

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ ИТМО»**

Факультет безопасности информационных технологий

Дисциплина:
«Операционные системы»

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 7

Выполнила:

Нгуен Хонг Хань N3249

(подпись)

Проверил:

Савков Сергей Витальевич

(подпись)

Санкт-Петербург

2022г.

Задание

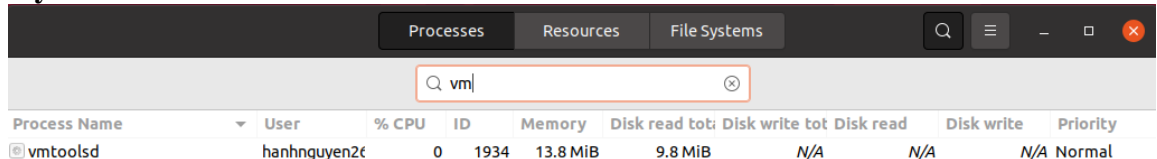
Перечислите все известные вам способы обнаружения работы в виртуальной машине. (>=5)

Сложный вариант (или)

1. Привести способ выхода из виртуальной машины
2. На ассемблере

1. Способы обнаружения работы в виртуальной машине.

1.1. System monitor



Process Name	User	% CPU	ID	Memory	Disk read tot	Disk write tot	Disk read	Disk write	Priority
vmttoolsd	hanhnguyen26	0	1934	13.8 MiB	9.8 MiB	N/A	N/A	N/A	Normal

1.2. lscpu

Команда `lscpu` отображает информацию об архитектуре процессора. При создании виртуальной машины в выводе должна появиться информация о поставщике гипервизора.

```
hanhnguyen26@ubuntu:~$ lscpu | grep "Hypervisor"
Hypervisor vendor: VMware
hanhnguyen26@ubuntu:~$
```

1.3. dmesg

`dmesg` используется для проверки содержимого или управления буфером кольца ядра. Действие по умолчанию - отобразить все сообщения из буфера кольца ядра

```
hanhnguyen26@ubuntu:~$ dmesg | grep -i virtual
[ 0.000000] DMI: VMware, Inc. VMware Virtual Platform/440BX Desktop Reference Platform, BIOS 6.00 11/12/2020
[ 0.019155] Booting paravirtualized kernel on VMware hypervisor
[ 1.798270] usb 2-1: Product: VMware Virtual USB Mouse
[ 2.124557] usb 2-2: Product: VMware Virtual USB Hub
[ 2.351911] input: VirtualPS/2 VMware VMMouse as /devices/platform/i8042/serio1/input/input4
[ 2.353322] input: VirtualPS/2 VMware VMMouse as /devices/platform/i8042/serio1/input/input3
[ 2.429302] input: VMware VMware Virtual USB Mouse as /devices/pci0000:00/0000:00:11.0/0000:02:00.0/usb2/2-1/2-1:1.0/0003:0E0F:0003.0001/input/input5
[ 2.429597] hid-generic 0003:0E0F:0003.0001: input,hidraw0: USB HID v1.10 Mouse [VMware VMwar e Virtual USB Mouse] on usb-0000:02:00.0-1/input0
[ 2.778842] ata4.00: ATAPI: VMware Virtual SATA CDRW Drive, 00000001, max UDMA/33
[ 2.858084] scsi 32:0:0:0: Direct-Access VMware, VMware Virtual S 1.0 PQ: 0 ANSI: 2
[ 3.476362] systemd[1]: Detected virtualization vmware.
hanhnguyen26@ubuntu:~$
```

1.4. dmidecode

это команда, которая используется для получения полезной информации об аппаратных компонентах вашей системы в удобочитаемом формате.

```
hanhnguyen26@ubuntu:~$ sudo dmidecode | grep -A3 "System Information"
System Information
    Manufacturer: VMware, Inc.
    Product Name: VMware Virtual Platform
    Version: None
hanhnguyen26@ubuntu:~$
```

1.5. lshw

Утилита lshw выводит на консоль полный список аппаратных компонентов системы вместе с информацией об устройствах.

```
hanhnguyen26@ubuntu:~$ sudo lshw | head
ubuntu
  description: Computer
  product: VMware Virtual Platform
  vendor: VMware, Inc.
  version: None
  serial: VMware-56 4d 5f 9d ab b4 d3 81-70 64 1f 2a 6f 6e 2e db
  width: 64 bits
  capabilities: smbios-2.7 dmi-2.7 smp vsyscall32
  configuration: administrator_password=enabled boot=normal frontpanel_password=unknown keyboard_password=unknown power-on_password=disabled uuid=564D5F9D-ABB4-D381-7064-1F2A6F6E2EDB
*-core
```

1.6. hostnamectl

Эта утилита позволяет нам запрашивать и изменять системное имя хоста и связанные с ним настройки. Мы также можем использовать команду hostnamectl для обнаружения технологии виртуализации.

```
hanhnguyen26@ubuntu:~$ hostnamectl
Static hostname: ubuntu
  Icon name: computer-vm
  Chassis: vm
  Machine ID: 9b76af3617d941cf94d30477eabd22e9
  Boot ID: 1324c225b1a74134a7ff28d663316a9c
  Virtualization: vmware
  Operating System: Ubuntu 20.04.3 LTS
  Kernel: Linux 5.13.0-39-generic
  Architecture: x86-64
```

1.7. systemd-detect-virt

Инструмент systemd-detect-virt обнаруживает технологию виртуализации и может отличить полную виртуализацию машины от аппаратной или контейнерной виртуализации.

```
hanhnguyen26@ubuntu:~$ systemd-detect-virt
vmware
```

1.8. Использование __cpuid()

- На платформах x64 программное обеспечение проверяет, работает ли оно в виртуализированной среде, выполняя инструкцию CPUID с входным значением (регистр EAX), равным 1. При выполнении инструкции CPUID код должен проверять бит 31 регистра ECX. Бит 31 — это бит присутствия гипервизора. Если установлен бит присутствия гипервизора, гипервизор присутствует. В неvirtуализированной среде бит присутствия гипервизора не установлен.
- Код программы

```
#include <iostream>
#include <intrin.h>
using namespace std;
bool isHypervisor() {
    int CPUInfo[4] = { -1 };
    __cpuid(CPUInfo, 1);
    if ((CPUInfo[2] >> 31) & 1) {
        return true;
    }
    return false;
}
```

```

int main() {
    if (isHypervisor()) {
        cout << "Virtual machine" << endl;
    }
    else {
        cout << "Physical machine" << endl;
    }
    return 0;
}

```

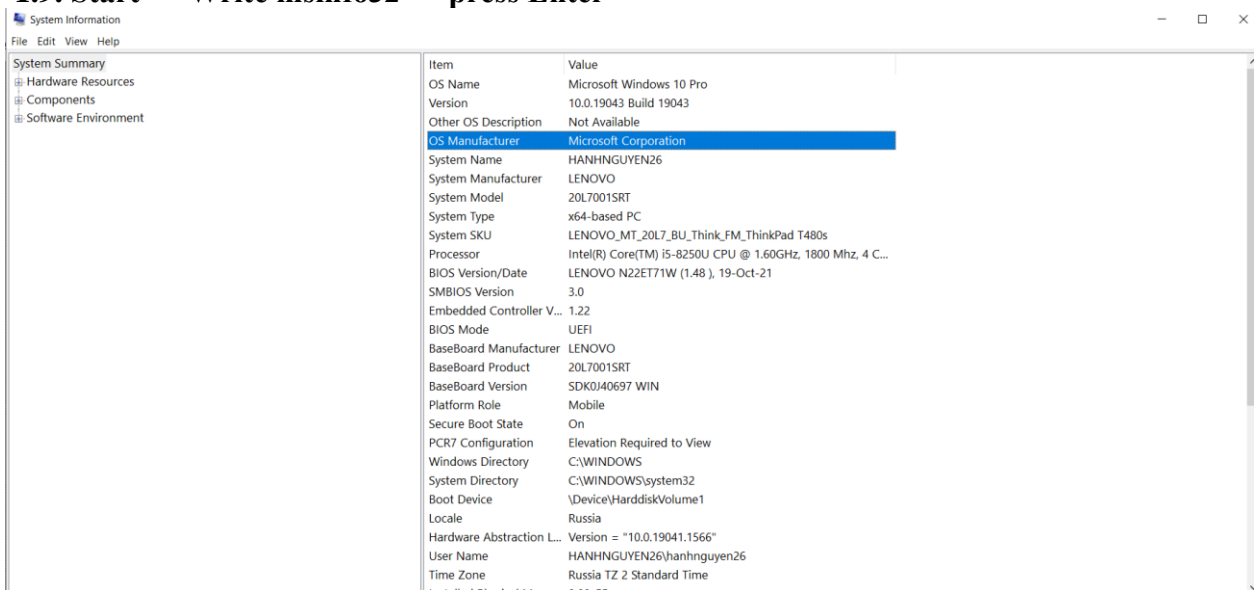
- **Результат**

```

C:\Users\hanhnguyen26\Desktop>.\a.exe
Physical machine

```

1.9. Start → Write msinfo32 → press Enter



1.10. powershell

```

PS C:\Users\hanhnguyen26> get-wmiobject win32_computersystem | fl model

model : 20L7001SRT

```

2. (Сложный вариант) Способы обнаружения работы в виртуальной машине на ассемблере (на язык nasm)

- Идей похож на идеи пункта 1.8 но реализуем на язык nasm (32-бит)
- Код программы

```

;nasm -felf64 lab7.asm -o a.o
;jld -g a.o
;./a.out

global _start

section .data
    VM:      db 'Virtual Machine',0xA
    VM_1:    equ $-VM
    PM:      db 'Physical Machine',0xA

```

```

        PM_1: equ $-PM

section .bss
section .text

_start:
        xor eax, eax            ; eax = 0
        mov eax, 1              ; eax = 1
        cpuid                   ; вызов cpuid
        bt ecx, 0x1f            ; проверка ecx[31]
        jc .VM                  ; если ecx[31] = 1, значит присутствия гипервизора
        jnc .PM                  ; если ecx[31] = 0, то это физическая машина

.VM:
        mov     edx, VM_1        ; длина строк
        mov     ecx, VM          ; адрес строк
        call .print

.PM:
        mov     edx, PM_1
        mov     ecx, PM

.print:
        mov     ebx, 1           ; файловый дескриптор 1 - с потоком стандартного вывода
        mov     eax, 4           ; Номер системного вызова (write())
        int     0x80             ; вывод результата

        mov     ebx, 0           ;
        mov     eax, 1           ; Номер системного вызова (exit())
        int     0x80             ; вызов exit()

hanhnguyen26@ubuntu:~$ nasm -f elf -g -F dwarf lab7.asm
hanhnguyen26@ubuntu:~$ ld -m elf_i386 -o lab7 lab7.o
hanhnguyen26@ubuntu:~$ ./lab7
Virtual Machine
hanhnguyen26@yuety25:~$ nasm -f elf64 lab7.asm -o a.o
hanhnguyen26@yuety25:~$ ld -g a.o
hanhnguyen26@yuety25:~$ ./a.out
Physical Machine

```

- Вывод
В ходе лабораторной работы я узнал о способах определения того, что ОС запущена на виртуальной машине.