

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего
образования

ТОМСКИЙ УНИВЕРСИТЕТ СИСТЕМ УПРАВЛЕНИЯ И
РАДИОЭЛЕКТРОНИКИ (ТУСУР)

Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

ОДНОПРОХОДНЫЕ АЛГОРИТМЫ

Отчёт по практической работе №2 по дисциплине «Технологии и методы
программирования»

Студент гр.728-2

_____ Полонский Е. В.

9 марта 2020 г.

Руководитель

Аспирант кафедры КИБЭВС

_____ Перминов П. В.

1 Введение

Целью данной работы является получения навыков в работе с git и gcc, написание двух простых программ на языке программирования C, по теме одно-проходные алгоритмы.

Задание:

1. Вариант 8: Количество неположительных чисел. На вход подаётся число $n \in \mathbb{N} : n \leq 2147483647$, а также n чисел $x_i \in \mathbb{Z} : |x_i| \leq 2147483647$ для любого целого i от 1 до n . Вывести значение $x_i : x_i \leq 0$;

2. Вариант 18: Сумма последовательности 2. На вход подаётся число $n \in \mathbb{N} : n \leq 2147483647$, а также n чисел $x_i \in \mathbb{Z} : |x_i| \leq 2147483647$ для любого целого i от 1 до n . Вывести значение $\sum_{i=1}^n (x_i - i)$.

2 Ход работы

Для выполнения работы сперва был актуализирован репозиторий, клонированный при выполнении практики 1. Для этого были выполнены следующие команды (Рисунок 2.1):

- `git remote add upstream git@gitlab.com:tusur_fb_timp/practices.git` (Добавление нового удаленного репозитория upstream);
- `git pull upstream` (Получение изменений с upstream);
- `git merge upstream/master` (Слияние upstream/master с локальной веткой master);
- Из-за ошибок слияния пришлось исправить `.gitlab-ci.yml`, далее выполнить `git add .; git commit` (Создает коммит слияния);
- `git push` (Отправить изменения в Origin/master).

```

/ practices ➤ master ● git remote add upstream git@gitlab.com:tusur_fb_timp/practices.git
/ practices ➤ master ● git pull upstream
remote: Enumerating objects: 19, done.
remote: Counting objects: 100% (19/19), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 16 (delta 5), reused 0 (delta 0), pack-reused 0
Распаковка объектов: 100% (16/16), 4.24 KiB | 149.00 KiB/s, готово.
Из gitlab.com:tusur_fb_timp/practices
* [новая ветка] master -> upstream/master
Вы попросили получить изменения со внешнего репозитория «upstream», но не указали ветку. Так как это не репозиторий по умолчанию для вашей текущей ветки, вы должны указать ветку в командной строке.
/ practices ➤ master ● git merge upstream/master
Автослияние .gitlab-ci.yml
КОНФЛИКТ (содержимое): Конфликт слияния в .gitlab-ci.yml
Не удалось провести автоматическое слияние; исправьте конфликты и сделайте коммит результата.
/ practices ➤ master ●+ >M< git commit
U .gitlab-ci.yml
error: Невозможно закоммитить, так как у вас имеются не слитые файлы.
подсказка: Исправьте их в рабочем каталоге, затем запустите «git add/rm <файл>»,
подсказка: чтобы пометить исправление и сделайте коммит.
fatal: Выход из-за неразрешенного конфликта.
/ practices ➤ master ●+ >M< git add .
/ practices ➤ master + >M< git commit
[master c2ddd0f] Merge remote-tracking branch 'upstream/master'
/ practices ➤ master ➤ git push
Перечисление объектов: 32, готово.
Подсчет объектов: 100% (31/31), готово.
При сжатии изменений используется до 4 потоков
Сжатие объектов: 100% (20/20), готово.
Запись объектов: 100% (22/22), 5.13 KiB | 1.71 MiB/s, готово.
Всего 22 (изменения 7), повторно использовано 0 (изменения 0)
To gitlab.com:Zorgy/practices.git
a61adf6..c2ddd0f master -> master

```

Рисунок 2.1 – Актуализация репозитория

Следующим шагом было написано 2 программы:

- Nonpositive (Рисунок 2.2), решение задания 8;

– Sum (Рисунок 2.3), решение задания 18.

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

int main(void)
{
    int N, a, Ans = 0;

    int num = scanf("%d", &N);
    assert(num > 0);

    int i = 1;
    for(i; i <= N; i++)
    {
        num = scanf("%d", &a);
        assert(num > 0);

        if(a <= 0){
            Ans++;
        }
    }
    printf("%d", Ans);
    return 0;
}
```

Рисунок 2.2 – Программа Nonpositive

```
#include <stdio.h>

int main (void)
{
    int N, a, Ans = 0;
    int num = scanf("%d", &N);
    assert(num > 0);

    int i = 1;
    for(i; i <= N; i++)
    {
        num = scanf("%d", &a);
        assert(num > 0);

        Ans += a-i;
    }
    printf("%d", Ans);
    return 0;
}
```

Рисунок 2.3 – Программа Sum

Ниже приведены блок-схемы программ (Рисунок 2.4, 2.5)

Для компиляции программ использовалась команда gcc имя файла

Далее исходные коды программ были запущены на гитлаб командой git push. Все пайплайны были пройдены успешно (Рисунок 2.6).

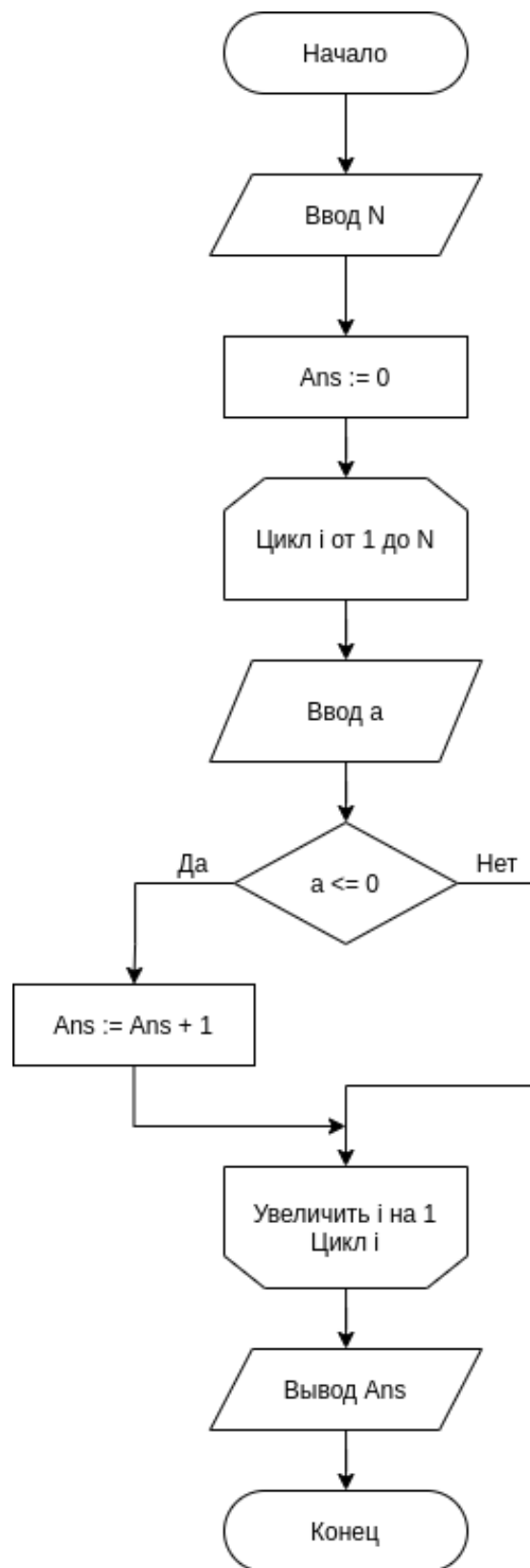


Рисунок 2.4 – Блок-схема программы Nonpositive



Рисунок 2.5 – Блок-схема программы Sum

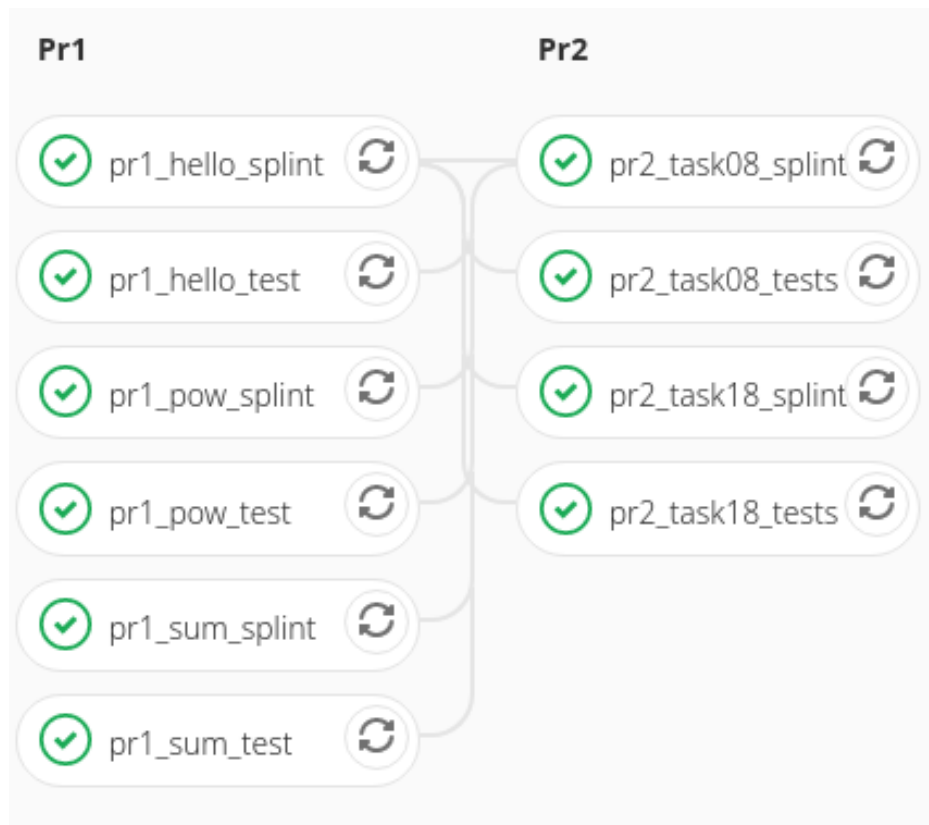


Рисунок 2.6 – Пройденные пайплайны

3 Заключение

В результате выполнения практической работы был изучен git и основные методы работы с ним, также были написаны 2 программы на языке программирования C, по теме однопроходные алгоритмы.