

Код программы:

```
#include <windows.h>
#include <tlhelp32.h>
#include <psapi.h>
#include <iostream>
#include <vector>

void DisplayMemoryStatus() {
    MEMORYSTATUS memStatus;
    GlobalMemoryStatus(&memStatus);

    std::cout << "Общий объем физической памяти: " <<
memStatus.dwTotalPhys / (1024 * 1024) << " МБ" << std::endl;
    std::cout << "Доступная физическая память: " <<
memStatus.dwAvailPhys / (1024 * 1024) << " МБ" << std::endl;
    std::cout << "Общий объем виртуальной памяти: " <<
memStatus.dwTotalVirtual / (1024 * 1024) << " МБ" << std::endl;
    std::cout << "Доступная виртуальная память: " <<
memStatus.dwAvailVirtual / (1024 * 1024) << " МБ" << std::endl;
}

void EnumerateProcesses() {
    HANDLE hProcessSnapshot;
    PROCESSENTRY32 pe32;
    hProcessSnapshot = CreateToolhelp32Snapshot(TH32CS_SNAPPROCESS, 0);

    if (hProcessSnapshot == INVALID_HANDLE_VALUE) {
        std::cerr << "Ошибка создания снимка процессов!" << std::endl;
        return;
    }

    pe32.dwSize = sizeof(PROCESSENTRY32);
    if (!Process32First(hProcessSnapshot, &pe32)) {
        std::cerr << "Ошибка получения первого процесса!" << std::endl;
        CloseHandle(hProcessSnapshot);
        return;
    }

    do {
        std::wcout << L"Имя процесса: " << pe32.szExeFile << L" (PID: "
<< pe32.th32ProcessID << L")" << std::endl;
```

```

        HANDLE hProcess = OpenProcess(PROCESS_QUERY_INFORMATION |
PROCESS_VM_READ, FALSE, pe32.th32ProcessID);
        if (hProcess != NULL) {
            PROCESS_MEMORY_COUNTERS pmc;
            if (GetProcessMemoryInfo(hProcess, &pmc, sizeof(pmc))) {
                std::cout << "    WorkingSetSize: " <<
pmc.WorkingSetSize / 1024 << " KB" << std::endl;
                std::cout << "    PagefileUsage: " << pmc.PagefileUsage
/ 1024 << " KB" << std::endl;
            }
            CloseHandle(hProcess);
        }
    } while (Process32Next(hProcessSnapshot, &pe32));

    CloseHandle(hProcessSnapshot);
}

int main() {
    DisplayMemoryStatus();
    EnumerateProcesses();
    return 0;
}

```

Пояснение кода:

- А) DisplayMemoryStatus: Функция для получения и вывода информации о физической и виртуальной памяти с использованием функции GlobalMemoryStatus.
- Б) EnumerateProcesses: Функция для создания снимка процессов, их перебора и получения информации о памяти каждого процесса с использованием функций Process32First, Process32Next, OpenProcess и GetProcessMemoryInfo.

Результат:

```
[Running] cd "d:\db_go\oc\" && g++ main.cpp -o mai
```

Общий объем физической памяти: 16183 МБ

Доступная физическая память: 7452 МБ

Общий объем виртуальной памяти: 134217727 МБ

Доступная виртуальная память: 134213574 МБ

WorkingSetSize: 32452 КБ

PagefileUsage: 6688 КБ

WorkingSetSize: 23628 КБ

PagefileUsage: 8836 КБ

WorkingSetSize: 12992 КБ

PagefileUsage: 3764 КБ

WorkingSetSize: 8668 КБ

PagefileUsage: 2036 КБ

WorkingSetSize: 33160 КБ

PagefileUsage: 9240 КБ