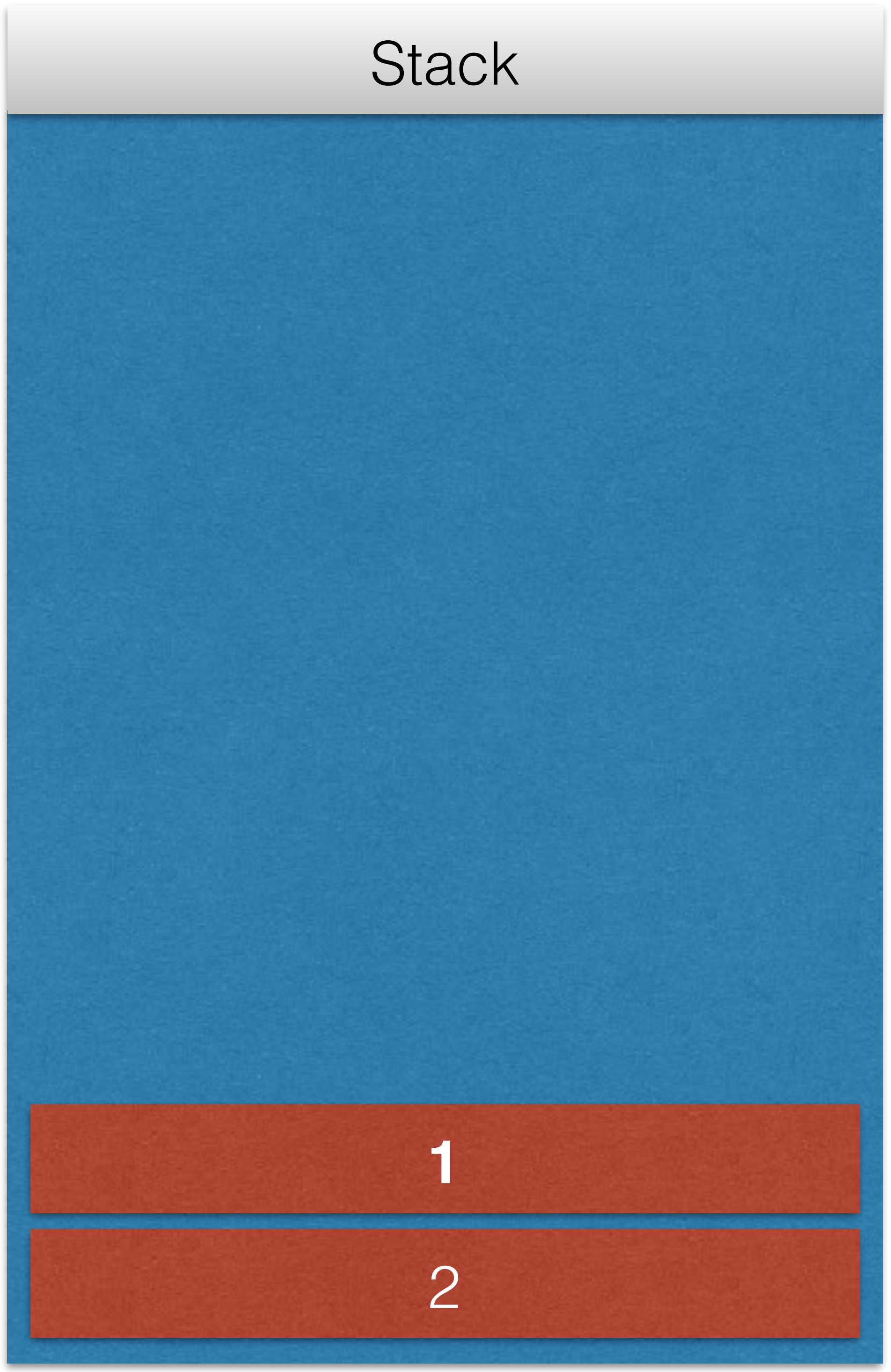
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



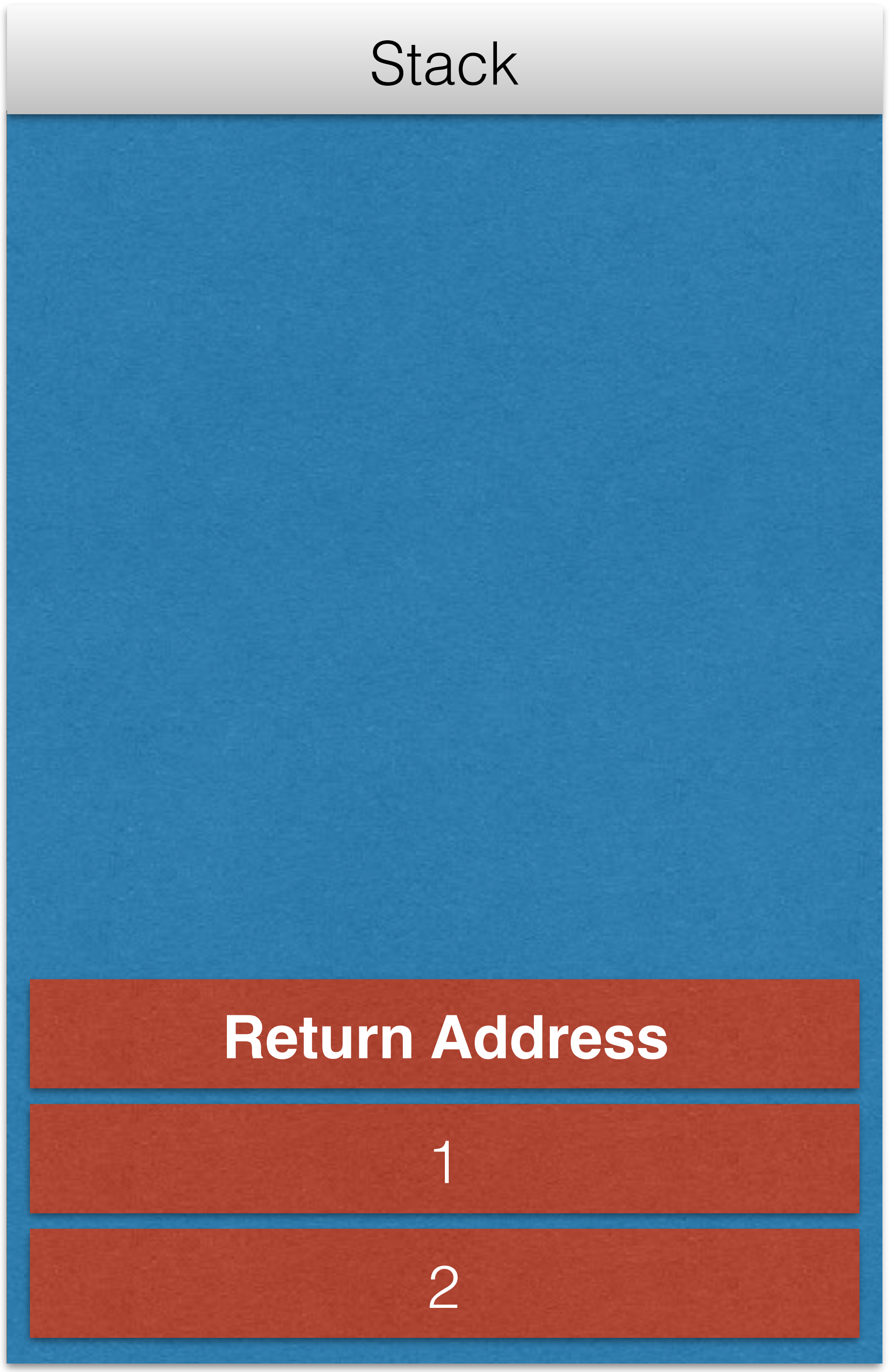
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



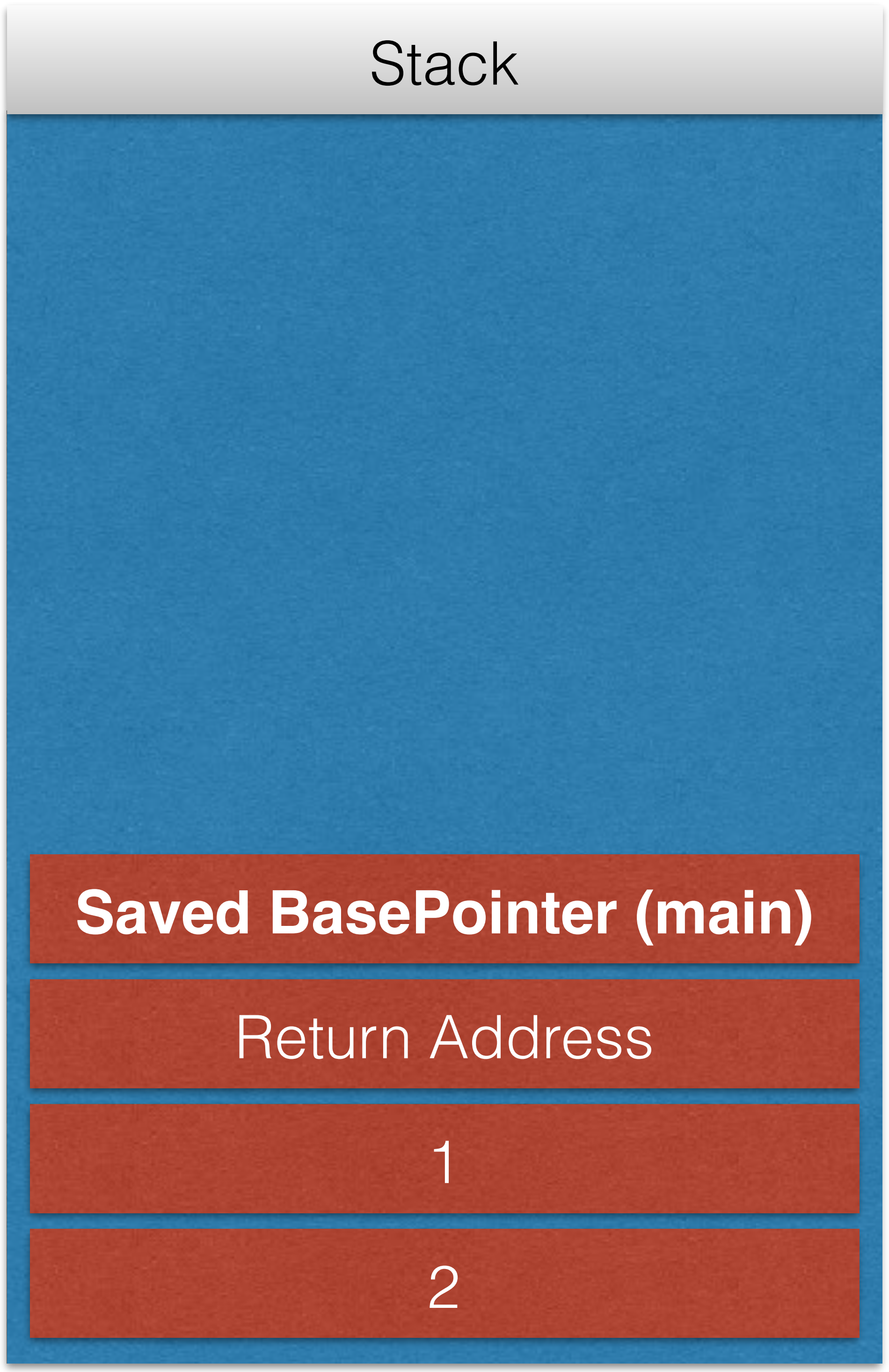
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



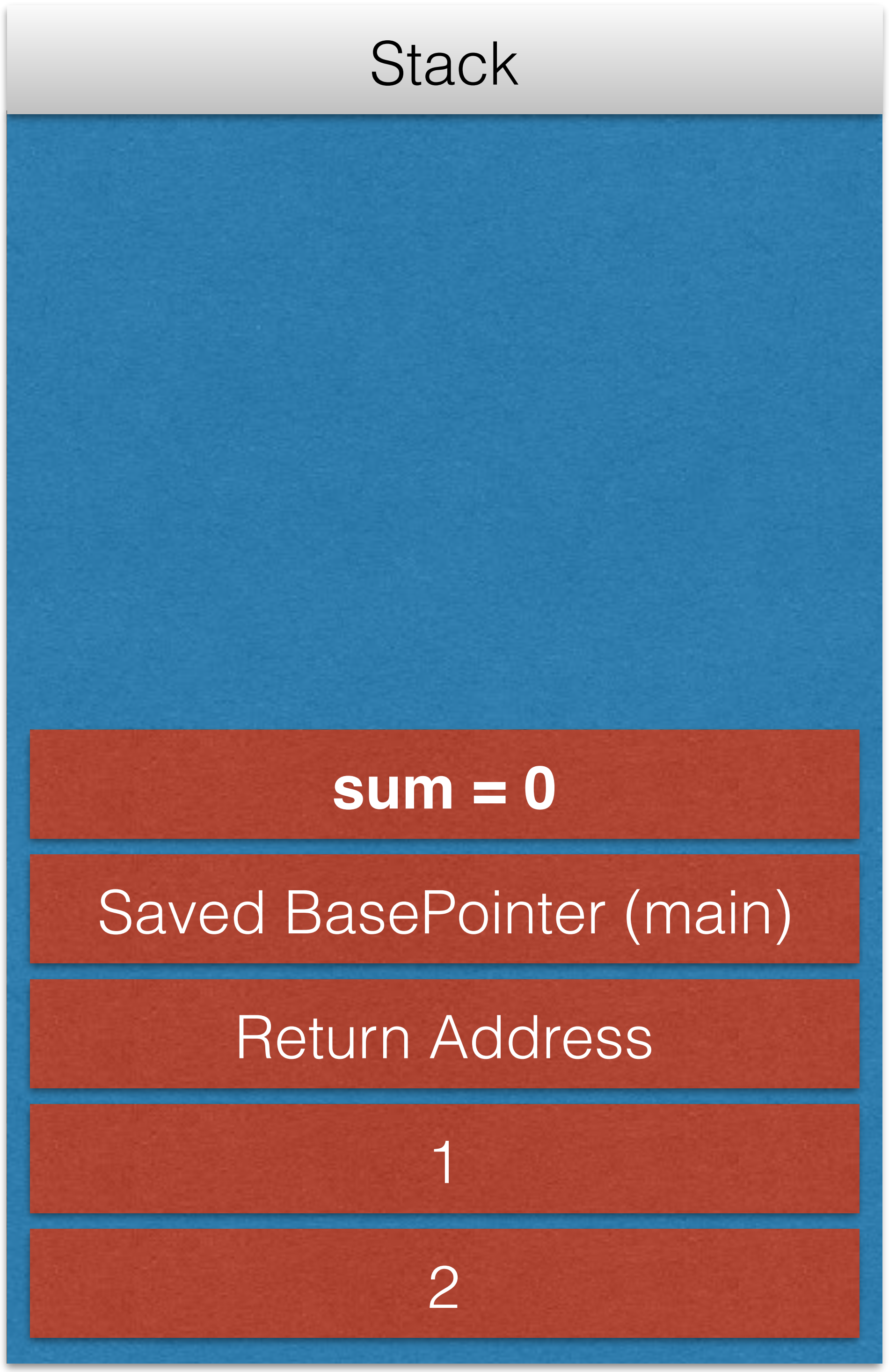
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



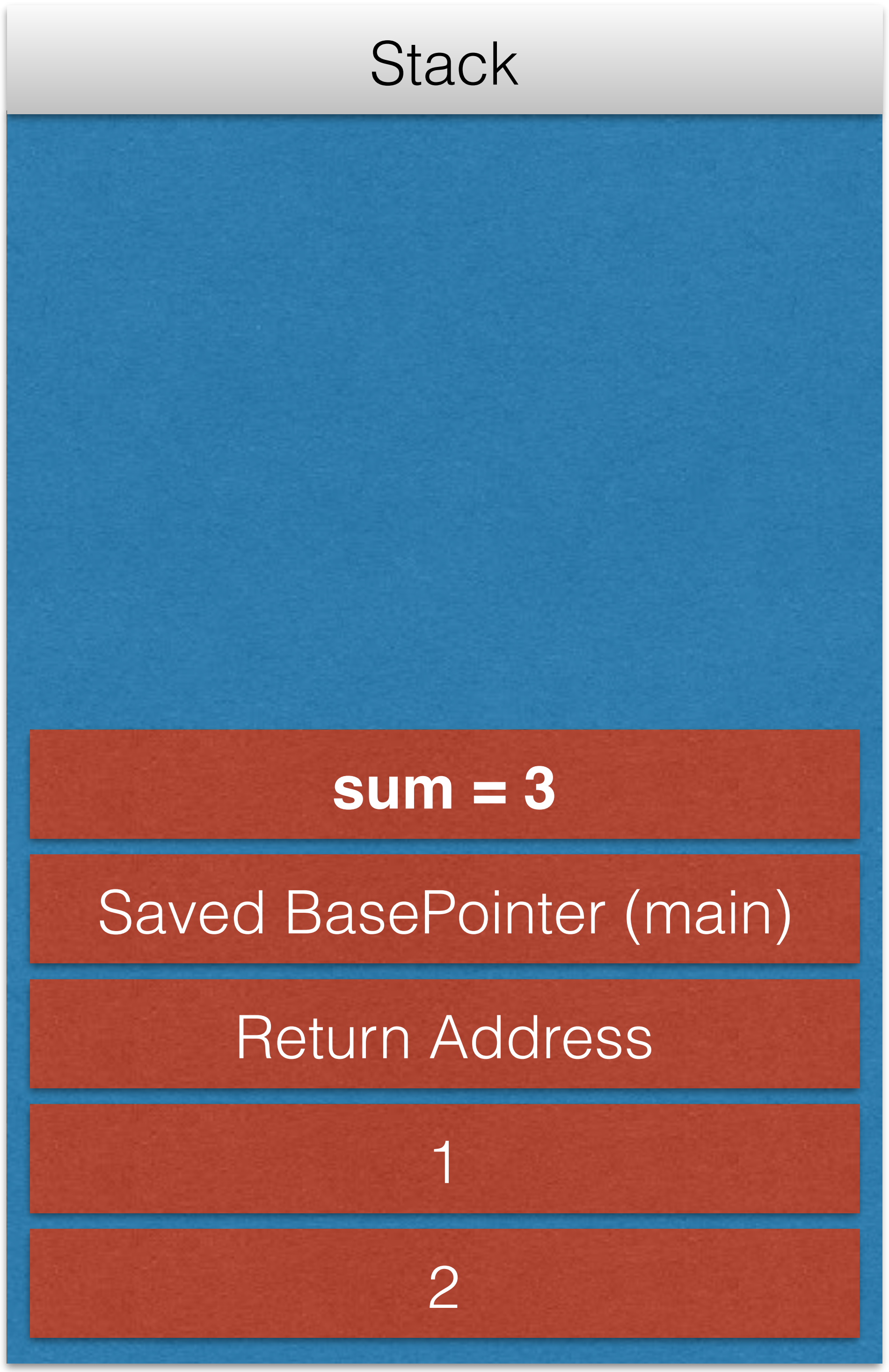
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



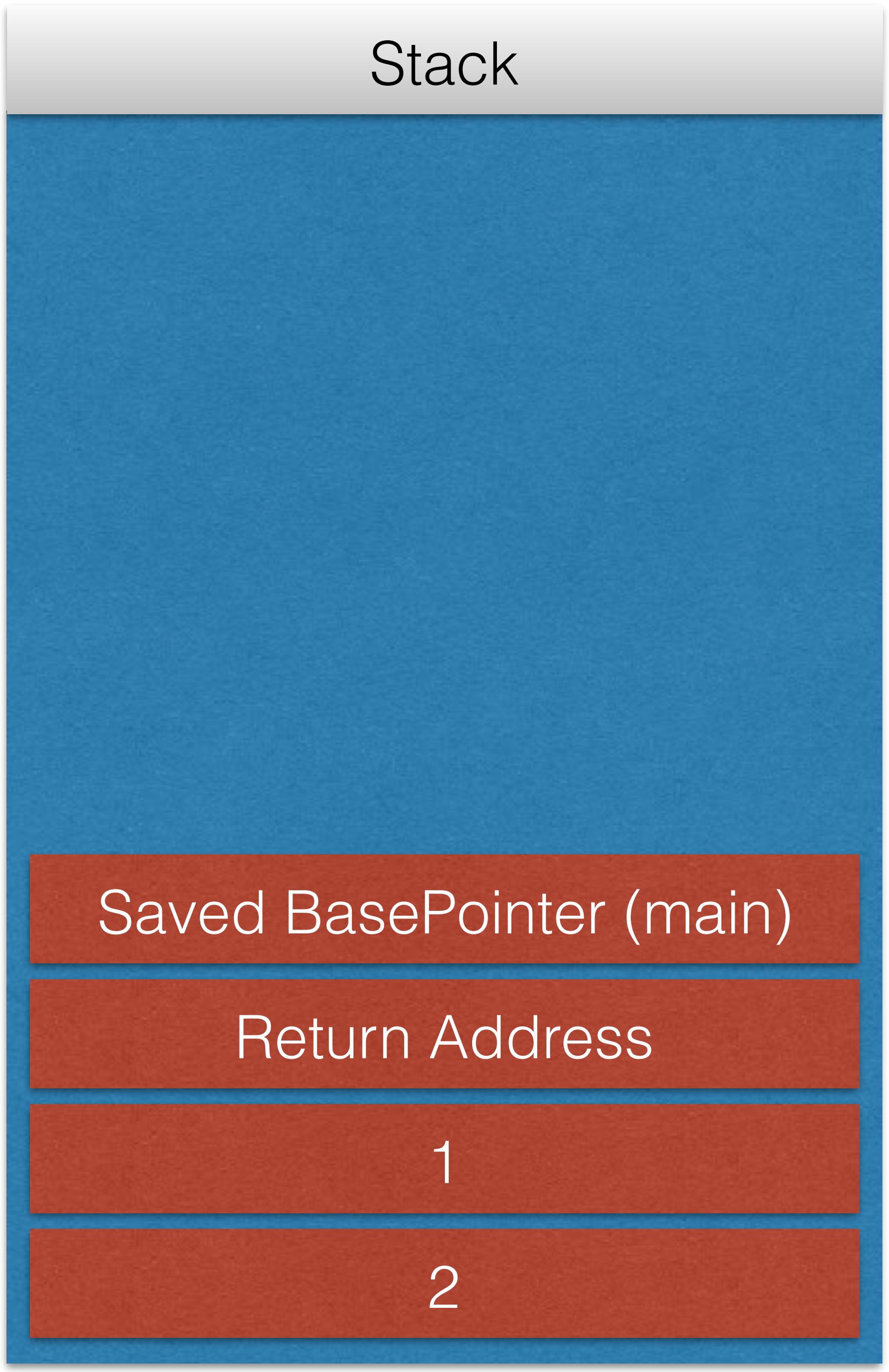
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



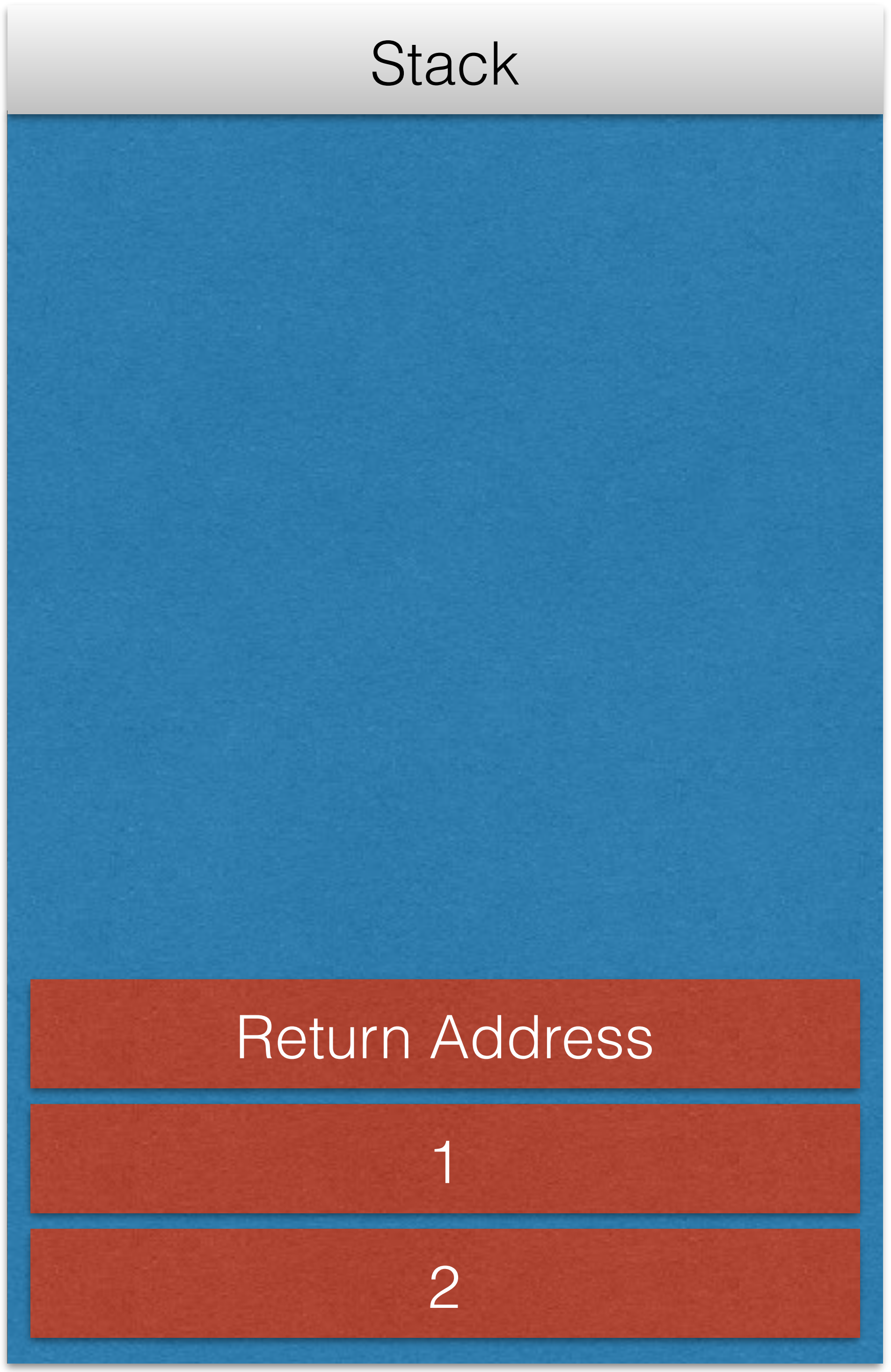
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



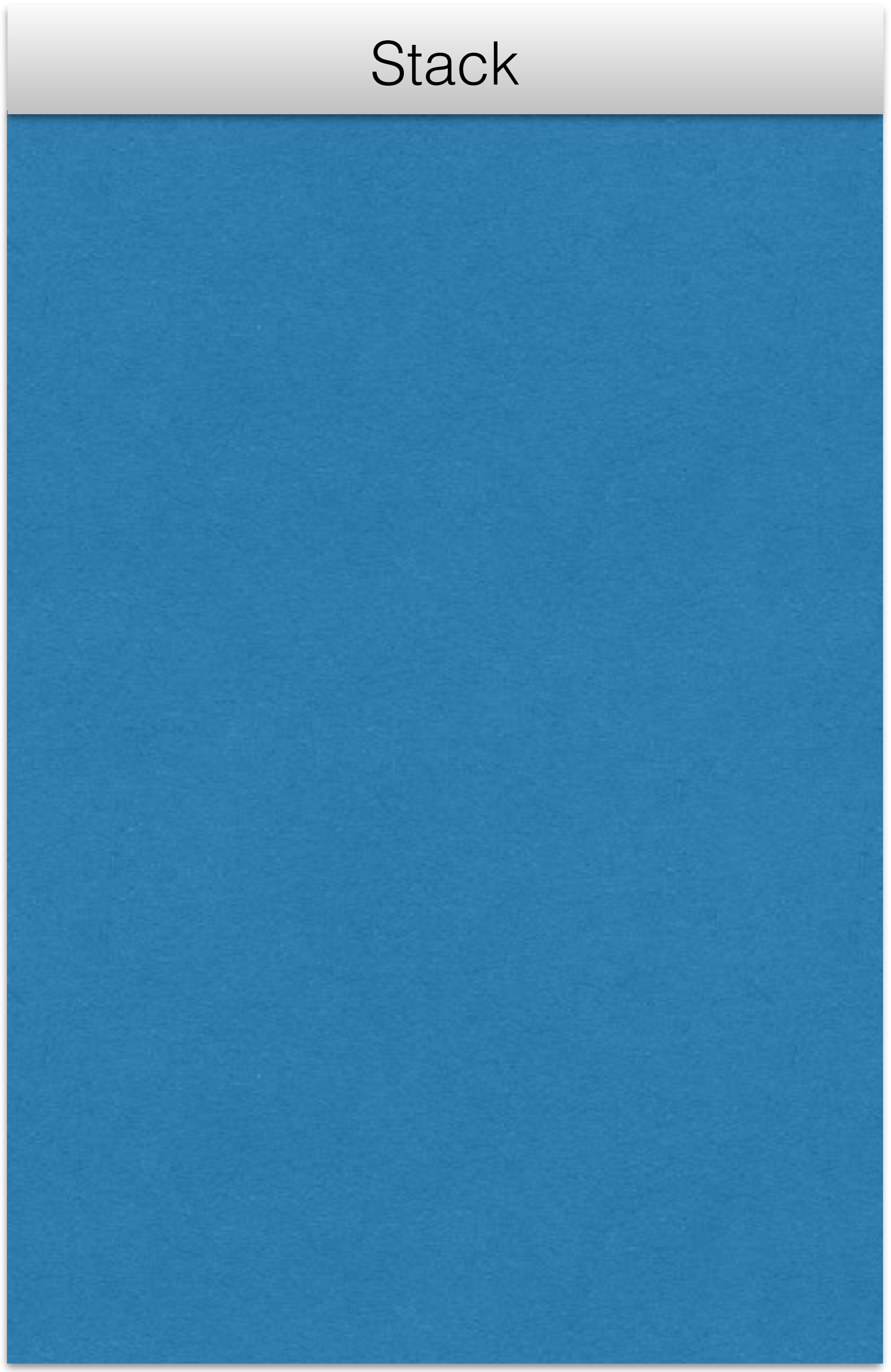
Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



Stack Frame

```
int main() {
    myfunc(1, 2);
    return 0;
}

int myfunc(int a, int b) {
    int sum = 0;
    sum = a + b;
    return sum;
}
```



Instructions

mov	Copy Value from src to dest
push	push value onto stack
pop	pop out value to operand from stack
cmp	compare values
jmp	Jump to the address (jmp, je, jne, jl, jg, ja, jb…)
call	call function
ret	pop the value from stack to eip (return to caller function)
nop	do nothing

Instructions

add	Addition
sub	Subtraction
mul	multiplication
div	division
xor	Bits Operation
and	Bits Operation
or	Bits Operation
rep/repe/repz/repne/repnz	repeat string operation prefix

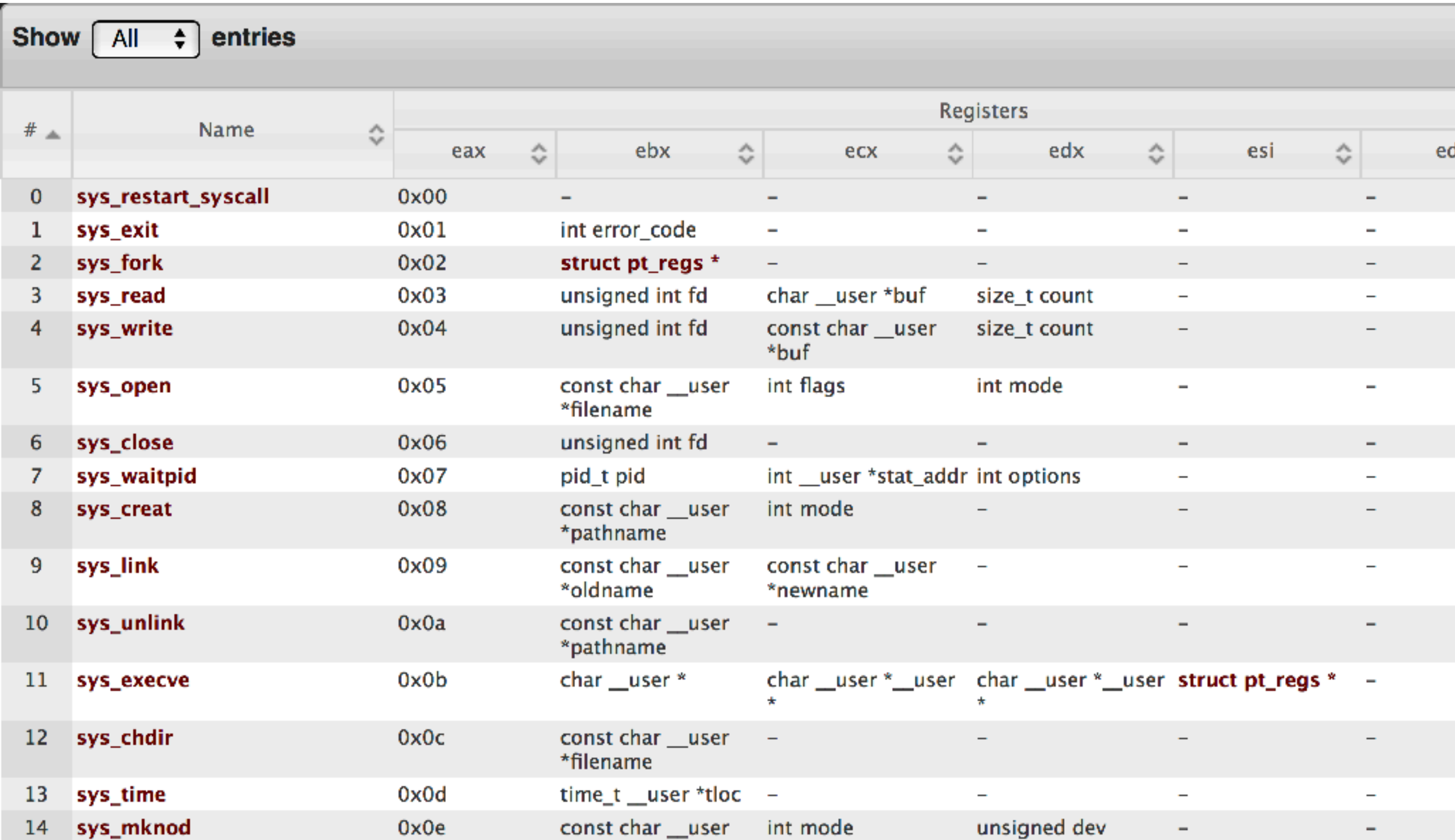
System Call

- 為 Kernel 與 User Layer 溝通的介面，當 User 層的程式需要OS提供服務的時候會用到，例如 IO、Process … 等等。

System Call

- 在 Linux 中每一個 system call 會有一個編號，程式要乎叫 system call 時，會把編號放入 EAX 中，參數部分放入其他像是 EDX, ECX 這些暫存器，最後由 int 0x80 來觸發呼叫 system call

System Call



Show All entries

#	Name	Registers					
		eax	ebx	ecx	edx	esi	ed
0	sys_restart_syscall	0x00	–	–	–	–	–
1	sys_exit	0x01	int error_code	–	–	–	–
2	sys_fork	0x02	struct pt_regs *	–	–	–	–
3	sys_read	0x03	unsigned int fd	char __user *buf	size_t count	–	–
4	sys_write	0x04	unsigned int fd	const char __user *buf	size_t count	–	–
5	sys_open	0x05	const char __user *filename	int flags	int mode	–	–
6	sys_close	0x06	unsigned int fd	–	–	–	–
7	sys_waitpid	0x07	pid_t pid	int __user *stat_addr	int options	–	–
8	sys_creat	0x08	const char __user *pathname	int mode	–	–	–
9	sys_link	0x09	const char __user *oldname	const char __user *newname	–	–	–
10	sys_unlink	0x0a	const char __user *pathname	–	–	–	–
11	sys_execve	0x0b	char __user *	char __user *__user *	char __user *__user *	struct pt_regs *	–
12	sys_chdir	0x0c	const char __user *filename	–	–	–	–
13	sys_time	0x0d	time_t __user *tloc	–	–	–	–
14	sys_mknod	0x0e	const char __user	int mode	unsigned dev	–	–

System Call

```
xor ecx, ecx          ;; ecx = 0
mul ecx               ;; eax = 0, edx = 0
push ecx              ;; \x00
push 0x68732f2f       ;; hs//
push 0x6e69622f       ;; nib/
mov ebx, esp          ;; ebx = “/bin//sh\x00”
mov al, 11            ;; eax = 11 (sys_execve)
int 0x80
```

Variables

Exercise1

```
080483db <main>:
 80483db: 55                   push   ebp
 80483dc: 89 e5                mov    ebp,esp
 80483de: 83 ec 04             sub    esp,0x4
 80483e1: c7 45 fc 33 33 23 00 mov    DWORD PTR [ebp-0x4],0x233333
 80483e8: b8 00 00 00 00       mov    eax,0x0
 80483ed: c9                   leave
 80483ee: c3                   ret
```

Exercise2

```
080483db <main>:
 80483db: 55                        push   ebp
 80483dc: 89 e5                     mov    ebp,esp
 80483de: 83 ec 04                  sub    esp,0x4
 80483e1: c6 45 ff 41               mov    BYTE PTR [ebp-0x1],0x41
 80483e5: b8 00 00 00 00            mov    eax,0x0
 80483ea: c9                        leave
 80483eb: c3                        ret
```

Exercise3

```
080483db <main>:
 80483db: 55                     push   ebp
 80483dc: 89 e5                  mov    ebp,esp
 80483de: 83 ec 04               sub    esp,0x4
 80483e1: 66 c7 45 fe 0a 00      mov    WORD PTR [ebp-0x2],0xa
 80483e7: b8 00 00 00 00         mov    eax,0x0
 80483ec: c9                     leave
 80483ed: c3                     ret
```

Exercise4

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 f8 0a 00 00 00    mov    DWORD PTR [ebp-0x8],0xa
 80483e8: 8d 45 f8                lea    eax,[ebp-0x8]
 80483eb: 89 45 fc                mov    DWORD PTR [ebp-0x4],eax
 80483ee: b8 00 00 00 00          mov    eax,0x0
 80483f3: c9                      leave
 80483f4: c3                      ret
```

Control Flow

Exercise5

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 04                sub    esp,0x4
 80483e1: c7 45 fc 0a 00 00 00    mov    DWORD PTR [ebp-0x4],0xa
 80483e8: 83 7d fc 7b             cmp    DWORD PTR [ebp-0x4],0x7b
 80483ec: 75 07                   jne    80483f5 <main+0x1a>
 80483ee: b8 01 00 00 00          mov    eax,0x1
 80483f3: eb 05                   jmp    80483fa <main+0x1f>
 80483f5: b8 00 00 00 00          mov    eax,0x0
 80483fa: c9                      leave
 80483fb: c3                      ret
```

Exercise6

```
080483db <main>:
 80483db: 55                        push   ebp
 80483dc: 89 e5                     mov    ebp,esp
 80483de: 83 ec 04                  sub    esp,0x4
 80483e1: c7 45 fc 0a 00 00 00      mov    DWORD PTR [ebp-0x4],0xa
 80483e8: 83 7d fc 13               cmp    DWORD PTR [ebp-0x4],0x13
 80483ec: 7f 0d                     jg     80483fb <main+0x20>
 80483ee: 83 7d fc 05               cmp    DWORD PTR [ebp-0x4],0x5
 80483f2: 7e 07                     jle    80483fb <main+0x20>
 80483f4: b8 00 00 00 00            mov    eax,0x0
 80483f9: eb 05                     jmp    8048400 <main+0x25>
 80483fb: b8 01 00 00 00            mov    eax,0x1
 8048400: c9                        leave
 8048401: c3                        ret
```

Exercise7

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 0a 00 00 00    mov    DWORD PTR [ebp-0x4],0xa
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 83 7d fc 63             cmp    DWORD PTR [ebp-0x4],0x63
 80483f3: 7f 09                   jg     80483fe <main+0x23>
 80483f5: c7 45 f8 33 33 02 00    mov    DWORD PTR [ebp-0x8],0x23333
 80483fc: eb 07                   jmp    8048405 <main+0x2a>
 80483fe: c7 45 f8 66 66 06 00    mov    DWORD PTR [ebp-0x8],0x66666
 8048405: b8 00 00 00 00          mov    eax,0x0
 804840a: c9                      leave
 804840b: c3                      ret
```

Exercise8 HINT: switch

```
080483db <main>:
 80483db:   55                      push   ebp
 80483dc:   89 e5                   mov    ebp,esp
 80483de:   83 ec 08                sub    esp,0x8
 80483e1:   c7 45 fc 0a 00 00 00    mov    DWORD PTR [ebp-0x4],0xa
 80483e8:   8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483eb:   83 f8 01                cmp    eax,0x1
 80483ee:   74 0f                   je     80483ff <main+0x24>
 80483f0:   83 f8 02                cmp    eax,0x2
 80483f3:   74 10                   je     8048405 <main+0x2a>
 80483f5:   85 c0                   test   eax,eax
 80483f7:   75 12                   jne    804840b <main+0x30>
 80483f9:   c6 45 fb 61             mov    BYTE PTR [ebp-0x5],0x61
 80483fd:   eb 11                   jmp    8048410 <main+0x35>
 80483ff:   c6 45 fb 62             mov    BYTE PTR [ebp-0x5],0x62
 8048403:   eb 0b                   jmp    8048410 <main+0x35>
 8048405:   c6 45 fb 63             mov    BYTE PTR [ebp-0x5],0x63
 8048409:   eb 05                   jmp    8048410 <main+0x35>
 804840b:   c6 45 fb 64             mov    BYTE PTR [ebp-0x5],0x64
 804840f:   90                      nop
 8048410:   b8 00 00 00 00          mov    eax,0x0
 8048415:   c9                      leave
 8048416:   c3                      ret
```

Exercise9

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483e8: c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
 80483ef: eb 0a                   jmp    80483fb <main+0x20>
 80483f1: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f4: 01 45 f8                add    DWORD PTR [ebp-0x8],eax
 80483f7: 83 45 fc 01             add    DWORD PTR [ebp-0x4],0x1
 80483fb: 83 7d fc 1d             cmp    DWORD PTR [ebp-0x4],0x1d
 80483ff: 7e f0                   jle    80483f1 <main+0x16>
 8048401: b8 00 00 00 00          mov    eax,0x0
 8048406: c9                      leave
 8048407: c3                      ret
```

Exercise10

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483e8: c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
 80483ef: 8b 45 f8                mov    eax,DWORD PTR [ebp-0x8]
 80483f2: 01 45 fc                add    DWORD PTR [ebp-0x4],eax
 80483f5: 83 7d f8 1d             cmp    DWORD PTR [ebp-0x8],0x1d
 80483f9: 7e f4                   jle    80483ef <main+0x14>
 80483fb: b8 00 00 00 00          mov    eax,0x0
 8048400: c9                      leave
 8048401: c3                      ret
```

Exercise11 HINT: imul is multiplication

```
080483db <main>:
 80483db:   55                      push   ebp
 80483dc:   89 e5                   mov    ebp,esp
 80483de:   83 ec 0c                sub    esp,0xc
 80483e1:   c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483e8:   c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
 80483ef:   eb 23                   jmp    8048414 <main+0x39>
 80483f1:   c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483f8:   eb 0e                   jmp    8048408 <main+0x2d>
 80483fa:   8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483fd:   0f af 45 f8             imul   eax,DWORD PTR [ebp-0x8]
 8048401:   89 45 f4                mov    DWORD PTR [ebp-0xc],eax
 8048404:   83 45 f8 01             add    DWORD PTR [ebp-0x8],0x1
 8048408:   8b 45 f8                mov    eax,DWORD PTR [ebp-0x8]
 804840b:   3b 45 fc                cmp    eax,DWORD PTR [ebp-0x4]
 804840e:   7c ea                   jl     80483fa <main+0x1f>
 8048410:   83 45 fc 01             add    DWORD PTR [ebp-0x4],0x1
 8048414:   83 7d fc 09             cmp    DWORD PTR [ebp-0x4],0x9
 8048418:   7e d7                   jle    80483f1 <main+0x16>
 804841a:   b8 00 00 00 00          mov    eax,0x0
 804841f:   c9                      leave
 8048420:   c3                      ret
```

Exercise12 HINT: goto

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 0a 00 00 00    mov    DWORD PTR [ebp-0x4],0xa
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 83 7d fc 0a             cmp    DWORD PTR [ebp-0x4],0xa
 80483f3: 75 08                   jne    80483fd <main+0x22>
 80483f5: 90                      nop
 80483f6: b8 01 00 00 00          mov    eax,0x1
 80483fb: eb 0d                   jmp    804840a <main+0x2f>
 80483fd: 90                      nop
 80483fe: c7 45 f8 02 00 00 00    mov    DWORD PTR [ebp-0x8],0x2
 8048405: b8 00 00 00 00          mov    eax,0x0
 804840a: c9                      leave
 804840b: c3                      ret
```

Exercise13

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 04                sub    esp,0x4
 80483e1: c7 45 fc 00 00 00 00    mov    DWORD PTR [ebp-0x4],0x0
 80483e8: eb 0a                   jmp    80483f4 <main+0x19>
 80483ea: 83 7d fc 14             cmp    DWORD PTR [ebp-0x4],0x14
 80483ee: 74 0c                   je     80483fc <main+0x21>
 80483f0: 83 45 fc 01             add    DWORD PTR [ebp-0x4],0x1
 80483f4: 83 7d fc 63             cmp    DWORD PTR [ebp-0x4],0x63
 80483f8: 7e f0                   jle    80483ea <main+0xf>
 80483fa: eb 01                   jmp    80483fd <main+0x22>
 80483fc: 90                      nop
 80483fd: b8 00 00 00 00          mov    eax,0x0
 8048402: c9                      leave
 8048403: c3                      ret
```

Arithmetic

Exercise14

```
080483db <main>:
 80483db:   55                     push   ebp
 80483dc:   89 e5                  mov    ebp,esp
 80483de:   83 ec 0c               sub    esp,0xc
 80483e1:   c7 45 fc 05 00 00 00   mov    DWORD PTR [ebp-0x4],0x5
 80483e8:   c7 45 f8 05 00 00 00   mov    DWORD PTR [ebp-0x8],0x5
 80483ef:   c7 45 f4 00 00 00 00   mov    DWORD PTR [ebp-0xc],0x0
 80483f6:   8b 55 fc               mov    edx,DWORD PTR [ebp-0x4]
 80483f9:   8b 45 f8               mov    eax,DWORD PTR [ebp-0x8]
 80483fc:   01 d0                  add    eax,edx
 80483fe:   89 45 f4               mov    DWORD PTR [ebp-0xc],eax
 8048401:   b8 00 00 00 00         mov    eax,0x0
 8048406:   c9                     leave
 8048407:   c3                     ret
```

Exercise15

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 0c                sub    esp,0xc
 80483e1: c7 45 fc 05 00 00 00    mov    DWORD PTR [ebp-0x4],0x5
 80483e8: c7 45 f8 05 00 00 00    mov    DWORD PTR [ebp-0x8],0x5
 80483ef: c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483f6: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f9: 2b 45 f8                sub    eax,DWORD PTR [ebp-0x8]
 80483fc: 89 45 f4                mov    DWORD PTR [ebp-0xc],eax
 80483ff: b8 00 00 00 00          mov    eax,0x0
 8048404: c9                      leave
 8048405: c3                      ret
```

Exercise16

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 0c                sub    esp,0xc
 80483e1: c7 45 fc 05 00 00 00    mov    DWORD PTR [ebp-0x4],0x5
 80483e8: c7 45 f8 05 00 00 00    mov    DWORD PTR [ebp-0x8],0x5
 80483ef: c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483f6: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f9: 0f af 45 f8             imul   eax,DWORD PTR [ebp-0x8]
 80483fd: 89 45 f4                mov    DWORD PTR [ebp-0xc],eax
 8048400: b8 00 00 00 00          mov    eax,0x0
 8048405: c9                      leave
 8048406: c3                      ret
```

Exercise17

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 0c                sub    esp,0xc
 80483e1: c7 45 fc 05 00 00 00    mov    DWORD PTR [ebp-0x4],0x5
 80483e8: c7 45 f8 05 00 00 00    mov    DWORD PTR [ebp-0x8],0x5
 80483ef: c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483f6: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f9: 99                      cdq
 80483fa: f7 7d f8                idiv   DWORD PTR [ebp-0x8]
 80483fd: 89 45 f4                mov    DWORD PTR [ebp-0xc],eax
 8048400: b8 00 00 00 00          mov    eax,0x0
 8048405: c9                      leave
 8048406: c3                      ret
```

Exercise18

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 0c                sub    esp,0xc
 80483e1: c7 45 fc 05 00 00 00    mov    DWORD PTR [ebp-0x4],0x5
 80483e8: c7 45 f8 05 00 00 00    mov    DWORD PTR [ebp-0x8],0x5
 80483ef: c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483f6: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f9: 99                      cdq
 80483fa: f7 7d f8                idiv   DWORD PTR [ebp-0x8]
 80483fd: 89 55 f4                mov    DWORD PTR [ebp-0xc],edx
 8048400: b8 00 00 00 00          mov    eax,0x0
 8048405: c9                      leave
 8048406: c3                      ret
```

Bits Operation

Exercise19

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 0c                sub    esp,0xc
 80483e1: c7 45 fc 04 00 00 00    mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: c7 45 f4 00 00 00 00    mov    DWORD PTR [ebp-0xc],0x0
 80483f6: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f9: 83 e0 01                and    eax,0x1
 80483fc: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483ff: b8 00 00 00 00          mov    eax,0x0
 8048404: c9                      leave
 8048405: c3                      ret
```

Exercise20

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 04 00 00 00    mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f2: 0d 0f ff 00 00          or     eax,0xff0f
 80483f7: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483fa: b8 00 00 00 00          mov    eax,0x0
 80483ff: c9                      leave
 8048400: c3                      ret
```

Exercise21

```
080483db <main>:
 80483db: 55                    push   ebp
 80483dc: 89 e5                 mov    ebp,esp
 80483de: 83 ec 08              sub    esp,0x8
 80483e1: c7 45 fc 04 00 00 00  mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00  mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 8b 45 fc              mov    eax,DWORD PTR [ebp-0x4]
 80483f2: 34 ff                 xor    al,0xff
 80483f4: 89 45 f8              mov    DWORD PTR [ebp-0x8],eax
 80483f7: b8 00 00 00 00        mov    eax,0x0
 80483fc: c9                    leave
 80483fd: c3                    ret
```

Exercise22

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 04 00 00 00    mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f2: f7 d0                   not    eax
 80483f4: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483f7: b8 00 00 00 00          mov    eax,0x0
 80483fc: c9                      leave
 80483fd: c3                      ret
```

Exercise23

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 04 00 00 00    mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f2: d1 f8                   sar    eax,1
 80483f4: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483f7: b8 00 00 00 00          mov    eax,0x0
 80483fc: c9                      leave
 80483fd: c3                      ret
```

Exercise24

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 04 00 00 00    mov    DWORD PTR [ebp-0x4],0x4
 80483e8: c7 45 f8 00 00 00 00    mov    DWORD PTR [ebp-0x8],0x0
 80483ef: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483f2: c1 e0 03                shl    eax,0x3
 80483f5: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483f8: b8 00 00 00 00          mov    eax,0x0
 80483fd: c9                      leave
 80483fe: c3                      ret
```

Functions

Exercise25

```
0804846b <main>:
 804846b: 55                         push   ebp
 804846c: 89 e5                      mov    ebp,esp
 804846e: 83 ec 0c                   sub    esp,0xc
 8048471: 8d 45 f6                   lea    eax,[ebp-0xa]
 8048474: 50                         push   eax
 8048475: 68 20 85 04 08             push   0x8048520
 804847a: e8 d1 fe ff ff             call   8048350 <__isoc99_scanf@plt>
 804847f: 83 c4 08                   add    esp,0x8
 8048482: 8d 45 f6                   lea    eax,[ebp-0xa]
 8048485: 50                         push   eax
 8048486: 68 25 85 04 08             push   0x8048525
 804848b: e8 a0 fe ff ff             call   8048330 <printf@plt>
 8048490: 83 c4 08                   add    esp,0x8
 8048493: b8 00 00 00 00             mov    eax,0x0
 8048498: c9                         leave
 8048499: c3                         ret
```

Exercise26

```
080483db <func>:
 80483db:  55                    push   ebp
 80483dc:  89 e5                 mov    ebp,esp
 80483de:  8b 55 08              mov    edx,DWORD PTR [ebp+0x8]
 80483e1:  8b 45 0c              mov    eax,DWORD PTR [ebp+0xc]
 80483e4:  01 d0                 add    eax,edx
 80483e6:  5d                    pop    ebp
 80483e7:  c3                    ret

080483e8 <main>:
 80483e8:  55                    push   ebp
 80483e9:  89 e5                 mov    ebp,esp
 80483eb:  83 ec 04              sub    esp,0x4
 80483ee:  6a 0a                 push   0xa
 80483f0:  6a 05                 push   0x5
 80483f2:  e8 e4 ff ff ff        call   80483db <func>
 80483f7:  83 c4 08              add    esp,0x8
 80483fa:  89 45 fc              mov    DWORD PTR [ebp-0x4],eax
 80483fd:  b8 00 00 00 00        mov    eax,0x0
 8048402:  c9                    leave
 8048403:  c3                    ret
```

Exercise27

```
080483db <func>:
 80483db:   55                      push   ebp
 80483dc:   89 e5                   mov    ebp,esp
 80483de:   8b 55 08                mov    edx,DWORD PTR [ebp+0x8]
 80483e1:   8b 45 0c                mov    eax,DWORD PTR [ebp+0xc]
 80483e4:   01 d0                   add    eax,edx
 80483e6:   5d                      pop    ebp
 80483e7:   c3                      ret

080483e8 <main>:
 80483e8:   55                      push   ebp
 80483e9:   89 e5                   mov    ebp,esp
 80483eb:   83 ec 08                sub    esp,0x8
 80483ee:   c7 45 fc db 83 04 08    mov    DWORD PTR [ebp-0x4],0x80483db
 80483f5:   6a 0a                   push   0xa
 80483f7:   6a 05                   push   0x5
 80483f9:   8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483fc:   ff d0                   call   eax
 80483fe:   83 c4 08                add    esp,0x8
 8048401:   89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 8048404:   b8 00 00 00 00          mov    eax,0x0
 8048409:   c9                      leave
 804840a:   c3                      ret
```

Type Convention

Exercise28

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: 66 c7 45 fe 05 00       mov    WORD PTR [ebp-0x2],0x5
 80483e7: 0f bf 45 fe             movsx  eax,WORD PTR [ebp-0x2]
 80483eb: 89 45 f8                mov    DWORD PTR [ebp-0x8],eax
 80483ee: b8 00 00 00 00          mov    eax,0x0
 80483f3: c9                      leave
 80483f4: c3                      ret
```

Exercise29

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 08                sub    esp,0x8
 80483e1: c7 45 fc 0a 00 00 00    mov    DWORD PTR [ebp-0x4],0xa
 80483e8: 8b 45 fc                mov    eax,DWORD PTR [ebp-0x4]
 80483eb: 66 89 45 fa             mov    WORD PTR [ebp-0x6],ax
 80483ef: b8 00 00 00 00          mov    eax,0x0
 80483f4: c9                      leave
 80483f5: c3                      ret
```

Other Types

Exercise30

```
0x8048490:   6.8899999999999997

080483db <main>:
 80483db: 8d 4c 24 04              lea    ecx,[esp+0x4]
 80483df: 83 e4 f8                 and    esp,0xfffffff8
 80483e2: ff 71 fc                 push   DWORD PTR [ecx-0x4]
 80483e5: 55                       push   ebp
 80483e6: 89 e5                    mov    ebp,esp
 80483e8: 51                       push   ecx
 80483e9: 83 ec 0c                 sub    esp,0xc
 80483ec: dd 05 90 84 04 08        fld    QWORD PTR ds:0x8048490
 80483f2: dd 5d f0                 fstp   QWORD PTR [ebp-0x10]
 80483f5: b8 00 00 00 00           mov    eax,0x0
 80483fa: 83 c4 0c                 add    esp,0xc
 80483fd: 59                       pop    ecx
 80483fe: 5d                       pop    ebp
 80483ff: 8d 61 fc                 lea    esp,[ecx-0x4]
 8048402: c3                       ret
```

Exercise31

HINT: array

```
080483db <main>:
 80483db: 55                        push   ebp
 80483dc: 89 e5                     mov    ebp,esp
 80483de: 83 ec 2c                  sub    esp,0x2c
 80483e1: c7 45 fc 00 00 00 00      mov    DWORD PTR [ebp-0x4],0x0
 80483e8: c7 45 fc 00 00 00 00      mov    DWORD PTR [ebp-0x4],0x0
 80483ef: eb 0e                     jmp    80483ff <main+0x24>
 80483f1: 8b 45 fc                  mov    eax,DWORD PTR [ebp-0x4]
 80483f4: 8b 55 fc                  mov    edx,DWORD PTR [ebp-0x4]
 80483f7: 89 54 85 d4               mov    DWORD PTR [ebp+eax*4-0x2c],e
 80483fb: 83 45 fc 01               add    DWORD PTR [ebp-0x4],0x1
 80483ff: 83 7d fc 09               cmp    DWORD PTR [ebp-0x4],0x9
 8048403: 7e ec                     jle    80483f1 <main+0x16>
 8048405: b8 00 00 00 00            mov    eax,0x0
 804840a: c9                        leave
 804840b: c3                        ret
```

Exercise32

HINT: 2D array

```
080483db <main>:
 80483db:55                     push   ebp
 80483dc:89 e5                  mov    ebp,esp
 80483de:81 ec 98 01 00 00      sub    esp,0x198
 80483e4:c7 45 fc 00 00 00 00   mov    DWORD PTR [ebp-0x4],0x0
 80483eb:eb 38                  jmp    8048425 <main+0x4a>
 80483ed:c7 45 f8 00 00 00 00   mov    DWORD PTR [ebp-0x8],0x0
 80483f4:eb 25                  jmp    804841b <main+0x40>
 80483f6:8b 55 fc               mov    edx,DWORD PTR [ebp-0x4]
 80483f9:8b 45 f8               mov    eax,DWORD PTR [ebp-0x8]
 80483fc:8d 0c 02               lea    ecx,[edx+eax*1]
 80483ff:8b 55 fc               mov    edx,DWORD PTR [ebp-0x4]
 8048402:89 d0                  mov    eax,edx
 8048404:c1 e0 02               shl    eax,0x2
 8048407:01 d0                  add    eax,edx
 8048409:01 c0                  add    eax,eax
 804840b:8b 55 f8               mov    edx,DWORD PTR [ebp-0x8]
 804840e:01 d0                  add    eax,edx
 8048410:89 8c 85 68 fe ff ff   mov    DWORD PTR [ebp+eax*4-0x198],ecx
 8048417:83 45 f8 01            add    DWORD PTR [ebp-0x8],0x1
 804841b:83 7d f8 09            cmp    DWORD PTR [ebp-0x8],0x9
 804841f:7e d5                  jle    80483f6 <main+0x1b>
 8048421:83 45 fc 01            add    DWORD PTR [ebp-0x4],0x1
 8048425:83 7d fc 09            cmp    DWORD PTR [ebp-0x4],0x9
 8048429:7e c2                  jle    80483ed <main+0x12>
 804842b:b8 00 00 00 00         mov    eax,0x0
 8048430:c9                     leave
 8048431:c3                     ret
```

Exercise33

HINT: struct

```
0804840b <main>:
 804840b: 55                       push   ebp
 804840c: 89 e5                    mov    ebp,esp
 804840e: 83 ec 08                 sub    esp,0x8
 8048411: c7 45 f8 01 00 00 00     mov    DWORD PTR [ebp-0x8],0x1
 8048418: c7 45 fc e9 00 00 00     mov    DWORD PTR [ebp-0x4],0xe9
 804841f: 8b 55 fc                 mov    edx,DWORD PTR [ebp-0x4]
 8048422: 8b 45 f8                 mov    eax,DWORD PTR [ebp-0x8]
 8048425: 52                       push   edx
 8048426: 50                       push   eax
 8048427: 68 c0 84 04 08           push   0x80484c0
 804842c: e8 af fe ff ff           call   80482e0 <printf@plt>
 8048431: 83 c4 0c                 add    esp,0xc
 8048434: b8 00 00 00 00           mov    eax,0x0
 8048439: c9                       leave
 804843a: c3                       ret
```

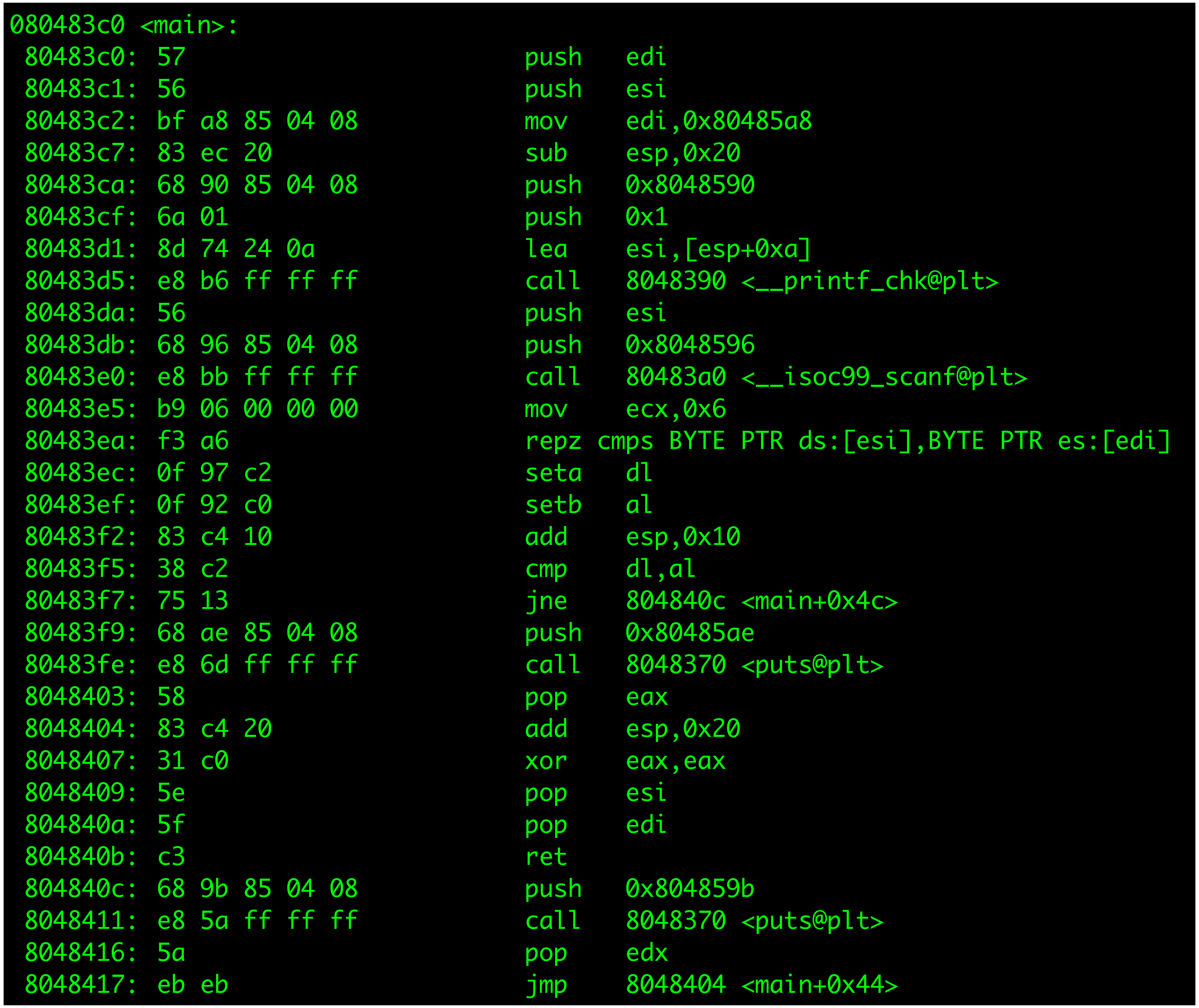
Exercise34

HINT: union

```
080483db <main>:
 80483db: 55                      push   ebp
 80483dc: 89 e5                   mov    ebp,esp
 80483de: 83 ec 04                sub    esp,0x4
 80483e1: c7 45 fc 01 00 00 00    mov    DWORD PTR [ebp-0x4],0x1
 80483e8: c7 45 fc 02 00 00 00    mov    DWORD PTR [ebp-0x4],0x2
 80483ef: b8 00 00 00 00          mov    eax,0x0
 80483f4: c9                      leave
 80483f5: c3                      ret
```

String Operation

Exercise35



```
080483c0 <main>:
 80483c0: 57                   push   edi
 80483c1: 56                   push   esi
 80483c2: bf a8 85 04 08       mov    edi,0x80485a8
 80483c7: 83 ec 20             sub    esp,0x20
 80483ca: 68 90 85 04 08       push   0x8048590
 80483cf: 6a 01                push   0x1
 80483d1: 8d 74 24 0a          lea    esi,[esp+0xa]
 80483d5: e8 b6 ff ff ff       call   8048390 <__printf_chk@plt>
 80483da: 56                   push   esi
 80483db: 68 96 85 04 08       push   0x8048596
 80483e0: e8 bb ff ff ff       call   80483a0 <__isoc99_scanf@plt>
 80483e5: b9 06 00 00 00       mov    ecx,0x6
 80483ea: f3 a6                repz cmps BYTE PTR ds:[esi],BYTE PTR es:[edi]
 80483ec: 0f 97 c2             seta   dl
 80483ef: 0f 92 c0             setb   al
 80483f2: 83 c4 10             add    esp,0x10
 80483f5: 38 c2                cmp    dl,al
 80483f7: 75 13                jne    804840c <main+0x4c>
 80483f9: 68 ae 85 04 08       push   0x80485ae
 80483fe: e8 6d ff ff ff       call   8048370 <puts@plt>
 8048403: 58                   pop    eax
 8048404: 83 c4 20             add    esp,0x20
 8048407: 31 c0                xor    eax,eax
 8048409: 5e                   pop    esi
 804840a: 5f                   pop    edi
 804840b: c3                   ret
 804840c: 68 9b 85 04 08       push   0x804859b
 8048411: e8 5a ff ff ff       call   8048370 <puts@plt>
 8048416: 5a                   pop    edx
 8048417: eb eb                jmp    8048404 <main+0x44>
```

Reference

- AIS3 2016 Reversing Lecture (AsukaNakajima)
- http://www.cs.virginia.edu/~evans/cs216/guides/x86.html