|  |  |  |
| --- | --- | --- |
| Week | Reverse Engineering Malware | Duration |
| 4 | Procedures | 60 mins |

Marks allocation: 2/100 for CA tutorial submission

**Lesson Objectives**

* Understand Procedures in assembly language

1. Which instruction pushes all of the 32-bit general-purpose registers on the stack?

**pusha**

2. Which instruction pushes the 32-bit EFLAGS register on the stack?

**pushf**

3. Which instruction pops the stack into the EFLAGS register?

**popf**

4. Suppose there were no PUSH instruction. Write a sequence of two other instructions that would accomplish the same as push eax.

Code example (32-bit mode):

**sub esp,4**

**mov [esp],eax**

5. (True/False): The RET instruction pops the top of the stack into the instruction pointer.

**True**

6. What will be the final value in EAX after these instructions execute?

push 5

push 6

pop eax

pop eax

**EAX = 5**

END