新竹人最愛逆向(工程)

terrynini38514

terrynini


WHO AM I ?

ID : Terrynini38514

- 跟我有點熟：
  逆逆
- 跟我麻吉麻：
  國立交通大學SENSE LAB
  資電玄客與安全碩士學位學程陳廷宇
- 稍微能拿來吹的東西：
  2019 年金盾獎冠軍
  2019, 2020 FireEye Flare On Challenge 破台
- CTF Team：
  DoubleSigma（已慘遭 Balsn 併吞化作其血肉）
  Balsn

沒照片可用 救命

Switch (less than 3 case)

```c
#include <stdio.h>

int main(){
    char a;
    scanf("%c", &a);
    switch(a){
        case 'A':
            printf("A");
            break;
        case 'B':
            printf("B? What the?");
            break;
        case 'C':
            printf("C? you took the wrong way...");
            break;
    }
    return 0;
}
```

```
...
 794:   e8 a7 fe ff ff          call   640 <__isoc99_scanf@plt>
 799:   0f b6 45 f7             movzx  eax,BYTE PTR [rbp-0x9]
 79d:   0f be c0                movsx  eax,al
 7a0:   83 f8 42                cmp    eax,0x42
 7a3:   74 16                   je     7bb <main+0x51>
 7a5:   83 f8 43                cmp    eax,0x43
 7a8:   74 24                   je     7ce <main+0x64>
 7aa:   83 f8 41                cmp    eax,0x41
 7ad:   75 31                   jne    7e0 <main+0x76>
 7af:   bf 41 00 00 00          mov    edi,0x41
 7b4:   e8 57 fe ff ff          call   610 <putchar@plt>
 7b9:   eb 25                   jmp    7e0 <main+0x76>
 7bb:   48 8d 3d c5 00 00 00    lea    rdi,[rip+0xc5]
 7c2:   b8 00 00 00 00          mov    eax,0x0
 7c7:   e8 64 fe ff ff          call   630 <printf@plt>
 7cc:   eb 12                   jmp    7e0 <main+0x76>
 7ce:   48 8d 3d bf 00 00 00    lea    rdi,[rip+0xbf]
 7d5:   b8 00 00 00 00          mov    eax,0x0
 7da:   e8 51 fe ff ff          call   630 <printf@plt>
...
```

Switch (more than 3 case, ordered, no-pie)

```c
#include <stdio.h>

int main(){
    char a;
    scanf("%c", &a);
    switch(a){
        case 'A':
            printf("A");
            break;
        case 'B':
            printf("B? What the?");
            break;
        case 'C':
            printf("C? you took the wrong way...");
            break;
        case 'D':
            printf("D? you took the wrong way...");
            break;
        case 'E':
            printf("E? you took the wrong way...");
            break;

    return 0;
}
```

```
794:   e8 a7 fe ff ff          call   640 <__isoc99_scanf@plt>
799:   0f b6 45 f7             movzx  eax,BYTE PTR [rbp-0x9]
79d:   0f be c0                movsx  eax,al
7a0:   83 e8 41                sub    eax,0x41
7a3:   83 f8 04                cmp    eax,0x4
7a6:   77 7a                   ja     822 <main+0xb8>
7a8:   89 c0                   mov    eax,eax
7aa:   48 8d 14 85 00 00 00    lea    rdx,[rax*4+0x0]
7b1:   00
7b2:   48 8d 05 73 01 00 00    lea    rax,[rip+0x173]
7b9:   8b 04 02                mov    eax,DWORD PTR [rdx+rax*1]
7bc:   48 63 d0                movsxd rdx,eax
7bf:   48 8d 05 66 01 00 00    lea    rax,[rip+0x166]
7c6:   48 01 d0                add    rax,rdx
7c9:   ff e0                   jmp    rax
7cb:   bf 41 00 00 00          mov    edi,0x41
7d0:   e8 3b fe ff ff          call   610 <putchar@plt>
...
```

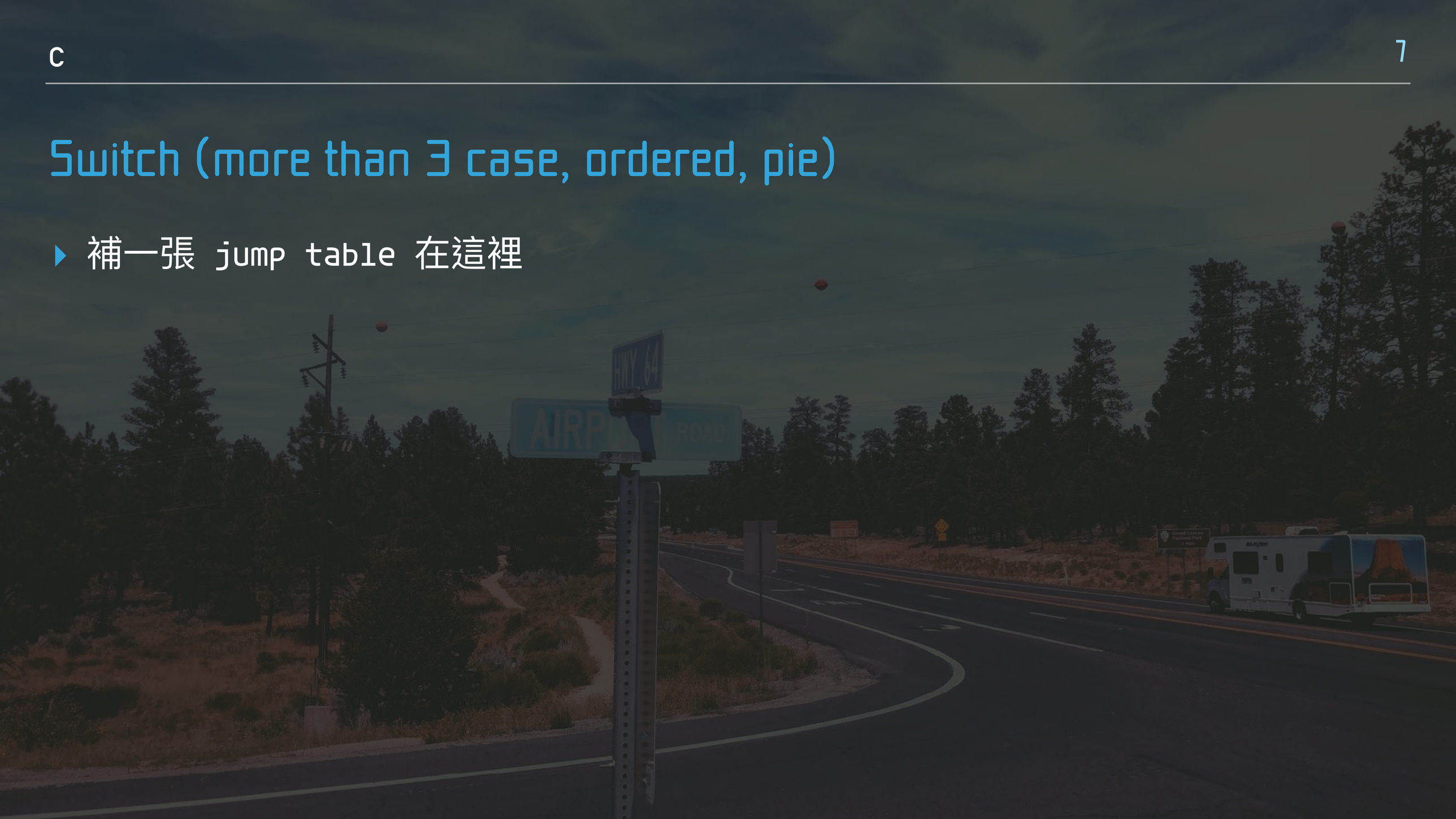
Switch (more than 3 case, ordered, pie)

```c
#include <stdio.h>

int main(){
    char a;
    scanf("%c", &a);
    switch(a){
        case 'A':
            printf("A");
            break;
        case 'B':
            printf("B? What the?");
            break;
        case 'C':
            printf("C? you took the wrong way...");
            break;
        case 'D':
            printf("D? you took the wrong way...");
            break;
        case 'E':
            printf("E? you took the wrong way...");
            break;

    return 0;
}
```

```
794:   e8 a7 fe ff ff          call   640 <__isoc99_scanf@plt>
799:   0f b6 45 f7             movzx  eax,BYTE PTR [rbp-0x9]
79d:   0f be c0                movsx  eax,al
7a0:   83 e8 41                sub    eax,0x41
7a3:   83 f8 04                cmp    eax,0x4
7a6:   77 7a                   ja     822 <main+0xb8>
7a8:   89 c0                   mov    eax,eax
7aa:   48 8d 14 85 00 00 00    lea    rdx,[rax*4+0x0]
7b1:   00
7b2:   48 8d 05 73 01 00 00    lea    rax,[rip+0x173]
7b9:   8b 04 02                mov    eax,DWORD PTR [rdx+rax*1]
7bc:   48 63 d0                movsxd rdx,eax
7bf:   48 8d 05 66 01 00 00    lea    rax,[rip+0x166]
7c6:   48 01 d0                add    rax,rdx
7c9:   ff e0                   jmp    rax
7cb:   bf 41 00 00 00          mov    edi,0x41
7d0:   e8 3b fe ff ff          call   610 <putchar@plt>
...
```

Switch (more than 3 case, ordered, pie)

- 補一張 jump table 在這裡



DEMO?

Switch (more than 3 case, random)

DEMO?

do-while loop

```c
#include <stdio.h>

int main(){
    int input;
    scanf("%d",&input);
    do{
        input += 1;
    }while(input > 0);
    printf("%d",input);
    return 0;
}
```

```
...
 744:   e8 a7 fe ff ff          call   5f0 <__isoc99_scanf@plt>
 749:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 74c:   83 c0 01                add    eax,0x1
 74f:   89 45 f4                mov    DWORD PTR [rbp-0xc],eax
 752:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 755:   85 c0                   test   eax,eax
 757:   7f f0                   jg     749 <main+0x2f>
 759:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 75c:   89 c6                   mov    esi,eax
 75e:   48 8d 3d af 00 00 00    lea    rdi,[rip+0xaf]
 765:   b8 00 00 00 00          mov    eax,0x0
 76a:   e8 71 fe ff ff          call   5e0 <printf@plt>
...
```

while loop

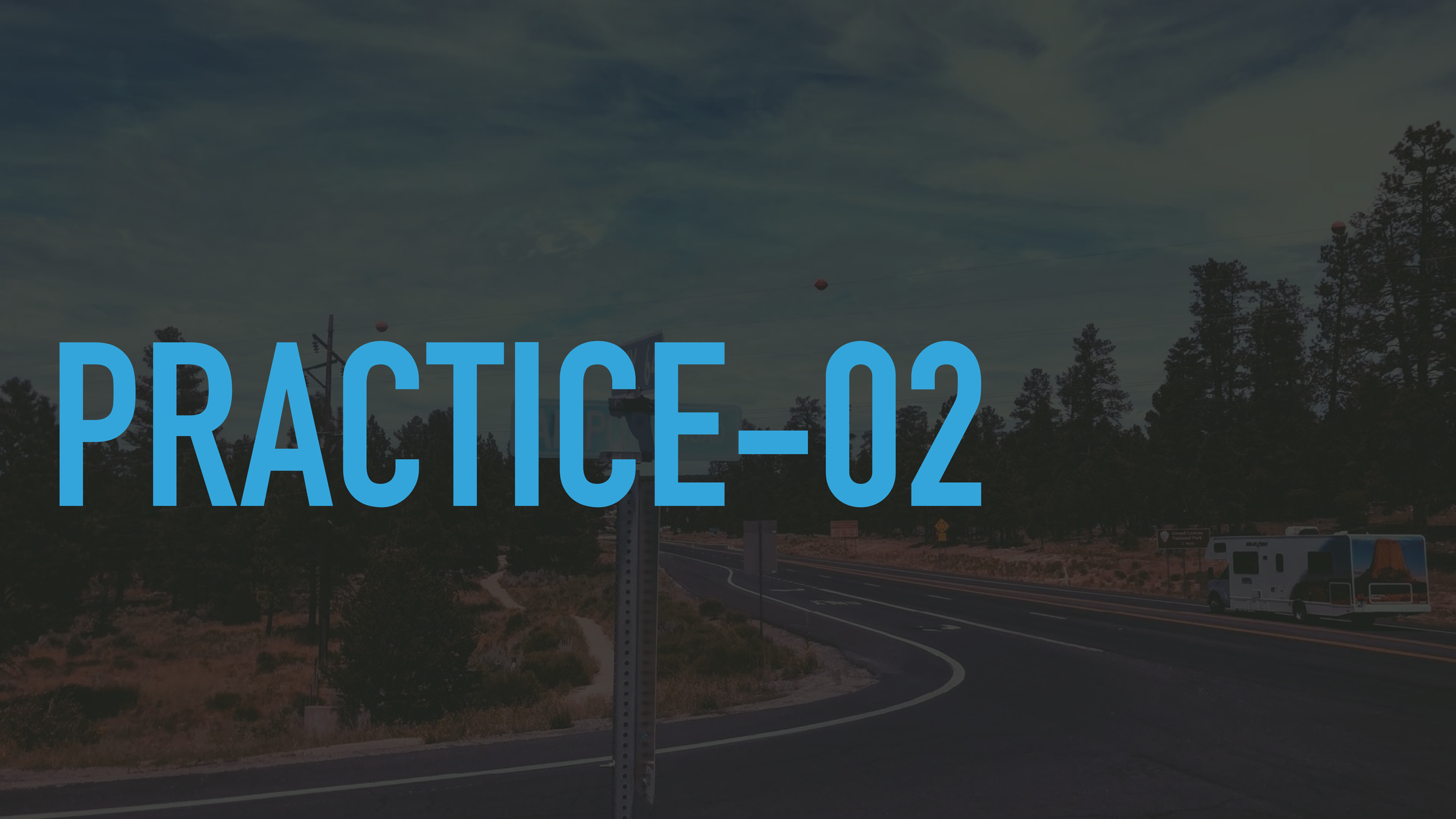
```c
#include <stdio.h>

int main(){
    int input;
    scanf("%d",&input);
    while(input > 0){
        input += 1;
    }
    printf("%d",input);
    return 0;
}
```

```
...
 744:   e8 a7 fe ff ff          call   5f0 <__isoc99_scanf@plt>
 749:   eb 09                   jmp    754 <main+0x3a>
 74b:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 74e:   83 c0 01                add    eax,0x1
 751:   89 45 f4                mov    DWORD PTR [rbp-0xc],eax
 754:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 757:   85 c0                   test   eax,eax
 759:   7f f0                   jg     74b <main+0x31>
 75b:   8b 45 f4                mov    eax,DWORD PTR [rbp-0xc]
 75e:   89 c6                   mov    esi,eax
 760:   48 8d 3d ad 00 00 00    lea    rdi,[rip+0xad]
 767:   b8 00 00 00 00          mov    eax,0x0
 76c:   e8 6f fe ff ff          call   5e0 <printf@plt>
...
```

for loop

```c
#include <stdio.h>

int main(){
    int input;
    scanf("%d",&input);
    for(int i = 0 ; i < 100; i += 1){
        printf("%d",input);
    }
    return 0;
}
```

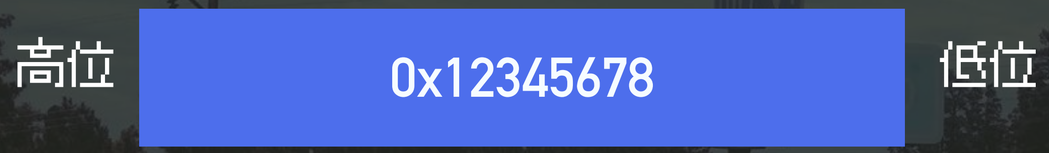
```
...
744:   e8 a7 fe ff ff          call   5f0 <__isoc99_scanf@plt>
749:   c7 45 f4 00 00 00 00    mov    DWORD PTR [rbp-0xc],0x0
750:   eb 1a                   jmp    76c <main+0x52>
752:   8b 45 f0                mov    eax,DWORD PTR [rbp-0x10]
755:   89 c6                   mov    esi,eax
757:   48 8d 3d b6 00 00 00    lea    rdi,[rip+0xb6]
75e:   b8 00 00 00 00          mov    eax,0x0
763:   e8 78 fe ff ff          call   5e0 <printf@plt>
768:   83 45 f4 01             add    DWORD PTR [rbp-0xc],0x1
76c:   83 7d f4 63             cmp    DWORD PTR [rbp-0xc],0x63
770:   7e e0                   jle    752 <main+0x38>
...
```

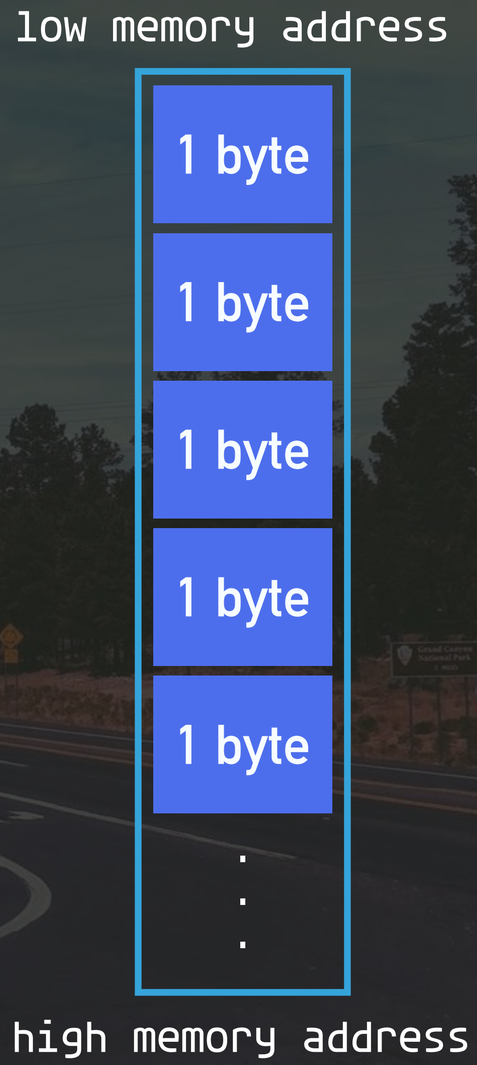


PRACTICE-02

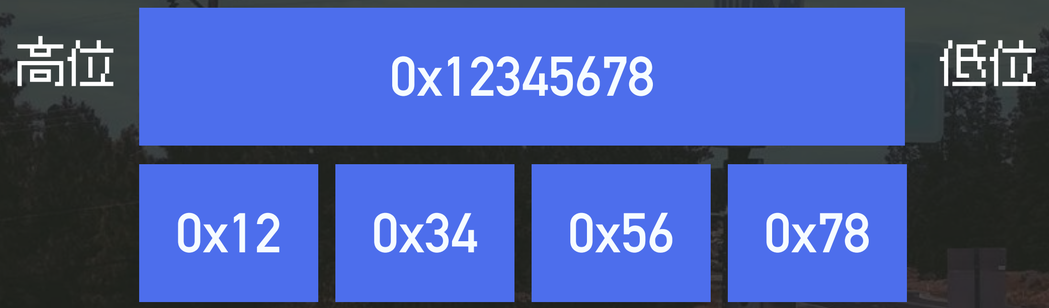

MEMORY

Endian


高位
0x12345678
低位

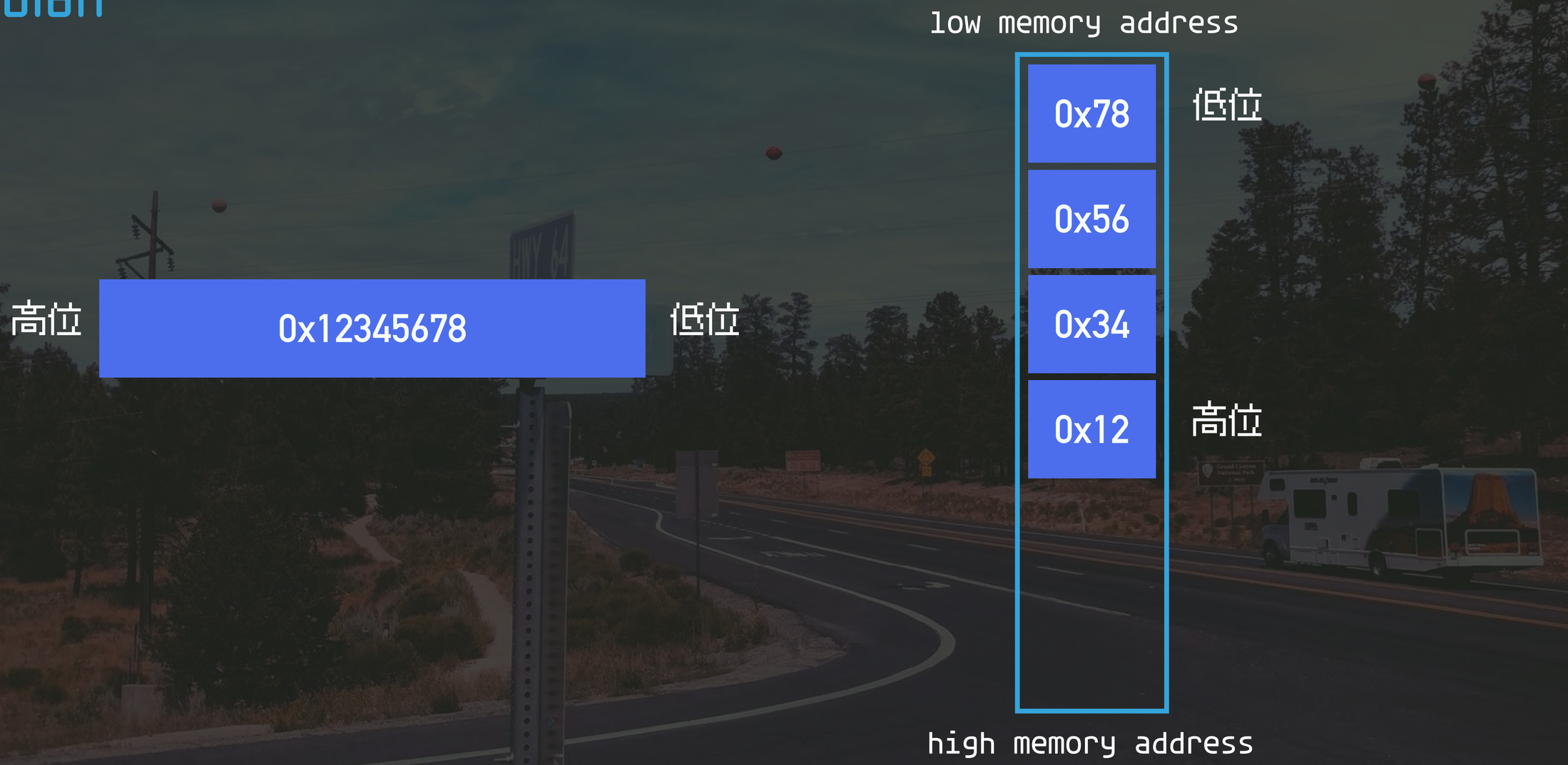

low memory address
1 byte
1 byte
1 byte
1 byte
1 byte
high memory address

Endian


高位
0x12345678
低位
0x12
0x34
0x56
0x78


low memory address
high memory address

Little-Endian

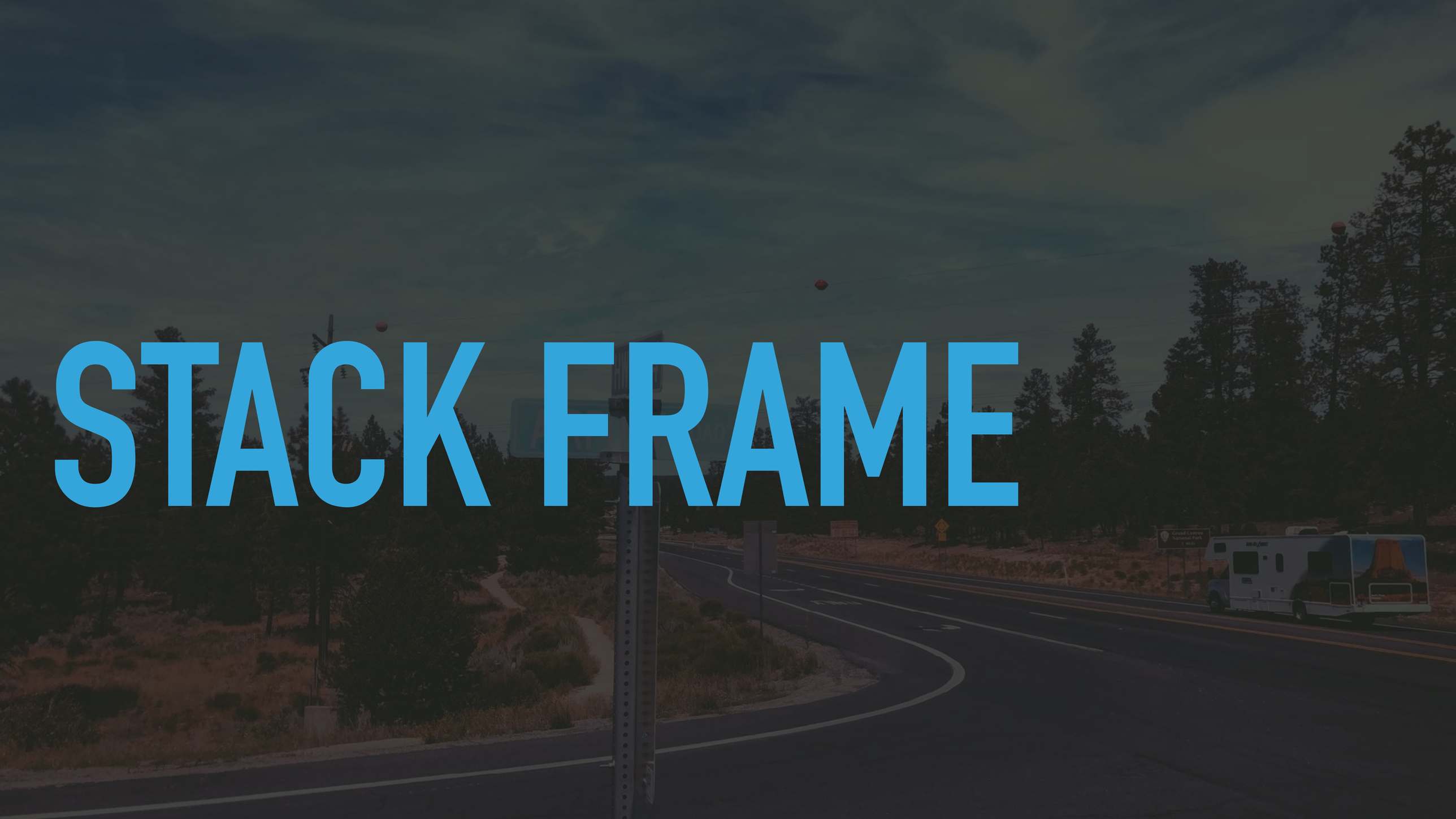


Little-Endian

byte ptr [rax]	word ptr [rax]	dword ptr [rax]
0x78	0x5678	0x12345678

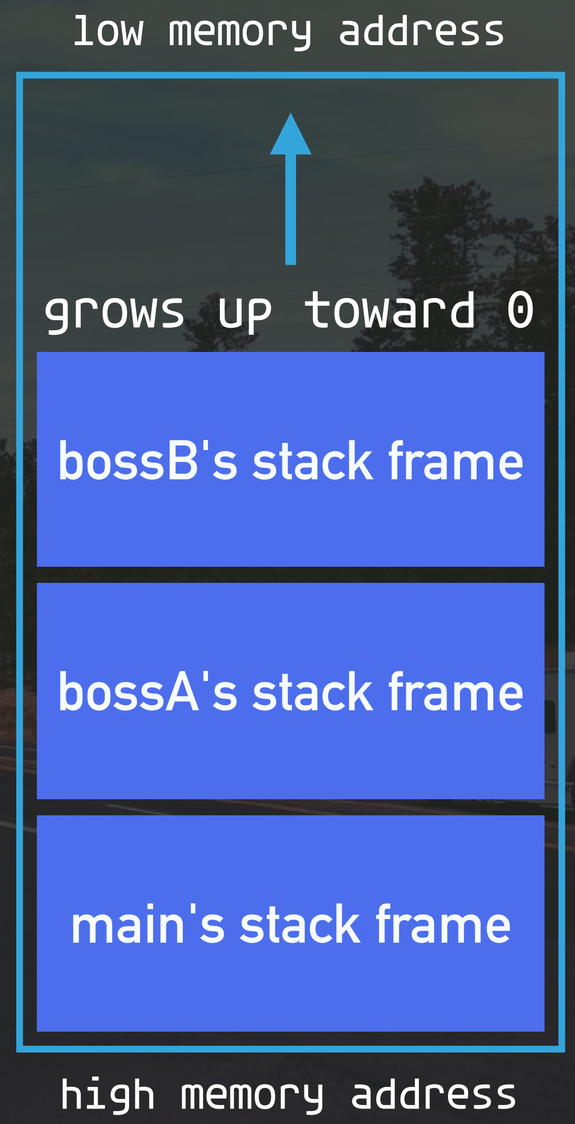
qword ptr [rax]

oword ptr [rax]


STACK FRAME

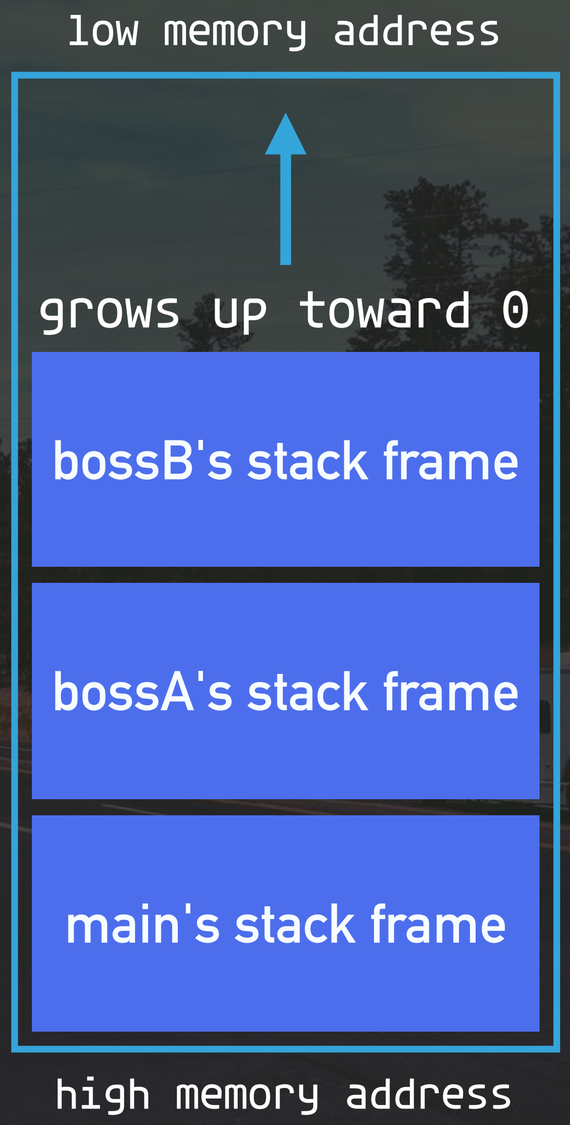
Stack Frame (Abstraction)

```c
#include <stdio.h>

void bossB(int time){
    printf("do this in %d hours", time/2);
}
void bossA(int time){
    int bossA_want = time/2;
    printf("do this in %d hours", bossA_want);
    bossB(bossA_want);
}

int main(){
    bossA(100);
    return 0;
}
```



Stack Frame (Abstraction)

```
 1  8048530:    50                   push   eax
 2  8048531:    e8 96 ff ff ff       call   80484cc <bossA>
 3  8048536:    83 c4 10             add    esp,0x10
 4  8048539:    83 ec 0c             sub    esp,0xc
 5  804853c:    68 f7 85 04 08       push   0x80485f7
 6  8048541:    e8 0a fe ff ff       call   8048350 <puts@plt>
 7  8048546:    83 c4 10             add    esp,0x10
 8  8048549:    b8 00 00 00 00       mov    eax,0x0
 9  804854e:    8b 4d fc             mov    ecx,DWORD PTR [ebp-0x4]
10  8048551:    c9                   lea
```



Stack Frame

```
1  8048530:    50                   push   eax
2  8048531:    e8 96 ff ff ff       call   80484cc <bossA>
3  8048536:    83 c4 10             add    esp,0x10
4  8048539:    83 ec 0c             sub    esp,0xc
5  804853c:    68 f7 85 04 08       push   0x80485f7
6  8048541:    e8 0a fe ff ff       call   8048350 <puts@plt>
7  8048546:    83 c4 10             add    esp,0x10
8  8048549:    b8 00 00 00 00       mov    eax,0x0
9  804854e:    8b 4d fc             mov    ecx,DWORD PTR [ebp-0x4]
10 8048551:    c9                   lea
```

highlight means RIP points to here

(假設我們的輸入是 0x40)

Stack Frame

```
1  8048530:      50                   push   eax
2  8048531:      e8 96 ff ff ff       call   80484cc <bossA>
3  8048536:      83 c4 10             add    esp,0x10
4  8048539:      83 ec 0c             sub    esp,0xc
5  804853c:      68 f7 85 04 08       push   0x80485f7
6  8048541:      e8 0a fe ff ff       call   8048350 <puts@plt>
7  8048546:      83 c4 10             add    esp,0x10
8  8048549:      b8 00 00 00 00       mov    eax,0x0
9  804854e:      8b 4d fc             mov    ecx,DWORD PTR [ebp-0x4]
10 8048551:      c9                   lea
```

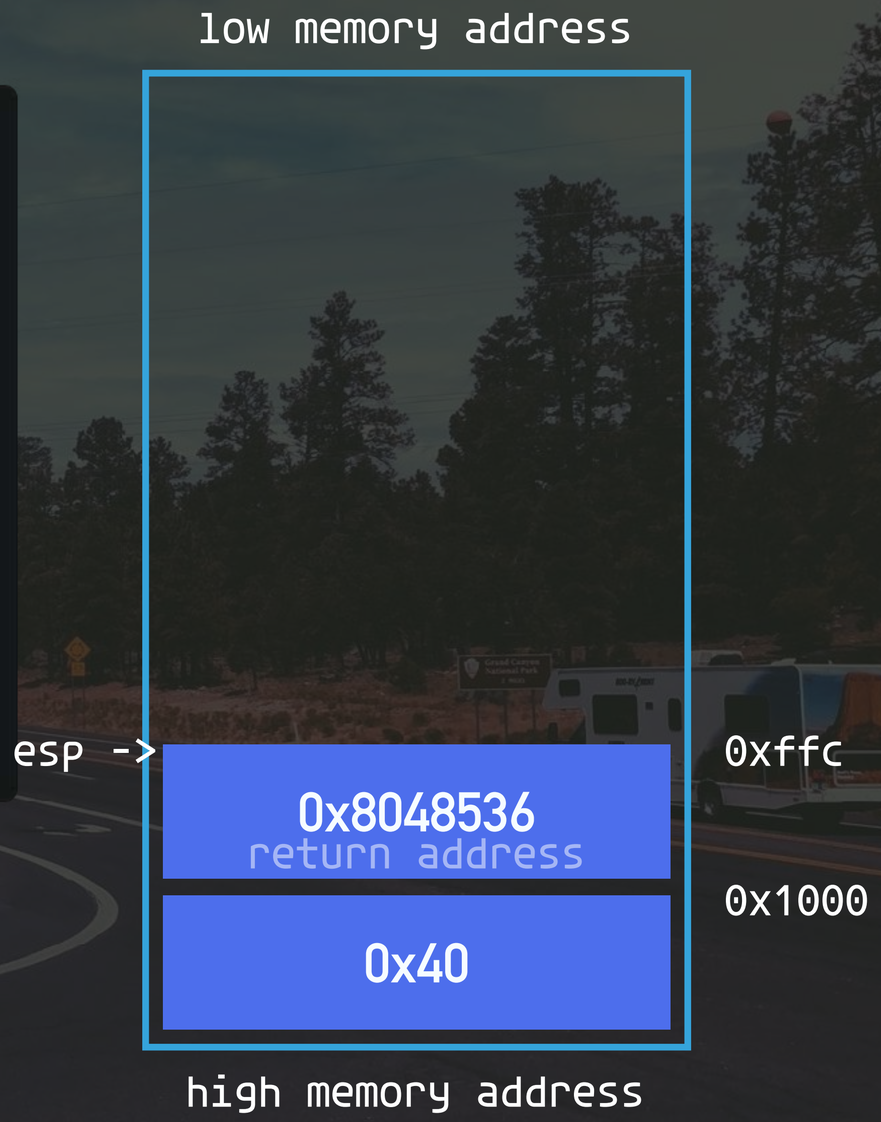
highlight means RIP points to here

(假設我們的輸入是 0x40)

Stack Frame

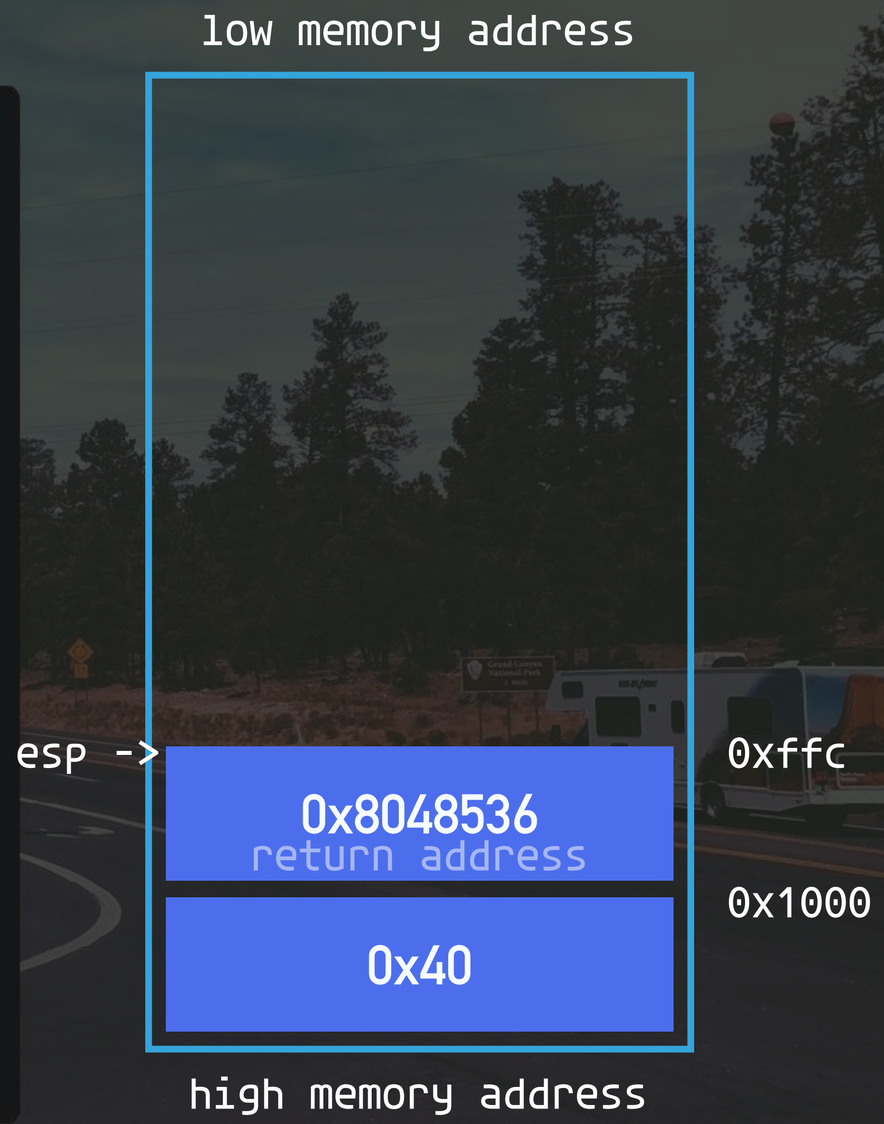
```
1  8048530:    50                   push   eax
2  8048531:    e8 96 ff ff ff       call   80484cc <bossA>
3  8048536:    83 c4 10             add    esp,0x10
4  8048539:    83 ec 0c             sub    esp,0xc
5  804853c:    68 f7 85 04 08       push   0x80485f7
6  8048541:    e8 0a fe ff ff       call   8048350 <puts@plt>
7  8048546:    83 c4 10             add    esp,0x10
8  8048549:    b8 00 00 00 00       mov    eax,0x0
9  804854e:    8b 4d fc             mov    ecx,DWORD PTR [ebp-0x4]
10 8048551:    c9                   lea
```

highlight means RIP points to here



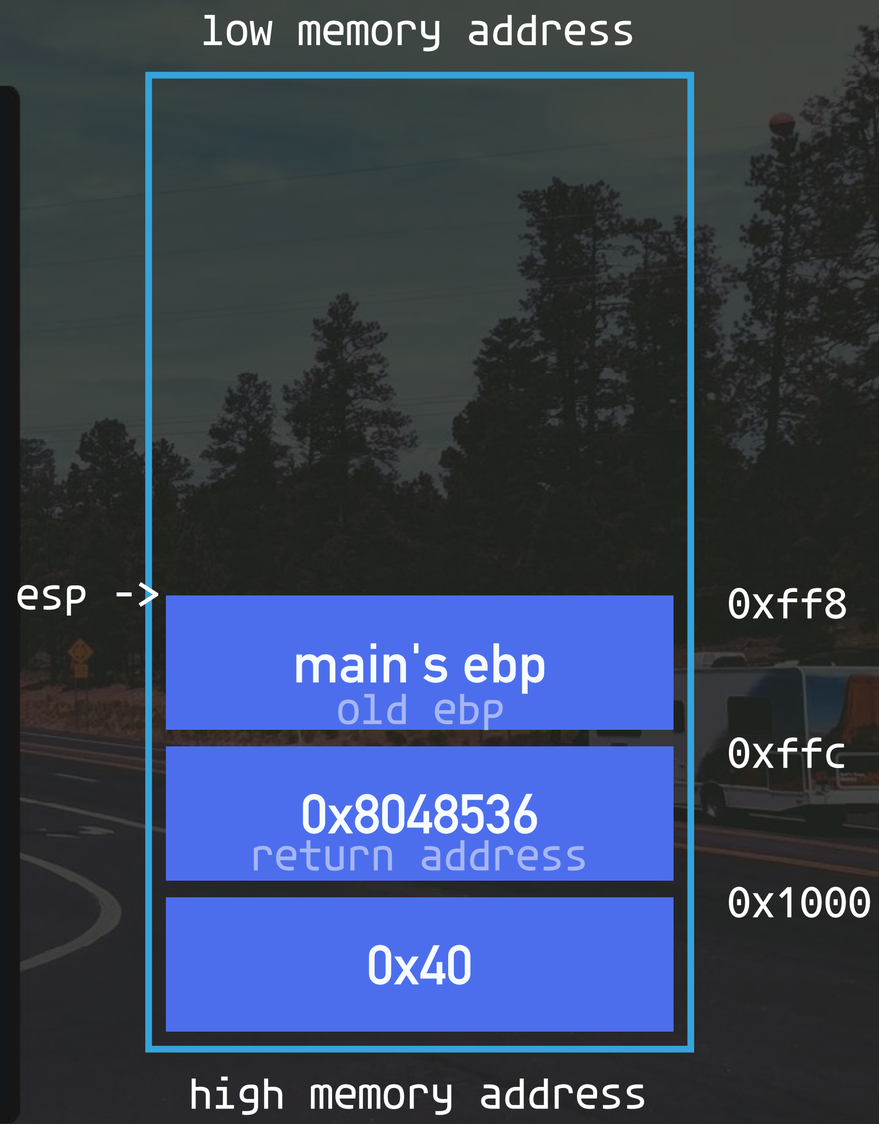
Stack Frame

```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
6  ...
7  80484de:       89 45 f4                mov    DWORD PTR [ebp-0xc],eax
8  ...
9  80484ec:       e8 4f fe ff ff          call   8048340 <printf@plt>
10 ...
11 80484f4:       83 ec 0c                sub    esp,0xc
12 80484f7:       ff 75 f4                push   DWORD PTR [ebp-0xc]
13 80484fa:       e8 a7 ff ff ff          call   80484a6 <bossB>
14 80484ff:       83 c4 10                add    esp,0x10
15 8048502:       90                      nop
16 8048503:       c9                      leave
17 8048504:       c3                      ret
```



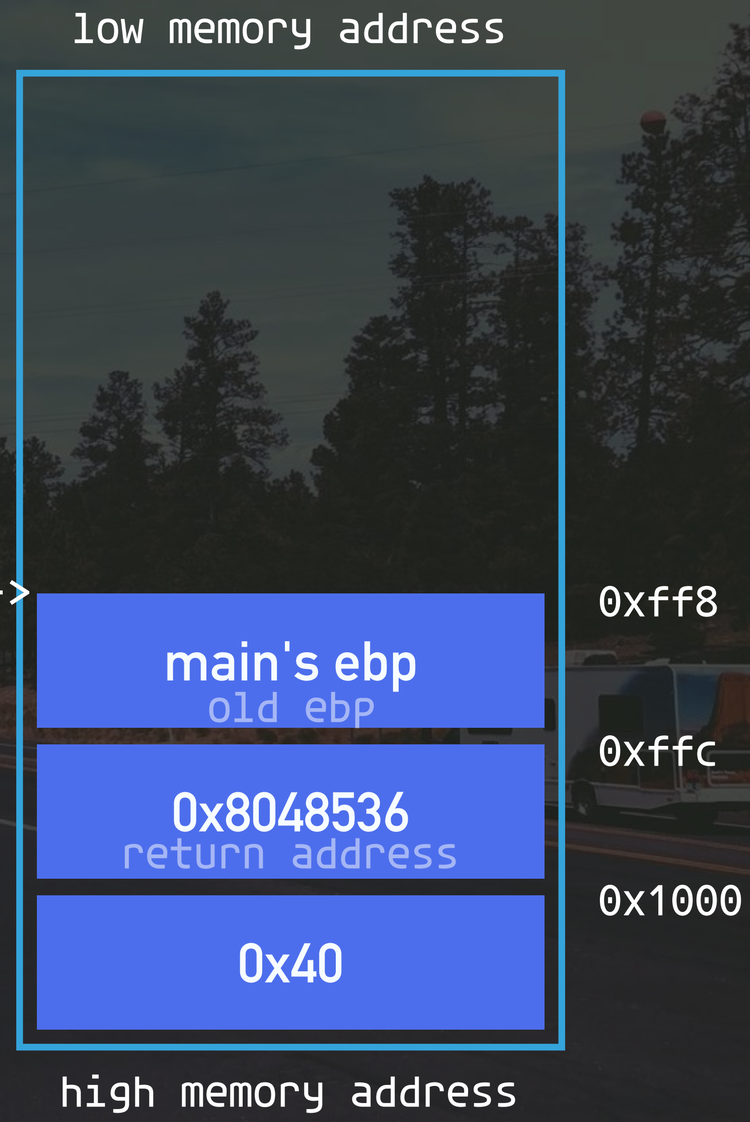
Stack Frame

```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
6  ...
7  80484de:       89 45 f4                mov    DWORD PTR [ebp-0xc],eax
8  ...
9  80484ec:       e8 4f fe ff ff          call   8048340 <printf@plt>
10 ...
11 80484f4:       83 ec 0c                sub    esp,0xc
12 80484f7:       ff 75 f4                push   DWORD PTR [ebp-0xc]
13 80484fa:       e8 a7 ff ff ff          call   80484a6 <bossB>
14 80484ff:       83 c4 10                add    esp,0x10
15 8048502:       90                      nop
16 8048503:       c9                      leave
17 8048504:       c3                      ret
```



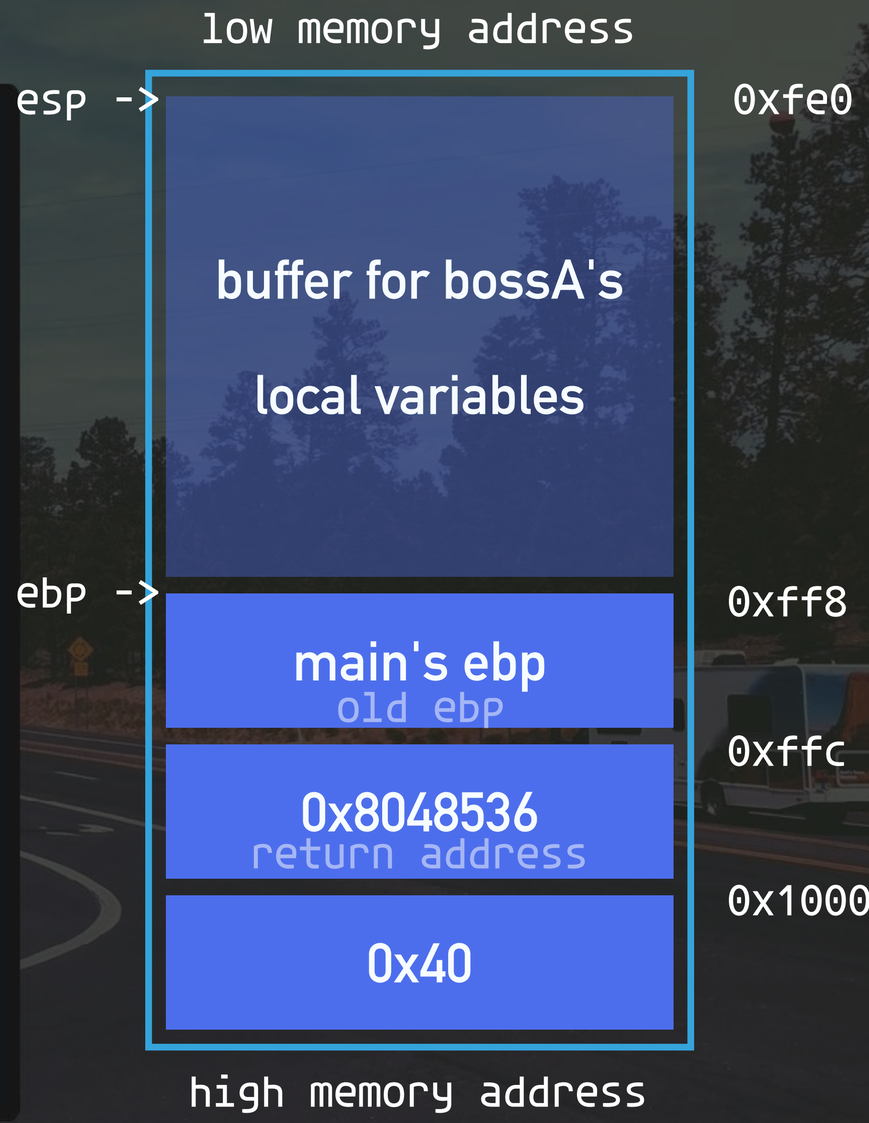
Stack Frame

```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
6  ...
7  80484de:       89 45 f4                mov    DWORD PTR [ebp-0xc],eax
8  ...
9  80484ec:       e8 4f fe ff ff          call   8048340 <printf@plt>
10 ...
11 80484f4:       83 ec 0c                sub    esp,0xc
12 80484f7:       ff 75 f4                push   DWORD PTR [ebp-0xc]
13 80484fa:       e8 a7 ff ff ff          call   80484a6 <bossB>
14 80484ff:       83 c4 10                add    esp,0x10
15 8048502:       90                      nop
16 8048503:       c9                      leave
17 8048504:       c3                      ret
```



Stack Frame

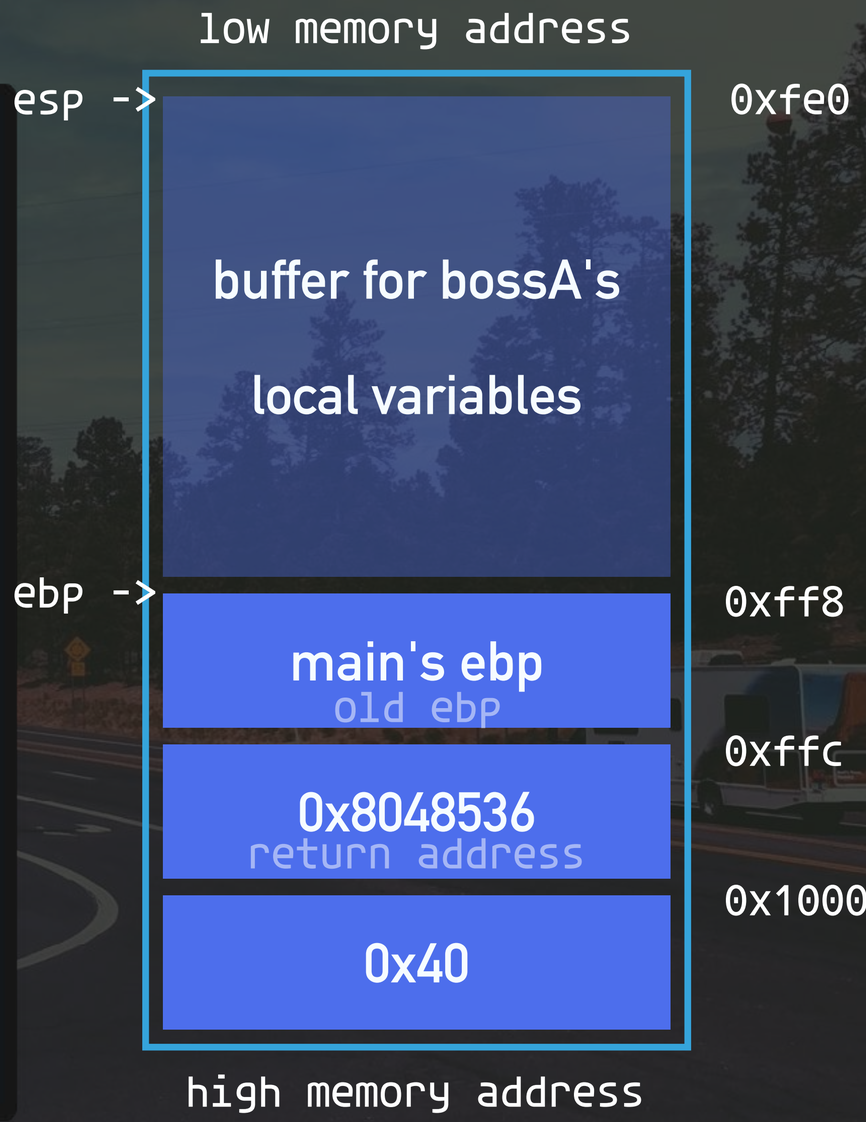
```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
6  ...
7  80484de:       89 45 f4                mov    DWORD PTR [ebp-0xc],eax
8  ...
9  80484ec:       e8 4f fe ff ff          call   8048340 <printf@plt>
10 ...
11 80484f4:       83 ec 0c                sub    esp,0xc
12 80484f7:       ff 75 f4                push   DWORD PTR [ebp-0xc]
13 80484fa:       e8 a7 ff ff ff          call   80484a6 <bossB>
14 80484ff:       83 c4 10                add    esp,0x10
15 8048502:       90                      nop
16 8048503:       c9                      leave
17 8048504:       c3                      ret
```



Stack Frame

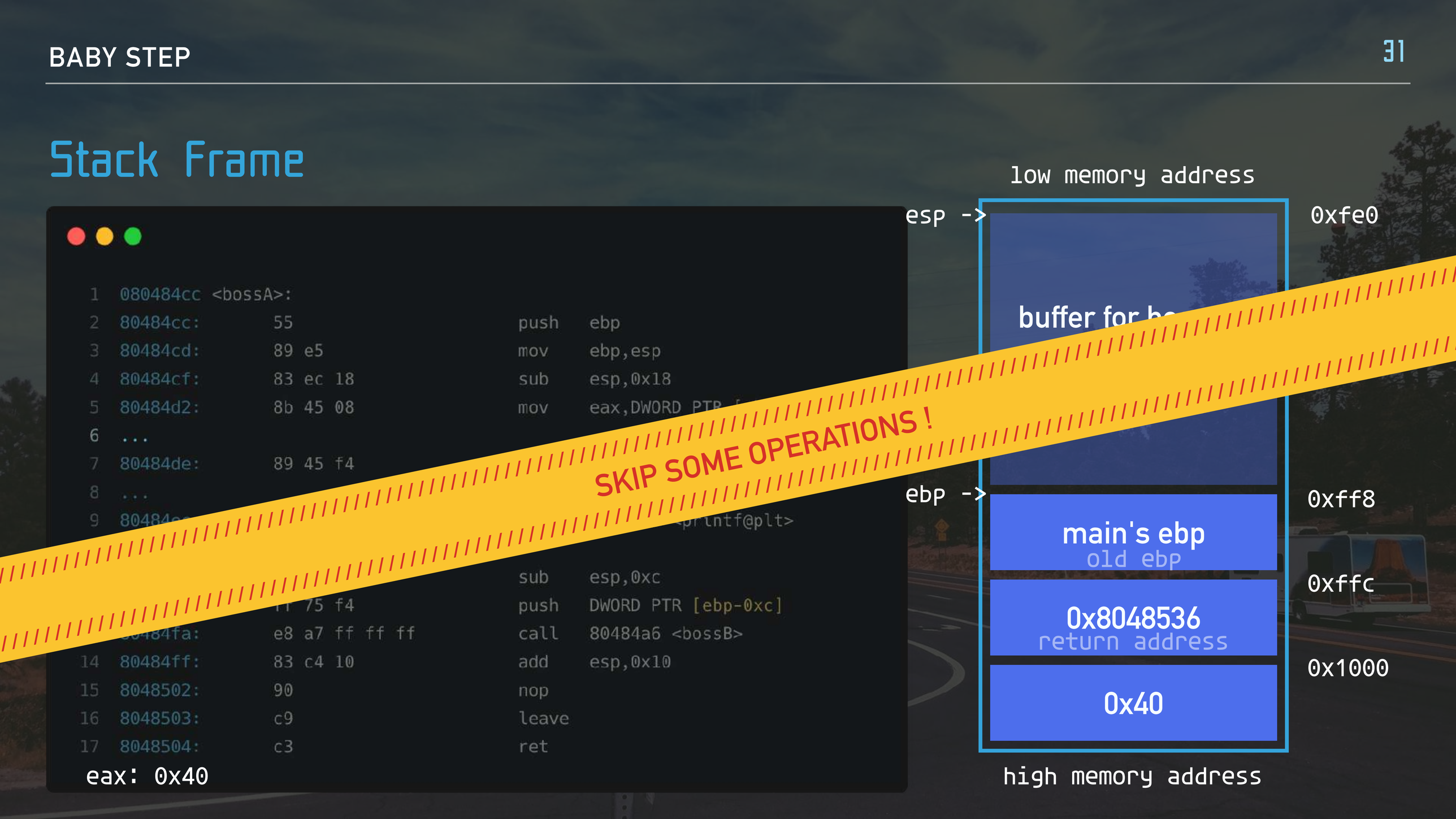
```
080484cc <bossA>:
80484cc:        55                      push   ebp
80484cd:        89 e5                   mov    ebp,esp
80484cf:        83 ec 18                sub    esp,0x18
80484d2:        8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
...
80484de:        89 45 f4                mov    DWORD PTR [ebp-0xc],eax
...
80484ec:        e8 4f fe ff ff          call   8048340 <printf@plt>
...
80484f4:        83 ec 0c                sub    esp,0xc
80484f7:        ff 75 f4                push   DWORD PTR [ebp-0xc]
80484fa:        e8 a7 ff ff ff          call   80484a6 <bossB>
80484ff:        83 c4 10                add    esp,0x10
8048502:        90                      nop
8048503:        c9                      leave
8048504:        c3                      ret
```

eax: 0x40



Stack Frame

```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [
6  ...
7  80484de:       89 45 f4
8  ...
9  80484e
                                          <printf@plt>
                                  sub    esp,0xc
                  ff 75 f4        push   DWORD PTR [ebp-0xc]
                  e8 a7 ff ff ff  call   80484a6 <bossB>
14 80484ff:       83 c4 10        add    esp,0x10
15 8048502:       90              nop
16 8048503:       c9              leave
17 8048504:       c3              ret
eax: 0x40
```



Stack Frame

```
1  080484cc <bossA>:
2  80484cc:       55                      push   ebp
3  80484cd:       89 e5                   mov    ebp,esp
4  80484cf:       83 ec 18                sub    esp,0x18
5  80484d2:       8b 45 08                mov    eax,DWORD PTR [ebp+0x8]
6  ...
7  80484de:       89 45 f4                mov    DWORD PTR [ebp-0xc],eax
8  ...
9  80484ec:       e8 4f fe ff ff          call   8048340 <printf@plt>
10 ...
11 80484f4:       83 ec 0c                sub    esp,0xc
12 80484f7:       ff 75 f4                push   DWORD PTR [ebp-0xc]
13 80484fa:       e8 a7 ff ff ff          call   80484a6 <bossB>
14 80484ff:       83 c4 10                add    esp,0x10
15 8048502:       90                      nop
16 8048503:       c9                      leave
17 8048504:       c3                      ret
```

eax: 0x20