

**Санкт-Петербургский национальный  
исследовательский университет  
информационных технологий,  
механики и оптики**

Кафедра информатики и прикладной математики

**Системное программное обеспечение**

Лабораторная работа №4

“Процессы”



Старался: **Шкаруба Н.Е.**

Проверил: **Зыков А.Г.**

Группа **P3218**

## Требования:

Разработать алгоритм и программу, реализующую следующие действия:

- Сгенерировать (или использовать из предыдущих заданий) массив данных, который будет использоваться для сортировки и поиска в нём элементов.
- Создать три отдельных процесса, выполняющих одно и тоже:
  - а) Сортировка исходного массива по заданному в командной строке шаблону.
  - б) Поиск элемента, указанного также в командной строке
- Вывести временные характеристики и ID процессов.
- Определить количество открытых дескрипторов.

## Исходный код программы:

main.cpp

```
#include "sortAlgorithms/Algorithms.h"
#include <sys/types.h>
#include <sys/wait.h>
#include <unistd.h>
#include <stdio.h>
#include <fstream>
#include <vector>
#include <string.h>
#include <dirent.h>

using namespace std;

int getDescriptorsCount(int p_id) {
    // It's not my shitty code. And i am fill of SPO-hatred to waste time refactoring this
    DIR *dp;
    int i = 0;
    struct dirent* ep;
    string s = "/proc/" + std::to_string(p_id) + "/fd";
    dp = opendir (s.c_str());

    if (dp != NULL) {
        while (ep = readdir (dp))
            i++;

        (void) closedir (dp);
    } else
        perror ("Couldn't open the directory");

    return i;
}

double getCpuTime(){
    return (double)clock() / CLOCKS_PER_SEC;
}

vector<int> getIntegersFromFile(string filename) {
    ifstream inStream(filename);
    vector<int> integers;

    string integer;
```

```

while(inStream >> integer) {
    integers.push_back(stoi(integer));
}

return integers;
}

void writeIntegersToFile(vector<int> integers, string filename) {
    ofstream outStream(filename, fstream::out | fstream::trunc);

    for(vector<int>::iterator it = integers.begin(); it != integers.end(); it++)
        outStream << *it << endl;
}

int main(int argc, char* argv[]) {
    string AlgorithmName, filename;
    int numberToFind = rand();
    int status;

    for(size_t i = 1; i < argc; i++) {
        if(strcmp(argv[i], "--sort-algorithm") == 0)
            AlgorithmName = argv[++i];
        else if(strcmp(argv[i], "--missed-number") == 0)
            numberToFind = stoi(argv[++i]);

        if(i == argc-1)
            filename = argv[i];
    }

    printf("This is the parent - caller. He is getting integers from file and forking other processes. P_id: %d, Cpu time is: %f\n", getpid(),
    getCpuTime());
    vector<int> numbers = getIntegersFromFile(filename);
    printf("Parent's descriptors count: %d\n\n", getDescriptorsCount(getpid()));

    for(int i = 0; i < 3; i++) {
        if(fork() == 0) {
            printf("Hey, I am the %d child, and my id is %d, my descriptors count is %d, Cpu time is: %f\n", i, getpid(),
            getDescriptorsCount(getpid()), getCpuTime());

            if (AlgorithmName == "merge")
                MergeSort(&numbers.front(), numbers.size());
            else if (AlgorithmName == "quick")
                QuickSort(&numbers.front(), numbers.size());
            else
                perror("No algorithm provided!");

            for(vector<int>::iterator it = numbers.begin(); it != numbers.end(); it++) {
                if(*it == numberToFind) {
                    printf("Found %d\n", numberToFind);
                    exit(0);
                }
            }
            printf("Can't find %d\n", numberToFind);
            exit(0);
        }
    }

    int processesExited = 0;
    while (processesExited++ < 3) {
        wait(&status);
        printf("Child exited\n");
    }

    return 0;
}

```

## Тестирующий скрипт:

```
#!/bin/bash

tput setaf 1;
echo "0) Generating 100500 integers in file randomNumbers.txt"
tput sgr0
rm -rf randomNumbers.txt
touch randomNumbers.txt
for i in `seq 1 100500`; do
    echo $RANDOM >> randomNumbers.txt
done

tput setaf 1;
echo "1) head randomNumbers.txt"
tput sgr0
head randomNumbers.txt

tput setaf 1;
echo "2) Running SortAndFind program:"
tput sgr0
./SortAndFind --sort-algorithm quick --missed-number 1230 randomNumbers.txt
```

## Вывод скрипта:

```
nikita@Pluto:~/Code/ITMO/Course 2/System_software/Lab4(Processes)/workspace(System_Software-develo)$ ./doSorts.sh
0) Generating 100500 integers in file randomNumbers.txt
1) head randomNumbers.txt
20677
29729
19220
16469
18452
7557
18251
17695
1071
16453
2) Running SortAndFind program:
This is the parent - caller. He is getting integers from file and forking other processes. P_id: 4407, Cpu time is: 0.003494
Parent's descriptors count: 6

Hey, I am the 0 child, and my id is 4408, my descriptors count is 6, Cpu time is: 0.000072
Hey, I am the 1 child, and my id is 4409, my descriptors count is 6, Cpu time is: 0.000048
Hey, I am the 2 child, and my id is 4410, my descriptors count is 6, Cpu time is: 0.000023
Found 1230
Child exited
Found 1230
Found 1230
Child exited
Child exited
```