

Міністерство освіти і науки України
Львівський національний університет імені Івана Франка
Факультет електроніки та комп'ютерних технологій

Кафедра радіоелектронних
і комп'ютерних систем

Звіт
про виконання лабораторних робіт №8
«Робота з програмними інтерфейсами»

Виконав студент
групи ФЕІ-23
Ковальчук Д. М.
Перевірів Сінькевич О. О.

Тема: Робота з програмними інтерфейсами файлових систем .

Мета: вивести статистику компютера використовуючи інтерфейсами файлових систем.

Завдання №1

Розробіть програму отримання системної інформації в ОС Linux, використовуючи файлову систему /proc. Програма повинна виводити на екран і записувати у файл інформацію про параметри центрального процесора, стан пам'яті тощо.

Code of part_1.c

```
#include <stdio.h>
#include <string.h>

int main ()
{
    FILE* fp;
    FILE* fm;

    char cpuinfo[830];
    size_t bytes_read;
    char* match;
    float clock_speed;

    fp = fopen ("/proc/cpuinfo", "r");
    bytes_read = fread (cpuinfo, 1, sizeof (cpuinfo), fp);

    fclose (fp);
    cpuinfo[bytes_read] == '\0';

    printf("\nCPU info\n\n%s\n\n", cpuinfo);
    char meminfo[140]

    fm = fopen ("/proc/meminfo", "r");
    bytes_read = fread (meminfo, 1, sizeof (meminfo), fm);
    fclose (fm);

    meminfo[bytes_read] == '\0';
    printf("\nMemory info\n\n%s\n\n", meminfo);

    return 0;
}
```

Результат роботи програми :

```
lab_8 : bash — Konsole
Файл  Зміни  Перегляд  Закладки  Параметри  Довідка
binchukchik@binchukchik-VirtualBox:~$ cd lab_8
binchukchik@binchukchik-VirtualBox:~/lab_8$ gcc part_1.c -o main
binchukchik@binchukchik-VirtualBox:~/lab_8$ ./main

CPU info

processor      : 0
vendor_id     : GenuineIntel
cpu family    : 6
model         : 61
model name    : Intel(R) Pentium(R) CPU 3825U @ 1.90GHz
stepping      : 4
cpu MHz       : 1895.609
cache size    : 2048 KB
physical id   : 0
siblings      : 1
core id       : 0
cpu cores     : 1
apicid        : 0
initial apicid : 0
fpu           : yes
fpu_exception : yes
cpuid level   : 20
wp            : yes
flags         : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse3
6 clflush mmx fxsr sse sse2 ht syscall nx rdtscp lm constant_tsc rep_good nopl xtopology
nonstop_tsc cpuid tsc_known_freq pni pclmulqdq monitor ssse3 cx16 pcid sse4_1 sse4_2 x2
apic movbe popcnt xsave rdrand hypervisor lahf_lm abm 3dnowprefetch invpcid_single pti f
sgsbase invpcid rdseed flush_l1d
bugs          : cpu_meltdown spectre_v1 spectre_v2 spec_store_bypass l1tf
bogomips      : 3791.21
clflush size  : 64
cache_alignm  : 64
address size

Memory info

MemTotal:      2038876 kB
MemFree:       427072 kB
MemAvailable:  1206800 kB
Buffers:       50456 kB
Cached:        894024 kB
SwapCached:    1264 kB
Active:        885008 kB
Inactive:      585956 kB
Active(anon):  302600 kB
Inactive(anon): 282044 kB
Active(file):  582408 kB
Inactive(file): 303912 kB
Unevictable:   16 kB
Mlocked:       16 kB
SwapTotal:     678096 kB
SwapFree:      666052 kB
Dirty:         104 kB
Writeback:      0 kB
AnonPages:     525880 kB
Mapped:        317032 kB
Shmem:         58152 kB
```

```

KReclaimable:      49840 kB
Slab:              100108 kB
SReclaimable:      49840 kB
SUnreclaim:        50268 kB
KernelStack:       4108 kB
PageTables:        12364 kB
NFS_Unstable:       0 kB
Bounce:            0 kB
WritebackTmp:       0 kB
CommitLimit:       1697532 kB
Committed_AS:      1998128 kB
VmallocTotal:      34359738367 kB
VmallocUsed:        0 kB
VmallocChunk:       0 kB
Percpu:            544 kB
HardwareCorrupted:  0 kB
AnonHugePages:      0 kB
ShmemHugePages:     0 kB
ShmemPmdMapped:     0 kB
CmaTotal:

```

Завдання №2.

Розробіть програму отримання системної інформації в ОС Windows XP, використовуючи програмний інтерфейс доступу до системного реєстру. Програма повинна виводити на екран і записувати у файл інформацію про стан і конфігурацію системи.

Code of part_2.c

```

#include "stdafx.h"
#include <conio.h>
#include <Windows.h>
#include <stdio.h>

int _tmain(int argc, _TCHAR* argv[])
{
    printf("\tConfiguration Of System\n");
    MEMORYSTATUS MemoryStatus;
    GlobalMemoryStatus(&MemoryStatus);
    printf("\tMemory Information:\n");
    printf("Phys.mem. in use: %d%\n",MemoryStatus.dwMemoryLoad);
    printf("Total size of phys.mem.: %u\n",MemoryStatus.dwTotalPhys);
    printf("Available size of phys.mem.: %u\n",MemoryStatus.dwAvailPhys);
    printf("Total size of page file: %u\n",MemoryStatus.dwTotalPageFile);
    printf("Available size of page file: %u\n",MemoryStatus.dwAvailPageFile);
    printf("Total user virtual memory size: %u\n",MemoryStatus.dwTotalVirtual);
    printf("Available user virtual memory size: %u\n",MemoryStatus.dwAvailVirtual);
    printf("\tDisplay devices:\n");
    DISPLAY_DEVICE DisplayDevice;
    DisplayDevice.cb=sizeof(DisplayDevice);
    for (UINT iIndex=0;EnumDisplayDevices(NULL,iIndex,&DisplayDevice,0);iIndex++)
    {
        *((char*)&DisplayDevice.StateFlags)=0;
        printf("%s\n",DisplayDevice.DeviceString);
    }
    printf("\tKeyboard type:\n");
    switch(GetKeyboardType(0))
    {
        case 1:printf("IBM PC/XT or compatible (83-key) keyboard\n");break;
        case 2:printf("Olivetti ICO (102 key) keyboard\n");break;
        case 3:printf("IBM PC/AT (84 key) or similar\n");break;
        case 4:printf("IBM enhanced (101 or 102 keys) keyboard\n");break;
        case 5:printf("NOKIA 1050 or similar\n");break;
        case 6:printf("NOKIA 9140 or similar\n");break;
        case 7:printf("Japanese keyboard\n");break;
    }
    HKEY BootKey;
    unsigned char Buffer[100];

```

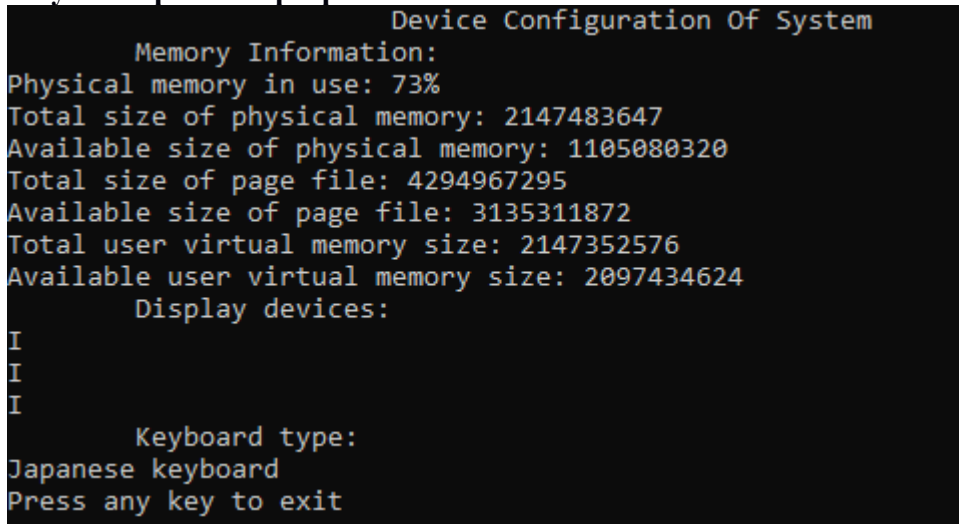
```

DWORD BufferSize=100;
ZeroMemory(Buffer,100);
if (!RegOpenKeyEx(HKEY_LOCAL_MACHINE,(LPCWSTR)"SOFTWARE\\Microsoft\\Windows
NT\\CurrentVersion\\WOW\\boot.description",0,KEY_QUERY_VALUE,&BootKey))
{
    if (!RegQueryValueEx(BootKey,(LPCWSTR)"mouse.drv",NULL,NULL,Buffer,&BufferSize))
    {
        printf("\tMouse type:\n");
        printf("%s\n",Buffer);
    }
}
ZeroMemory(Buffer,100);
BufferSize=100;
if (!RegQueryValueEx(BootKey,(LPCWSTR)"display.drv",NULL,NULL,Buffer,&BufferSize))
{
    printf("\tDisplay type:\n");
    printf("%s\n",Buffer);
}
ZeroMemory(Buffer,100);
BufferSize=100;
if (!RegOpenKeyEx(HKEY_LOCAL_MACHINE,(LPCWSTR)"HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0",0,KEY_Q
UERY_VALUE,&BootKey))
{
    if (!RegQueryValueEx(BootKey,(LPCWSTR)"ProcessorNameString",NULL,NULL,Buffer,&BufferSize))
    {
        printf("\tProcessorName:\n");
        printf("%s\n",Buffer);
    }
}

printf("Press any key to exit\n");
system("pause");
return 0;
}

```

Результат роботи програми:



```

Device Configuration Of System
Memory Information:
Physical memory in use: 73%
Total size of physical memory: 2147483647
Available size of physical memory: 1105080320
Total size of page file: 4294967295
Available size of page file: 3135311872
Total user virtual memory size: 2147352576
Available user virtual memory size: 2097434624
Display devices:
I
I
I
Keyboard type:
Japanese keyboard
Press any key to exit

```

Висновок: На даній лабораторній роботі я навчився роботі з програмними інтерфейсами файлових систем та вивів статистику компютера використовуючи інтерфейсами файлових систем. Це було доволі цікаво та я дізнався багато нових функцій реалізованих в стандартних бібліотеках мови C. Реалізував потрібні програми та продемонстрував їхню роботу у скріншотах вище.