Задача 10. Разработать алгоритм, реализующий утилиту “grep o f1 f2 … fn”.
Параллельность необходимо заводить по чтению и поиску соответствия, т.е. следуя модели буфера, соединяющего производителей данных с потребителями данных.

Исследование провёл студент группы 22207 Гордеев Никита

Дата выполнения работы 25.12.2022 (Вариант 2)

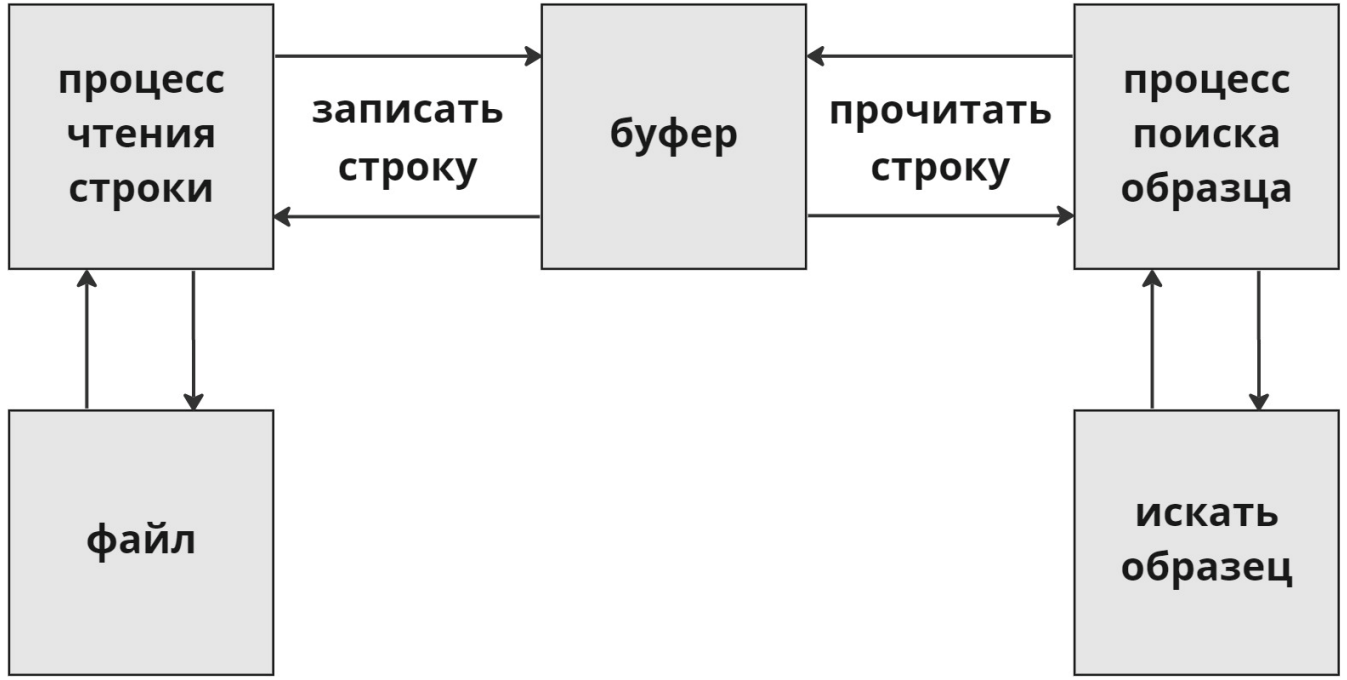
Задачи

- **Разработать алгоритм, реализующий утилиту “grep o f1 f2 … fn”, где:**
  - **o - шаблон для соответствия строки,**
  - **f1 f2 … fn - текстовые файлы для поиска соответствующих шаблону строк.**

- **Параллельность необходимо заводить по чтению и поиску соответствия, т.е. следуя модели буфера, соединяющего производителей данных с потребителями данных.**

Идея параллельности

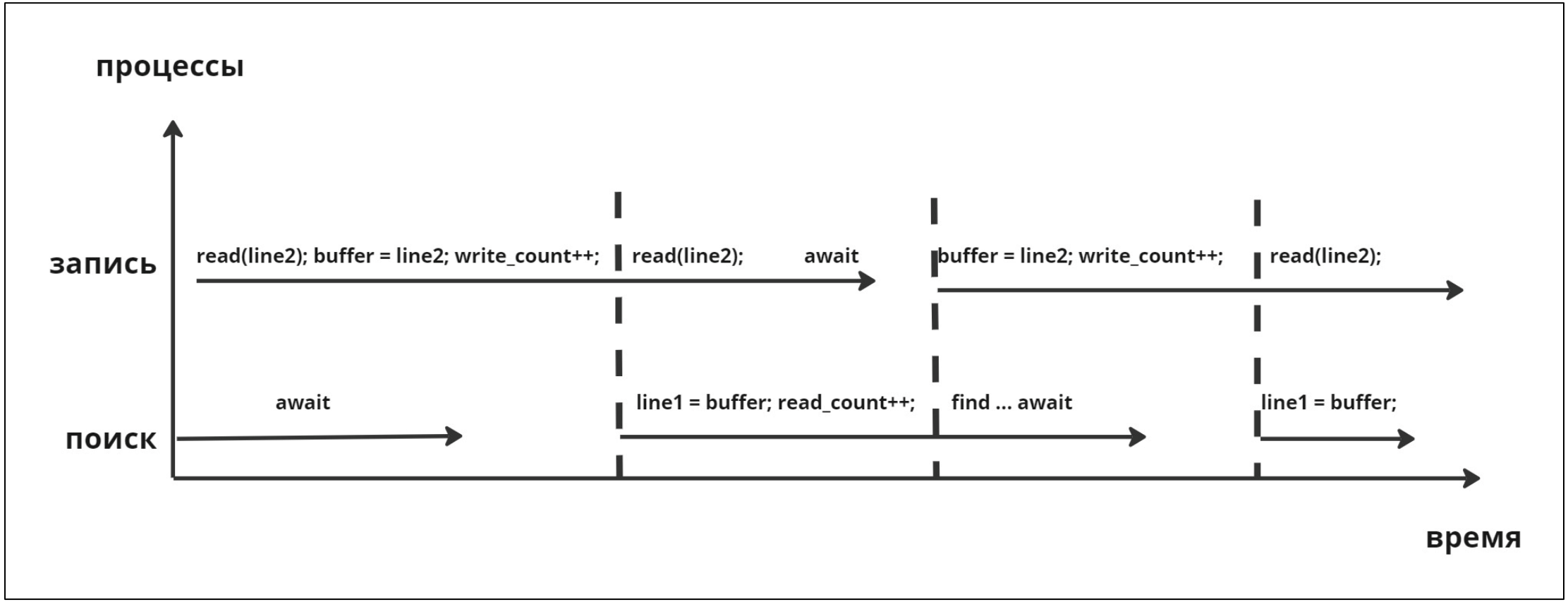
- **Сделать два параллельных процесса: процесс чтения строки и процесс поиска образца(соответствия).**



Параллельная программа

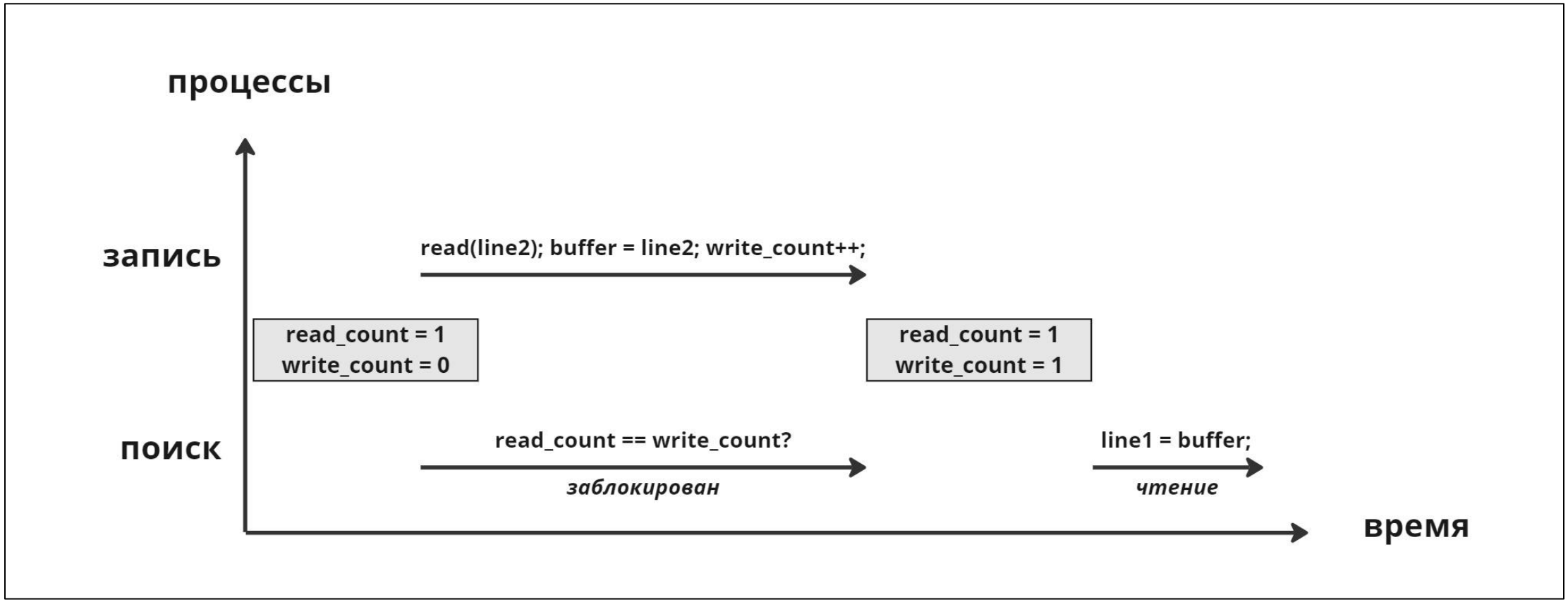
```
string buffer; // буфер для хранения прочитанной строки
int read_count = 1; // счётчик прочитанных строк
int write_count = 0; // счётчик записанных строк
bool done = false; // переменная, сигнализирующая о конце файла

co
    // процесс поиска шаблона
    string line1; // строка, в которой ведется поиск
    while (1) {
        <await (read _count == write_count OR done);> //кол-во прочит. равно кол-ву написанных, или конец файла
        if (done) break; // если конец файла, то завершаем процесс
        line1 = buffer; // читаем строку из буфера
        read_count++; // увеличиваем счётчик прочитанных строк
        find(pattern, line); // поиск шаблона
        if (found) print(line); // вывод подходящих строк
    }

    // процесс чтения строк
    string line2; // строка, в которую ведется запись
    while (1) {
        read(line2) // читаем строку
        if (EOF) { // если конец файла, завершаем процесс
            done = true;
            break;
        }

        <await (read_count > write_count);> // ждем, кол-во прочит. строк станет больше кол-ва написанных
        buffer = line2; // помещаем прочитанную строку в буфер
        write_count++; // увеличиваем счётчик прочитанных строк
    }
oc;
```

Схема работы



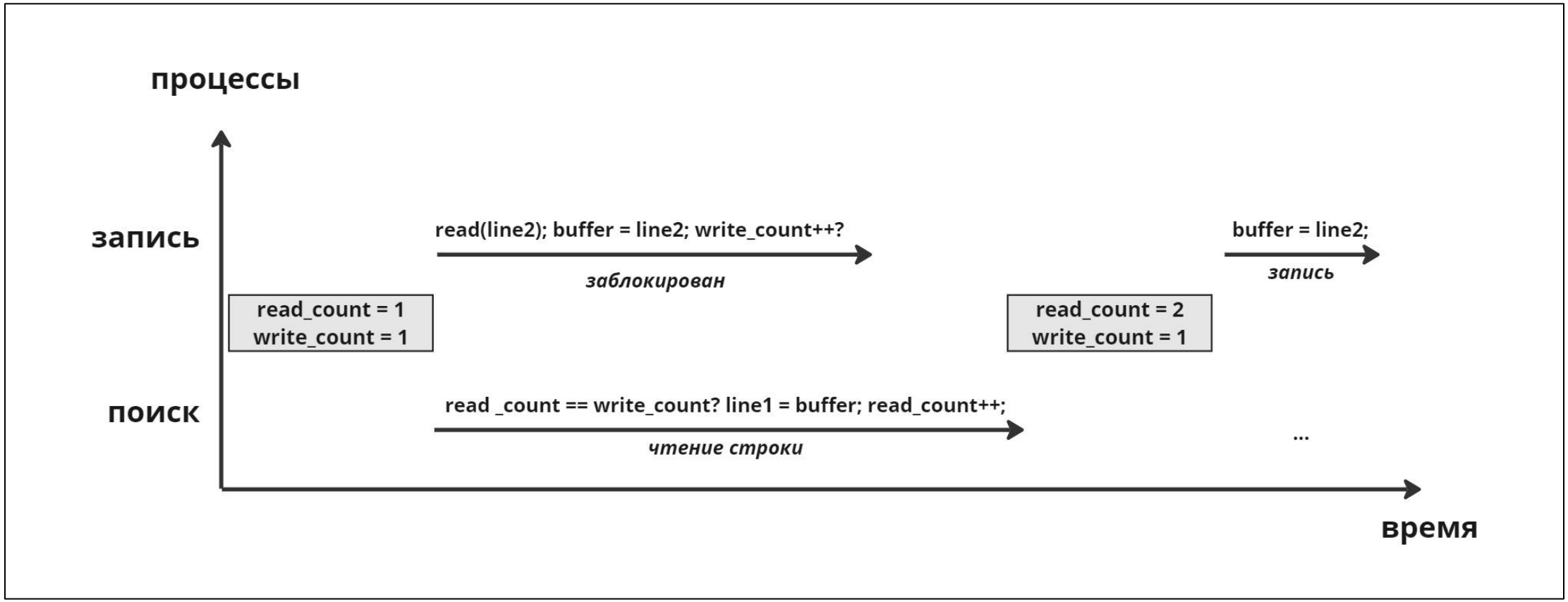
Возможные коллизии

1) Если буфер пуст, и процесс поиска образца пытается читать строку.



Возможные коллизии

2) Если буфер не прочитан, и процесс записи пытается записать новую строку



Выводы

- **Разработан алгоритм, реализующий утилиту “grep o f1 f2 … fn”;**
- **Параллельность заводится по чтению и поиску соответствия;**
- **Программа работает без коллизий.**

Материалы:

- **Задачи поиска по образцу // Национальная библиотека им. Н. Э. Баумана Bauman National Library URL: https://ru.bmstu.wiki/Задачи_поиска_по_образцу (дата обращения: 18.12.2022).**
- **Пример: параллельный поиск указанного текстового шаблона // ВикиЧтение Системное программирование в среде Windows Харт Джонсон М URL: https://it.wikireading.ru/1365 (дата обращения: 18.12.2022).**
- **Реализация простого механизма регулярных выражений в 70 строк кода // Хабр URL: https://habr.com/ru/post/581336/ (дата обращения: 18.12.2022).**

Изменения

- **Версия 2**
  - **Добавил комментарии в код**
  - **Изменил схемы**