

Московский Авиационный Институт
(Национальный Исследовательский Университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Лабораторная работа №3 по курсу
«Операционные системы»

Студент: Пивницкий Д.С.

Группа: М8о–206Б–19

Вариант: 19

Преподаватель: Соколов Андрей Алексеевич

Оценка: _____

Дата: _____

Подпись: _____

Москва, 2020.

Постановка задачи

Цель работы

Приобретение практических навыков в:

- Управление потоками в ОС
- Обеспечение синхронизации между потоками

Задание

Составить программу на языке Си, обрабатывающую данные в многопоточном режиме. При обработке использовать стандартные средства создания потоков операционной системы (Windows/Unix). Ограничение потоков должно быть задано ключом запуска программы.

Необходимо уметь продемонстрировать количество потоков, используемых программой, с помощью стандартных средств операционной системы.

Привести исследование зависимости ускорения и эффективности алгоритма от входящих данных и количества потоков. Объяснить получившиеся результаты.

Вариант 19: Необходимо реализовать проверку числа на простоту при помощи алгоритма «решето Эратосфена».

Общие сведения о программе

Программа написана на языке Си в UNIX-подобной операционной системе (Linux Mint). Для компиляции программы требуется указать ключ `-pthread`. Для запуска программы в качестве 1 аргумента командной строки необходимо указать количество потоков, которые могут быть использованы программой, а в качестве 2 аргумента — тестируемое число.

Программа содержит две глобальные переменные – массив для решета Эратосфена и число, простоту которого проверяем. Переменные объявлены глобальными, чтобы любой поток имел к ним доступ.

Программа включает в себя потоковую функцию

`void*sieve_step(void* i)`, в которой помечаются все числа решета, кратные `i`. Так как все потоки программы работают в одном и том же пространстве памяти, аргументы для передачи потоковой функции хранятся по разным адресам (в массиве, размер которого равен количеству потоков).

В программе предусмотрена проверка на системные ошибки – ошибки выделения памяти, ошибки запуска.

Общий метод и алгоритм решения.

При запуске программы у пользователя запрашивается число `num`, которое необходимо проверить на простоту. Проверить на простоту можно только неотрицательное число.

Из аргументов командной строки берётся количество потоков, которое может использовать программа и само число для проверки. Производится выделение памяти для массива потоков, для массива аргументов потоковой функции и для самого решета. Решето представляет собой массив символов `sieve` (т.к. размер символьного типа `char` минимальный). `sieve[i]` равно нулю, если число простое и единице в противном случае.

По определению числа 0 и 1 не являются простыми, поэтому сразу помечаем их единицами в решете. Необходимо проверить все числа от 2 до `num` включительно. Если ячейка решета, соответствующая числу `i`, равна нулю, то это число простое и требуется «вычеркнуть» (позначить единицей) все числа, кратные `i`. Эта задача и делегируется другим потокам.

Потоковая функция `sieve_step` принимает на вход число `i` и помечает единицами все числа, кратные `i`. Заметим, что первое число, кратное `i` и которое еще НЕ было помечено единицей – это число i^2 . Для ускорения

алгоритма начнём проверку именно с этого числа и будем помечать каждое i -ое число, начиная с i^2 . По этой же причине в главной функции перебор элементов решета будет вестись от 2 до корня из i . Потоки не смогут повлиять на работу друг друга, поэтому mutex не используется.

Укажем правило, по которому будет выбираться поток для выполнения функции. Заведём переменную `cur_thread`, изначально равную нулю. Для выполнения функции будет создаваться поток с индексом `cur_thread % threads_num`, где `threads_num` – общее количество потоков. Таким образом, потоки будут использоваться в порядке закольцованной очереди. Когда `cur_thread` становится больше количества потоков, потоки начинают использоваться повторно. Во избежание ситуации, когда задача будет делегирована потоку, работа которого еще не окончена, будем дожидаться окончания работы потока. После делегирования задач, переменная `cur_thread` инкрементируется.

После обработки всего решета необходимо дождаться окончания работы всех активных потоков. После этого необходимо посмотреть число в `sieve[num]` и сделать вывод о простоте этого числа.

Основные файлы программы

lab3.c:

```
#include <stdio.h>
#include <stdlib.h>
#include <pthread.h>

char* sieve;
long long num;

// функция потока – маркировка чисел, кратных i
void* sieve_step(void* i_void) {
    long long i = *(long long*)i_void;
    for (long long j = i * i; j <= num; j += i) {
        sieve[j] = 1;
    }
}
```

```

    }
    pthread_exit(NULL);
}

int main(int argc, char* argv[]) {
    if (argc != 3) {
        printf("Syntax: ./out Number_of_threads Number_for_test\n");
        exit(1);
    }
    int threads_num = atoi(argv[1]);
    num = atoi(argv[2]);

    pthread_t* threads = (pthread_t*)calloc(threads_num, sizeof(pthread_t));
    if (threads == NULL) {
        printf("Can't allocate space for threads\n");
        exit(2);
    }

    // массив аргументов, которые будут переданы в функцию потока
    long long* args = (long long*)malloc(threads_num* sizeof(long long));
    if (args == NULL) {
        printf("Can't create an array for arguments for threads\n");
        exit(3);
    }

    // создаем массив, заполненный 0 для решета
    // 0 - простое число, 1 - непростое число
    sieve = (char*)calloc((num + 1), sizeof(char));
    if (sieve == NULL) {
        printf("Can't create an array for sieve\n");
        exit(3);
    }

    // маркировка чисел, не являющихся простыми по определению
    sieve[0] = 1;
    sieve[1] = 1;

    int cur_thread = 0; // id текущего потока

```

```

for (long long i = 2; i * i <= num; ++i) {
    if (sieve[i] == 1) { // пропуск не простых чисел
        continue;
    }
    if (cur_thread >= threads_num) {
        // мы должны подождать, пока работает необходимый поток
        pthread_join(threads[cur_thread % threads_num], NULL);
    }

    args[cur_thread % threads_num] = i; // копирование аргумента для
    функции потока в специальный массив
    pthread_create(&threads[cur_thread % threads_num], NULL, sieve_step,
    &args[cur_thread % threads_num]);
    ++cur_thread;
}

// ждем все потоки
for (int i = 0; i < threads_num; ++i) {
    pthread_join(threads[i], NULL);
}

if (sieve[num] == 1) {
    printf("%lld is not a prime number\n", num);
}
else {
    printf("%lld is a prime number\n", num);
}

free(sieve);
free(threads);
free(args);
}

```

Пример работы

daniel@daniel-Ideapad-Z570: ~ cd os/oslab3

daniel@daniel-Ideapad-Z570: ~ oslab3 gcc lab3.c -o out -pthread

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out

Syntax: ./out Number_of_threads Number_for_test

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 10

10 is not a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 11

11 is a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 953

953 is a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 1

1 is not a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 0

0 is not a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 3

3 is a prime number

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 -3

Can't create an array for sieve

daniel@daniel-Ideapad-Z570: ~ oslab3 ./out 1 10736762

10736762 is not a prime number

(py37) → oslab3 time ./out 1 123123123

123123123 is not a prime number

./out 1 123123123 2,69s user 0,16s system 100% cpu 2,850 total

(py37) → oslab3 time ./out 2 123123123

123123123 is not a prime number

./out 2 123123123 2,69s user 0,12s system 192% cpu 1,459 total

(py37) → oslab3 time ./out 3 123123123

123123123 is not a prime number

./out 3 123123123 2,81s user 0,13s system 279% cpu 1,049 total

(py37) → oslab3 time ./out 4 123123123

123123123 is not a prime number

./out 4 123123123 3,22s user 0,12s system 368% cpu 0,906 total

(py37) → oslab3 time ./out 5 123123123

123123123 is not a prime number

./out 5 123123123 3,43s user 0,10s system 437% cpu 0,807 total

(py37) → oslab3 time ./out 6 123123123

123123123 is not a prime number

./out 6 123123123 4,48s user 0,21s system 532% cpu 0,882 total

(py37) → oslab3 time ./out 7 123123123

123123123 is not a prime number

./out 7 123123123 5,11s user 0,18s system 600% cpu 0,881 total

```
(py37) → oslab3 time ./out 8 123123123
123123123 is not a prime number
./out 8 123123123 5,06s user 0,27s system 636% cpu 0,837 total

(py37) → oslab3 time ./out 9 123123123
123123123 is not a prime number
./out 9 123123123 6,94s user 0,29s system 688% cpu 1,051 total

(py37) → oslab3 time ./out 100 123123123
123123123 is not a prime number
./out 100 123123123 5,99s user 0,48s system 713% cpu 0,907 total

(py37) → oslab3 time ./out 100000 123123123
123123123 is not a prime number
./out 100000 123123123 6,03s user 0,53s system 741% cpu 0,885 total

(py37) → oslab3 time ./out 10000000 123123123
123123123 is not a prime number
./out 10000000 123123123 6,97s user 0,44s system 684% cpu 1,082 total
```

Проверка правильного выделения кол-ва потоков во время процесса работы программы, стоит учитывать что выводя занятое количество потоков процессом всегда показывает на 1 больше указанного в аргументе программы или меньше при самом исполнении программы, но это значение не постоянно

Это связано с тем что ещё считается основной поток
консоль 1:

```
(py37) → oslab3 ./out 1 1073676287
1073676287 is a prime number
```

консоль 2:

```
(py37) → ~ ps aux | grep ./out
daniel  6118  0.1  3.3 15418004 266784 ?   Sl   09:49   1:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-
keys=64052a8a-6604-4b72-a365-
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-
```



```

resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-
integration --webview-tag --no-sandbox --no-zygote --native-window-open
--preload=/usr/share/code/resources/app/out/vs/base/parts/sandbox/electron-browser/preload.js --
background-color=#160024 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-
client-id=5 --no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100
daniel 6182 0.0 1.7 4707620 143932 ? SI 09:49 0:29 /usr/share/code/code --inspect-port=0 /usr/
share/code/resources/app/out/bootstrap-fork --type=extensionHost
daniel 6183 0.0 0.8 4619904 65980 ? SI 09:49 0:01 /usr/share/code/code
/usr/share/code/resources/app/out/bootstrap-fork --type=watcherService
daniel 6221 0.0 1.4 4783232 120588 ? SI 09:49 0:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-
keys=64052a8a-6604-4b72-a365-
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-
resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-
integration --no-sandbox --no-zygote --native-window-open --preload=/usr/share/code/resources/app/
out/vs/base/parts/sandbox/electron-browser/preload.js --background-color=#160024 --disable-blink-
features=Auxclick --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=7
--no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100
daniel 9718 92.5 11.6 1063412 944100 pts/0 Sl+ 21:22 0:01 ./out 1 1073676287
daniel 9721 0.0 0.0 15660 1076 pts/1 S+ 21:22 0:00 grep --color ./out
(py37) → ~ ps -o nlwp 9718
NLWP
2

```

консоль 1:

```

(py37) → oslab3 ./out 25 1073676287
1073676287 is a prime number

```

консоль 2:

(py37) → ~ ps aux | grep ./out

```
daniel  6118  0.1  3.2 15418004 266524 ?    Sl  09:49  1:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-
keys=64052a8a-6604-4b72-a365-
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-
resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-
integration --webview-tag --no-sandbox --no-zygote --native-window-open
--preload=/usr/share/code/resources/app/out/vs/base/parts/sandbox/electron-browser/preload.js --
background-color=#160024 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-
client-id=5 --no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100
daniel  6182  0.0  1.7 4707620 143932 ?    Sl  09:49  0:29 /usr/share/code/code --inspect-port=0 /usr/
share/code/resources/app/out/bootstrap-fork --type=extensionHost
daniel  6183  0.0  0.8 4619904 65980 ?    Sl  09:49  0:01 /usr/share/code/code
/usr/share/code/resources/app/out/bootstrap-fork --type=watcherService
daniel  6221  0.0  1.4 4783232 120772 ?    Sl  09:49  0:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-
keys=64052a8a-6604-4b72-a365-
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-
resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-
integration --no-sandbox --no-zygote --native-window-open --preload=/usr/share/code/resources/app/
out/vs/base/parts/sandbox/electron-browser/preload.js --background-color=#160024 --disable-blink-
features=Auxclick --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=7
--no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100
daniel  20653  628  8.4 1465016 684064 pts/0  Sl+  21:29  0:18 ./out 50 1073676287
```

daniel 20705 0.0 0.0 15660 1092 pts/1 R+ 21:29 0:00 grep --color ./out

(py37) → ~ ps -o nlwp 20653

NLWP

26

консоль 1:

(py37) → oslab3 ./out 50 1073676287

1073676287 is a prime number

консоль 2:

(py37) → ~ ps aux | grep ./out

daniel 6118 0.1 3.2 15418004 266524 ? Sl 09:49 1:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-
keys=64052a8a-6604-4b72-a365-
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-
resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-
integration --webview-tag --no-sandbox --no-zygote --native-window-open
--preload=/usr/share/code/resources/app/out/vs/base/parts/sandbox/electron-browser/preload.js --
background-color=#160024 --num-raster-threads=4 --enable-main-frame-before-activation --renderer-
client-id=5 --no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100
daniel 6182 0.0 1.7 4707620 143932 ? Sl 09:49 0:29 /usr/share/code/code --inspect-port=0 /usr/
share/code/resources/app/out/bootstrap-fork --type=extensionHost
daniel 6183 0.0 0.8 4619904 65980 ? Sl 09:49 0:01 /usr/share/code/code
/usr/share/code/resources/app/out/bootstrap-fork --type=watcherService
daniel 6221 0.0 1.4 4783232 120772 ? Sl 09:49 0:10 /usr/share/code/code --type=renderer --
disable-color-correct-rendering --no-sandbox --field-trial-
handle=14512560717055666749,16235639312606079544,131072 --enable-
features=WebComponentsV0Enabled --disable-features=SpareRendererForSitePerProcess --lang=ru --

```
enable-crash-reporter=64052a8a-6604-4b72-a365-c17ea0953ad9,no_channel --global-crash-  
keys=64052a8a-6604-4b72-a365-  
c17ea0953ad9,no_channel,_companyName=Microsoft,_productName=VSCode,_version=1.52.1 --  
standard-schemes=vscode-webview,vscode-webview-resource,vscode-file --secure-schemes=vscode-  
webview,vscode-webview-resource,vscode-file --bypasscsp-schemes --cors-schemes=vscode-  
webview,vscode-webview-resource,vscode-file --fetch-schemes=vscode-webview,vscode-webview-  
resource,vscode-file --service-worker-schemes --app-path=/usr/share/code/resources/app --node-  
integration --no-sandbox --no-zygote --native-window-open --preload=/usr/share/code/resources/app/  
out/vs/base/parts/sandbox/electron-browser/preload.js --background-color=#160024 --disable-blink-  
features=Auxclick --num-raster-threads=4 --enable-main-frame-before-activation --renderer-client-id=7  
--no-v8-untrusted-code-mitigations --shared-files=v8_snapshot_data:100  
daniel 13559 507 12.9 1327796 1049672 pts/0 Sl+ 21:28 0:20 ./out 25 1073676287  
daniel 13590 0.0 0.0 15660 1036 pts/1 S+ 21:28 0:00 grep --color ./out  
(py37) → ~ ps -o nlwp 13559
```

NLWP

Исследование ускорения и эффективности

Для исследования ускорения и эффективности параллельного решета Эратосфена замерим время работы программы для следующих входных данных:

1. 10000
2. 1000000
3. 1000000000

Время работы программы будет замеряться стандартной утилитой time. Нужно учитывать, что время работы может варьироваться в небольших пределах из-за постоянной работы фоновых процессов. Результаты будут занесены в таблицы.

Таблица 1. Исследование обработки числа 10000

Количество потоков (n)	Время работы программы (T_n), сек	Ускорение ($S_n = T_1 / T_n$)	Эффективность ($X_n = S_n / n$)
1	0,010	-	-
2	0,009	1,11	0,56
3	0,008	1,25	0,42
4	0,008	1,25	0,31
5	0,008	1,25	0,25
6	0,008	1,25	0,21
7	0,008	1,25	0,18
8	0,008	1,25	0,16
9	0,008	1,25	0,14
10	0,007	1,43	0,14

Таблица 2. Исследование обработки числа 1000000

Количество потоков (n)	Время работы программы (T _n), сек	Ускорение (S _n = T ₁ / T _n)	Эффективность (X _n = S _n / n)
1	0,033	-	-
2	0,020	1,65	0,83
3	0,016	2,06	0,69
4	0,018	1,83	0,46
5	0,018	1,83	0,37
6	0,016	2,06	0,34
7	0,018	1,83	0,26
8	0,016	2,06	0,26
9	0,015	2,2	0,24
10	0,016	2,06	0,21

Таблица 3. Исследование обработки числа 1000000000

Количество потоков (n)	Время работы программы (T _n), сек	Ускорение (S _n = T ₁ / T _n)	Эффективность (X _n = S _n / n)
1	21,308	-	-
2	15,527	1,37	0,69
3	13,948	1,53	0,51
4	13,323	1,60	0,40
5	13,706	1,55	0,31
6	14,048	1,52	0,25
7	13,708	1,55	0,22

8	16,457	1,29	0,16
9	17,098	1,25	0,14
10	17,563	1,21	0,12

По этим данным можно увидеть, что значительный выигрыш по времени мы имеем только при обработке действительно больших чисел (порядка 10^7 и выше). Оптимальнее всего использовать 2-3 потока. Таким образом можно ускорить время работы программы примерно в полтора раза. Средняя эффективность составляет 0,5. Если использовать большее количество потоков, то выигрыш от параллельной обработки будет перекрываться затратами на создание и регулирование потоков.

Вывод

Язык Си позволяет пользователю взаимодействовать с потоками операционной системы. Для этого на Unix-подобных системах требуется подключить библиотеку `pthread.h`.

Создание потоков происходит быстрее, чем создание процессов, а все потоки используют одну и ту же область данных. Поэтому многопоточность – один из способов ускорить обработку каких-либо данных: выполнение однотипных, не зависящих друг от друга задач, можно поручить отдельным потокам, которые будут работать параллельно.

Средствами языка Си можно совершать системные запросы на создание потока, ожидания завершения потока, а также использовать различные примитивы синхронизации.

В данной лабораторной работе был реализован и исследован алгоритм проверки числа на простоту при помощи решета Эратосфена. Установили, что при использовании двух-трёх потоков можно получить выигрыш по времени примерно в полтора раза, что значительно ускоряет обработку большого количества чисел. Но при использовании большего количества потоков ускорение не будет большим, так как операционной системе

приходится тратить больше времени на выделение памяти под потоки и на их регулирование.