

Национальный исследовательский университет «МЭИ»
Институт информационных и вычислительных технологий
Кафедра Вычислительных технологий

Отчет по лабораторной работе № 4
по курсу «Операционные системы»

Группа: А-09-22

Выполнили:

Лойко С.И. (аналитик)
Рябчиков А.Д. (менеджер)
Зобков О.В. (разработчик)
Кошеваров А.Д. (тестировщик)

Проверил:

Емельянов Д.М.

Москва

2025

Оглавление

1. ПОСТАНОВКА ЗАДАЧИ.....	3
2. УТОЧНЕННОЕ ЗАДАНИЕ.....	4
3. ОПИСАНИЕ АЛГОРИТМА ПРОГРАММЫ.....	5
4. КОД ПРОГРАММЫ.....	7
5. РАСПРЕДЕЛЕНИЕ РОЛЕЙ В КОМАНДЕ	23

1. ПОСТАНОВКА ЗАДАЧИ

Задание:

Лабораторная работа №4 выполняется всей бригадой и требует разработки одного или нескольких bash/shell скриптов.

Для этого участникам бригады необходимо распределить между собой роли менеджера, аналитика, разработчика и тестировщика.

Преподаватель (далее - заказчик) выдвигает следующие функциональные требования к разрабатываемому _Решению_ под ОС Linux.

1. В качестве предметной области и исходных данных выступает Файловая Система Преподавателя из Лабораторной работы №3.

2. Базовое решение для успешной сдачи лабораторной работы должно предоставлять следующие возможности по заданному пользователем номеру группы:

- * вывод имени студента с наилучшей посещаемостью и количество посещенных им занятий;
- * вывод списка студентов, не сдавших заданный тест;

3. Для получения Отличной оценки (и потенциально дополнительных баллов) также необходимо в рамках решения реализовать:

- * вывод списка студентов, пропустивших N-ую лекцию, но сдавших K-ый тест на 5;
- * вывод студентов с наилучшими результатами (по количеству правильных ответов) за каждый год сдачи теста;

Требования могут уточняться менеджером или аналитиком бригады лично с заказчиком.

В качестве отчетности, кроме демонстрации заказчику работоспособных скриптов, необходимо предоставить краткую документацию решения (в электронном виде), отчет о проведенном тестировании (в электронном виде), а также документировать основные моменты разработки.

2. УТОЧНЕННОЕ ЗАДАНИЕ

Базовое решение:

2.2) нужны только те, кто в итоге не пересдал или вообще не сдавал.

Решение на оценку «Отлично»:

3.1) Можно учитывать тех, кто пересдал.

3.2) При выводе нужны год, ФИО, группа, количество правильных ответов.

3. ОПИСАНИЕ АЛГОРИТМА ПРОГРАММЫ

В нашем случае программа разделена на пять скриптов, один из которых (Start.sh) содержит основное меню, запускающее остальные.

1) Start.sh - главный управляющий скрипт (меню).

Алгоритм работы:

1. Определяет SCRIPT_DIR - папку, где лежит сам Start.sh.

2. Показывает основное меню из 4 пунктов.

3. Ждёт ввода пользователя (0–4).

4. Делает проверку:

* если пустая строка → ошибка;

* если не число 0–4 → ошибка.

5. В зависимости от выбора:

* запускает bash p2-1.sh

* запускает bash p2-2.sh

* запускает bash p3-1.sh

* запускает bash p3-2.sh

6. Если выбранный файл отсутствует — сообщает ошибку.

7. Цикл продолжается бесконечно

1) p2-1.sh - Лучшие по посещаемости

Алгоритм:

1. Определяет BASE_DIR - каталог labfiles-25.

2. Показывает меню выбора предмета (Поп-Культурология / Цирковое дело).

3. Проверяет, существует ли папка предмета.

4. Далее - цикл выбора группы:

* ожидает название группы;

* проверяет, существует ли файл:

5. После выбора группы:

* через awk проходится по файлу посещаемости.

6. awk:

* считает количество единиц "1" в строке;

* определяет максимальное количество посещений;

* выводит всех, у кого посещений столько же, сколько максимум.

Вывод: список студентов с наилучшей посещаемостью.

2) p2-2.sh - Студенты, не пересдавшие тест

Алгоритм:

1. Выбор предмета.

2. Проверка папки предмета и каталога tests/.

3. Выбор группы:

* проверяет, существует ли хоть в одном TEST-* запись для этой группы.

4. Выбор номера теста.

5. Проверка существования файла:

TEST-K

6. awk анализирует строки и для каждой строки:

* берёт оценку из 5-го поля;

* извлекает только цифру

* если оценки нет - считает "нет оценки".

8. В конце выводит:

* тех, у кого лучшая оценка = 2

* тех, у кого оценки нет совсем

3) p3-1.sh - Пропустили лекцию N и сдали тест K на 5

Алгоритм:

1. Выбор предмета.

2. Выбор группы + проверка файла посещаемости.

3. Определение количества лекций.

4. Внутренний цикл:

* ввод номера лекции N;

* ввод номера теста K.

5. Через awk происходит совместная обработка:

* первый файл - посещаемость

→ создается таблица missed для тех, кто пропустил N-ю лекцию

* второй файл - тест TEST-K

→ создается массив best_grade

6. После обработки:

* выводятся только те студенты, кто:

* есть в missed

* и best_grade == 5

4) p3-2.sh - Лучшие результаты по годам

Алгоритм:

1. Выбор предмета.

2. Определение папки тестов.

3. Выбор номера теста.

4. Чтение файла.

5. В awk:

* извлекается год = первые 4 символа из поля DATE.

* создается массив всех годов.

* для каждого года определяется максимальное количество correct.

* на каждого студента создается ключ год|группа|логин и сохраняется его лучший результат.

4. КОД ПРОГРАММЫ

Start.sh:

```
SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"

while true; do
    echo
    echo "===== Лабораторная работа №4 ====="
    echo "1) Пункт 1: лучший(ие) по посещаемости"
    echo "2) Пункт 2: не пересдавшие тест (оценка 2 или без оценки)"
    echo "3) Пункт 3: пропустили лекцию N, но сдали тест K на 5"
    echo "4) Пункт 4: лучшие результаты по годам сдачи теста"
    echo "0) Выход"
    read -rp "Ваш выбор (0-4): " choice

    if [ -z "$choice" ]; then
        echo "Ошибка: выбор не может быть пустым."
        continue
    fi

    case "$choice" in
        1)
            if [ -f "$SCRIPT_DIR/p2-1.sh" ]; then
                bash "$SCRIPT_DIR/p2-1.sh"
            else
                echo "Ошибка: не найден $SCRIPT_DIR/p2-1.sh"
            fi
            ;;
        2)
            if [ -f "$SCRIPT_DIR/p2-2.sh" ]; then
                bash "$SCRIPT_DIR/p2-2.sh"
            else
                echo "Ошибка: не найден $SCRIPT_DIR/p2-2.sh"
            fi
            ;;
        3)
            if [ -f "$SCRIPT_DIR/p3-1.sh" ]; then
                bash "$SCRIPT_DIR/p3-1.sh"
            else
                echo "Ошибка: не найден $SCRIPT_DIR/p3-1.sh"
            fi
            ;;
        4)
            if [ -f "$SCRIPT_DIR/p3-2.sh" ]; then
                bash "$SCRIPT_DIR/p3-2.sh"
            else
                echo "Ошибка: не найден $SCRIPT_DIR/p3-2.sh"
            fi
    esac
done
```

```

        ;;
0)
    echo "Выход из меню."
    exit 0
    ;;
*)
    echo "Некорректный выбор. Введите число от 0 до 4."
    ;;
esac
done

```

p2-1.sh:

```

SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
BASE_DIR="$SCRIPT_DIR/../labfiles-25"

while true; do
    echo
    echo "===== ВЫБОР ПРЕДМЕТА ====="
    echo "1) Поп-Культурология"
    echo "2) Цирковое_Дело"
    echo "0) Выход"
    read -rp "Ваш выбор (1-2, 0): " subj_choice

    if [ -z "$subj_choice" ]; then
        echo "Ошибка: выбор не может быть пустым. Введите 1, 2 или 0."
        continue
    fi

    case "$subj_choice" in
        0)
            echo "Выход из скрипта."
            exit 0
            ;;
        1)
            SUBJECT="Поп-Культурология"
            echo "Выбран предмет: $SUBJECT"
            ;;
        2)
            SUBJECT="Цирковое_Дело"
            echo "Выбран предмет: $SUBJECT"
            ;;
        *)
            echo "Некорректный выбор. Введите 1, 2 или 0."
            continue
            ;;
    esac

    SUBJECT_DIR="$BASE_DIR/$SUBJECT"

```

```

if [ ! -d "$SUBJECT_DIR" ]; then
    echo "Ошибка: не найден каталог предмета: $SUBJECT_DIR" >&2

    continue
fi

while true; do
    echo
    echo "Предмет: $SUBJECT"
    read -rp "Введите группу (например A-06-04/Аe-21-22, 0 –
сменить предмет/выход; Ввод на англ-ом): " GROUP

    if [ "$GROUP" = "0" ]; then
        echo "Возврат к выбору предмета..."
        break      # выходим из цикла групп, снова будет выбор
предмета
    fi

    if [ -z "$GROUP" ]; then
        echo "Группа не может быть пустой."
        continue
    fi

    ATT_FILE="$SUBJECT_DIR/${GROUP}-attendance"

    if [ ! -f "$ATT_FILE" ]; then
        echo "Ошибка: не найден файл посещаемости для группы
$GROUP:"
        echo "          $ATT_FILE"
        echo "Попробуйте ввести другую группу или 0 для смены
предмета."
        continue      # остаёмся в цикле групп
    fi

    echo
    echo "==== Предмет: $SUBJECT, группа: $GROUP ==="
    echo "Поиск студентов с лучшей посещаемостью..."

    awk '
{
    login = $1
    bits  = $2
    cnt   = gsub(/1/, "", bits)    # количество посещённых
занятий
    count[login] = cnt
    if (cnt > best) {

```

```

        best = cnt
    }
}
END {
    if (length(count) == 0) {
        print "Нет данных по посещаемости."
        exit
    }
    print "Лучшие по посещаемости (занятий: " best "):"
    for (l in count) {
        if (count[l] == best) {
            print " " l
        }
    }
}
' "$ATT_FILE"

echo
echo "Можешь ввести ещё одну группу для этого предмета,"
echo "или 0 – чтобы выбрать другой предмет / выйти."
done
done

```

p2-2.sh:

```

SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
BASE_DIR="$SCRIPT_DIR/../labfiles-25"

while true; do
    echo
    echo "===== ВЫБОР ПРЕДМЕТА ====="
    echo "1) Поп-Культурология"
    echo "2) Цирковое_Дело"
    echo "0) Выход"
    read -rp "Ваш выбор (1-2, 0): " subj_choice

    if [ -z "$subj_choice" ]; then
        echo "Ошибка: выбор не может быть пустым. Введите 1, 2 или 0."
        continue
    fi

    case "$subj_choice" in
    0)
        echo "Выход из скрипта."
        exit 0
        ;;
    1)
        SUBJECT="Поп-Культурология"
        echo "Выбран предмет: $SUBJECT"

```

```

;;
2)
SUBJECT="Цирковое_Дело"
echo "Выбран предмет: $SUBJECT"
;;
*)
echo "Некорректный выбор. Введите 1, 2 или 0."
continue
;;
esac

SUBJECT_DIR="$BASE_DIR/$SUBJECT"
TESTS_DIR="$SUBJECT_DIR/tests"

if [ ! -d "$SUBJECT_DIR" ]; then
    echo "Ошибка: не найден каталог предмета: $SUBJECT_DIR" >&2
    continue
fi
if [ ! -d "$TESTS_DIR" ]; then
    echo "Ошибка: не найден каталог с тестами: $TESTS_DIR" >&2
    continue
fi

# ----- ЦИКЛ ВЫБОРА ГРУППЫ -----
while true; do
    echo
    echo "Предмет: $SUBJECT"
    read -rp "Введите группу (например A-06-04/Ae-21-22, 0 –
сменить предмет/выход; Ввод на англ-ом): " GROUP

    # 0 – вернуться к выбору предмета
    if [ "$GROUP" = "0" ]; then
        echo "Возврат к выбору предмета..."
        break
    fi

    if [ -z "$GROUP" ]; then
        echo "Группа не может быть пустой."
        continue
    fi

    # для сравнения внутри awk делаем верхний регистр
    GROUP_UPPER=$(echo "$GROUP" | tr '[:lower:]' '[:upper:]')

    # --- ПРОВЕРКА КОРРЕКТНОСТИ ВВОДА ГРУППЫ ---
    # ищем, есть ли такая группа хотя бы в одном файле TEST-*
    if ! awk -v grp="$GROUP_UPPER" -F';' '
        BEGIN { grp_upper = toupper(grp); found = 0 }

```

```

{
    # первое поле – группа
    if (toupper($1) == grp_upper) { found = 1; exit }
}
END { exit (found ? 0 : 1) }
' "$TESTS_DIR"/TEST-* 2>/dev/null; then
echo "Ошибка: группа $GROUP_UPPER не найдена в файлах
тестов предмета."
echo "Проверьте правильность ввода и попробуйте ещё раз."
continue
fi

while true; do
    echo
    echo "Предмет: $SUBJECT, группа: $GROUP_UPPER"
    echo "0 – выбрать другую группу"
    read -rp "Введите номер теста: " TEST_NO

    if [ "$TEST_NO" = "0" ]; then
        echo "Возврат к выбору группы..."
        break
    fi

    if [ -z "$TEST_NO" ]; then
        echo "Номер теста не может быть пустым."
        continue
    fi

    if ! [[ "$TEST_NO" =~ ^[0-9]+$ ]]; then
        echo "Номер теста должен быть целым числом."
        continue
    fi

    TEST_FILE="$TESTS_DIR/TEST-$TEST_NO"

    if [ ! -f "$TEST_FILE" ]; then
        echo "Ошибка: файл теста TEST-$TEST_NO не найден:"
        echo "          $TEST_FILE"
        echo "Попробуйте другой номер теста или 0 для возврата
к выбору группы."
        continue
    fi

    echo
    echo "==== Предмет: $SUBJECT, группа: $GROUP_UPPER, тест:
$TEST_NO ==="

```

```

echo "Поиск студентов, которые написали тест только на 2
или вообще без оценки..."

awk -v grp="$GROUP_UPPER" '
BEGIN {
    grp_upper = toupper(grp)
}
{
    # Разбиваем строку по ;
    split($0, f, ";")
    if (length(f) < 5) next

    # Фильтруем строки по группе без учёта регистра
    if (toupper(f[1]) != grp_upper) next

    login = f[2]

    # Сырой текст оценки (может быть пустой, "2", "2-", "5+", "4--" и т.п.)
    raw = f[5]

    # Если столбец с оценкой пустой (или только пробелы) –
    # считаем, что у этого студента пока НЕТ оценки
    tmp = raw
    gsub(/[:space:]/, "", tmp)
    if (tmp == "") {
        no_grade[login] = 1
        any[login] = 1
        next
    }

    # Выдираем цифры из поля оценки
    grade_str = raw
    gsub(/\^0-9/, "", grade_str)

    # Если цифр нет (какая-то странная запись) – просто
    считаем,
    # что студент "есть", но оценки не учитываем
    if (grade_str == "") {
        any[login] = 1
        next
    }

    grade = grade_str + 0

    # Запоминаем ЛУЧШУЮ оценку по этому тесту для каждого
    студента
}
'
```

```

        if (!(login in best) || grade > best[login]) {
            best[login] = grade
        }
        any[login] = 1
    }
END {
    found = 0
    print "Студенты группы " grp " с лучшей оценкой 2 или
без оценки по этому тесту:"
    for (login in any) {
        # вариант 1: есть числовая оценка, и лучшая = 2
        if ((login in best) && best[login] == 2) {
            print " " login
            found = 1
        }
        # вариант 2: ни одной числовой оценки, хотя записи
        есть (пустой 5-й столбец)
        else if (!(login in best)) {
            print " " login
            found = 1
        }
    }
    if (!found) {
        print "Таких студентов не найдено"
    }
}
' "$TEST_FILE"

```

```

echo
echo "Можешь ввести другой номер теста для этой группы,"
echo "или 0 – чтобы выбрать другую группу."
done
done
done

```

p3-1.sh:

```

SCRIPT_DIR=$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)
BASE_DIR="$SCRIPT_DIR/../labfiles-25"

while true; do
echo
echo "===== ВЫБОР ПРЕДМЕТА ====="
echo "1) Поп-Культурология"
echo "2) Цирковое Дело"
echo "0) Выход"
read -rp "Ваш выбор (1-2, 0): " subj_choice

```

```

if [ -z "$subj_choice" ]; then
    echo "Ошибка: выбор не может быть пустым. Введите 1, 2 или 0."
    continue
fi

case "$subj_choice" in
0)
    echo "Выход из скрипта."
    exit 0
    ;;
1)
    SUBJECT="Поп-Культурология"
    echo "Выбран предмет: $SUBJECT"
    ;;
2)
    SUBJECT="Цирковое_Дело"
    echo "Выбран предмет: $SUBJECT"
    ;;
*)
    echo "Некорректный выбор. Введите 1, 2 или 0."
    continue
    ;;
esac

SUBJECT_DIR="$BASE_DIR/$SUBJECT"
TESTS_DIR="$SUBJECT_DIR/tests"

if [ ! -d "$SUBJECT_DIR" ]; then
    echo "Ошибка: не найден каталог предмета: $SUBJECT_DIR" >&2
    continue
fi
if [ ! -d "$TESTS_DIR" ]; then
    echo "Ошибка: не найден каталог с тестами: $TESTS_DIR" >&2
    continue
fi

while true; do
    echo
    echo "Предмет: $SUBJECT"
    read -rp "Введите группу (например А-06-04/Ае-21-22, 0 – сменить предмет/выход; Ввод на англ-ом): " GROUP

    if [ "$GROUP" = "0" ]; then
        echo "Возврат к выбору предмета..."
        break
    fi

```

```

if [ -z "$GROUP" ]; then
    echo "Группа не может быть пустой."
    continue
fi

GROUP_UPPER=$(echo "$GROUP" | tr '[:lower:]' '[:upper:]')

ATT_FILE="$SUBJECT_DIR/${GROUP_UPPER}-attendance"

if [ ! -f "$ATT_FILE" ]; then
    echo "Ошибка: не найден файл посещаемости для группы
$GROUP_UPPER:"
    echo "        $ATT_FILE"
    echo "Проверьте название группы и попробуйте ещё раз."
    continue
fi

# Узнаём количество лекций (длина строки из 0/1)
MAX_LECTURES=$(awk 'NR==1 { print length($2); exit }'
"$ATT_FILE")
if [ -z "$MAX_LECTURES" ]; then
    echo "Ошибка: файл посещаемости пуст или имеет неверный
формат."
    continue
fi

echo "Для группы $GROUP_UPPER найдено занятий: $MAX_LECTURES"

while true; do
    echo
    echo "Предмет: $SUBJECT, группа: $GROUP_UPPER"
    echo "0 – выбрать другую группу"
    read -rp "Введите номер лекции N (1..$MAX_LECTURES): "
    LEC_NO

    if [ "$LEC_NO" = "0" ]; then
        echo "Возврат к выбору группы..."
        break
    fi

    if [ -z "$LEC_NO" ]; then
        echo "Номер лекции не может быть пустым."
        continue
    fi

    if ! [[ "$LEC_NO" =~ ^[0-9]+$ ]]; then

```

```

        echo "Номер лекции должен быть целым числом."
        continue
    fi

    if [ "$LEC_NO" -lt 1 ] || [ "$LEC_NO" -gt "$MAX_LECTURES"
]; then
        echo "Номер лекции должен быть от 1 до $MAX_LECTURES."
        continue
    fi

    read -rp "Введите номер теста K (0 – выбрать другую
группу): " TEST_NO

    if [ "$TEST_NO" = "0" ]; then
        echo "Возврат к выбору группы..."
        break
    fi

    if [ -z "$TEST_NO" ]; then
        echo "Номер теста не может быть пустым."
        continue
    fi

    if ! [[ "$TEST_NO" =~ ^[0-9]+$ ]]; then
        echo "Номер теста должен быть целым числом."
        continue
    fi

    TEST_FILE="$TESTS_DIR/TEST-$TEST_NO"

    if [ ! -f "$TEST_FILE" ]; then
        echo "Ошибка: файл теста TEST-$TEST_NO не найден:"
        echo "          $TEST_FILE"
        echo "Попробуйте другой номер теста или 0 для возврата
к выбору группы."
        continue
    fi

    echo
    echo "==== Предмет: $SUBJECT, группа: $GROUP_UPPER, лекция
N=$LEC_NO, тест K=$TEST_NO ==="
    echo "Поиск студентов, пропустивших лекцию и сдавших тест
на 5..."


awk -v grp="$GROUP_UPPER" -v lec="$LEC_NO" -v
testno="$TEST_NO" '
BEGIN {

```

```

        grp_upper = toupper(grp)
    }
    NR==FNR {
        # Первый файл – посещаемость: отмечаем, кто ПРОПУСТИЛ
лекцию lec
        login = $1
        bits  = $2
        if (lec <= length(bits)) {
            b = substr(bits, lec, 1)
            if (b == "0") {
                missed[login] = 1
            }
        }
        next
    }
{
    # Второй файл – TEST-K
    split($0, f, ";")
    if (length(f) < 5) next

    # Группа должна совпадать (без учёта регистра)
    if (toupper(f[1]) != grp_upper) next

    login = f[2]

    # Оценка может быть "5", "5+", "5--" и т.п.
    grade_str = f[5]
    gsub(/[^0-9]/, "", grade_str)
    if (grade_str == "") next
    grade = grade_str + 0

    # Запоминаем ЛУЧШУЮ оценку по этому тесту
    if (!(login in best_grade) || grade >
best_grade[login]) {
        best_grade[login] = grade
    }
    seen[login] = 1
}
END {
    found = 0
    print "Студенты группы " grp " пропустившие лекцию "
lec " и сдавшие тест " testno " на 5:"
    for (login in missed) {
        if (login in best_grade && best_grade[login] == 5)
{
            print " " login
            found = 1
        }
}

```

```

        }
        if (!found) {
            print "Таких студентов не найдено"
        }
    }
' "$ATT_FILE" "$TEST_FILE"

echo
echo "Можешь ввести другие N и K для этой же группы,"
echo "или 0 при вводе N/K – чтобы выбрать другую группу."
done
done
done

```

p3-2.sh:

```

SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
BASE_DIR="$SCRIPT_DIR/../labfiles-25"

while true; do
    echo
    echo "===== ВЫБОР ПРЕДМЕТА ====="
    echo "1) Поп-Культурология"
    echo "2) Цирковое_Дело"
    echo "0) Выход"
    read -rp "Ваш выбор (1-2, 0): " subj_choice

    if [ -z "$subj_choice" ]; then
        echo "Ошибка: выбор не может быть пустым. Введите 1, 2 или 0."
        continue
    fi

    case "$subj_choice" in
        0)
            echo "Выход."
            exit 0
            ;;
        1)
            SUBJECT="Поп-Культурология"
            echo "Выбран предмет: $SUBJECT"
            ;;
        2)
            SUBJECT="Цирковое_Дело"
            echo "Выбран предмет: $SUBJECT"
            ;;
        *)
            echo "Некорректный выбор. Введите 1, 2 или 0."
            continue
            ;;
    esac
done

```

```

esac

SUBJECT_DIR="$BASE_DIR/$SUBJECT"
TESTS_DIR="$SUBJECT_DIR/tests"

if [ ! -d "$SUBJECT_DIR" ]; then
    echo "Ошибка: не найден каталог предмета: $SUBJECT_DIR" >&2
    continue
fi
if [ ! -d "$TESTS_DIR" ]; then
    echo "Ошибка: не найден каталог с тестами: $TESTS_DIR" >&2
    continue
fi

while true; do
    echo
    echo "Предмет: $SUBJECT"
    echo "0 – сменить предмет"
    read -rp "Введите номер теста (например 1,2,...): " TEST_NO

    if [ "$TEST_NO" = "0" ]; then
        echo "Возврат к выбору предмета..."
        break
    fi

    if [ -z "$TEST_NO" ]; then
        echo "Номер теста не может быть пустым."
        continue
    fi

    if ! [[ "$TEST_NO" =~ ^[0-9]+$ ]]; then
        echo "Номер теста должен быть целым числом."
        continue
    fi

    TEST_FILE="$TESTS_DIR/TEST-$TEST_NO"

    if [ ! -f "$TEST_FILE" ]; then
        echo "Ошибка: файл теста TEST-$TEST_NO не найден:"
        echo "          $TEST_FILE"
        echo "Попробуйте другой номер теста или 0 для возврата к"
        echo "выбору предмета."
        continue
    fi

    echo
    echo "==== Предмет: $SUBJECT, тест: TEST-$TEST_NO ==="

```

```

echo "Поиск студентов с лучшими результатами по годам (по
количество правильных ответов)...""

awk '
BEGIN {
    FS = ";"
}
{
    if (NF < 4) next

    group    = $1      # группа
    login    = $2      # ФИО/логин
    date     = $3      # дата
    correct  = $4 + 0  # количество правильных ответов

    # Берём год как первые 4 символа даты (подходит для "2007-
09-21" и "2007 September")
    year = substr(date, 1, 4)
    if (year !~ /^[0-9][0-9][0-9][0-9]$/) next

    key = year "|" group "|" login

    # максимальное число правильных ответов по этому году
    if ( !(year in best_year) || correct > best_year[year] ) {
        best_year[year] = correct
    }

    # лучший результат конкретного студента (группа+логин) в
этом году
    if ( !(key in best_student) || correct > best_student[key] )
    {
        best_student[key] = correct
    }

    years[year] = 1
}
END {
    have = 0
    for (y in years) { have = 1; break }

    if (!have) {
        print "Нет данных по этому тесту для выбранного
предмета."
        exit
    }

    print "Лучшие результаты по годам сдачи теста:"
}
'

```

```

        for (y in years) {
            maxc = best_year[y]
            print "Год " y ", максимальное количество правильных
ответов: " maxc
            print " Студенты (группа, ФИО):"
            for (k in best_student) {
                split(k, a, "|")
                year_k = a[1]
                group_k = a[2]
                login_k = a[3]
                if (year_k == y && best_student[k] == maxc) {
                    # раньше тут было ещё количество баллов,
                    теперь только группа + ФИО
                    print "    " group_k ", " login_k
                }
            }
        }
        '$TEST_FILE'

echo
echo "Можешь ввести другой номер теста для этого предмета,"
echo "или 0 – чтобы выбрать другой предмет."
done
done

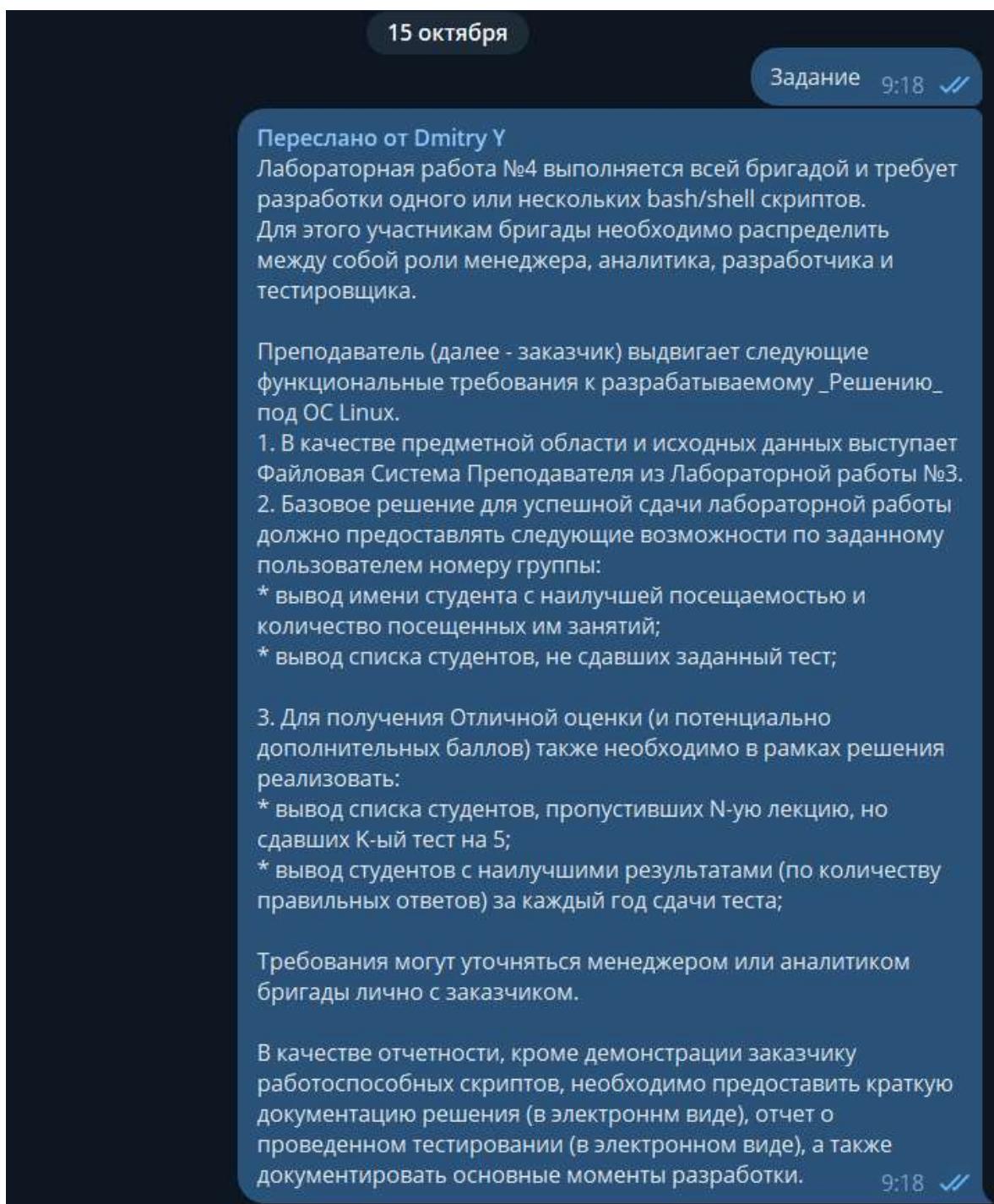
```

5. РАСПРЕДЕЛЕНИЕ РОЛЕЙ В КОМАНДЕ

Для выполнения проекта было выбрано следующее распределение ролей:

Аналитик – Лойко С.И.
Менеджер – Рябчиков А.Д.
Разработчик – Зобков О.В.
Тестирующий – Кошеваров А.Д.

Процесс разработки запечатлён на скриншотах ниже.



13 ноября

короче план действий такой:

Аналитик (@g0set) смотрит на задачу обрабатывает её если что не так пишет емельянову и передает задачу разрабу (@d1sappo1ntment)

Тот собственно разрабатывают и отправляет тестировщику (@drewshich)

Тот в свою очередь проверяет если все не ок то отправляет разрабу а если все ок то кидает менеджеру (Я)

Тот собирает отчет и все дела

14 ноября

Давайте завтра звонок устроим и сделаем



13:21 ✓

15 ноября

Олег

1. Номер группы вводится динамически, то есть по желанию пользователя, или статически должен быть прописан в скрипте?

2. Нужно ли реализовать подпункты пункта 2 в одном или

разных скриптах?

Если в одном то нужно ли реализовывать диалог внутри скрипта с выбором возможностей(отдельно вывод студента с наилучшей посещаемостью и отдельно вывод студентов не сдавших тест или выводить все сразу)?

3. Нужно ли реализовать подпункты пункта 3 в одном или

разных скриптах?

Если в одном то нужно ли реализовывать диалог внутри скрипта с выбором возможностей(отдельно вывод списка студентов пропустивших N-ую лекцию, но сдавших K-ый тест на 5 и вывод студентов с наилучшими результатами (по количеству правильных ответов) за каждый год сдачи теста или выводить все сразу)?

4. Необходимо ли обрабатывать исключения при неправильном вводе данных(номер группы, номер лекции, номер теста)?

изменено 18:35



Степан Лойко

Переслано от Dmitry Y

1. Как-то задается пользователем, например, параметром, вводом или в конфиг файле

20:03

Переслано от Dmitry Y

2. Можно в одном скрипте, но нужно, чтобы вывод был отдельным/независимым для каждого пункта, все вместе не годится

20:03

Переслано от Dmitry Y

3. аналогично, можно вообще все пункты в одном скрипте

20:03



Переслано от Dmitry Y

4. да, необходимо

20:03



Олег



p2-1.sh

2.6 KB

20:37



Андрей Кошеваров

Ответить

Добавь возможность выбора предметов в одной сессии

20:43

изменено 20:47

Олег



p2-1.sh

3.4 KB

изменено 20:59

Пункт 2.2

Человек сначала получил 2, потом переписал

Вопрос: должен ли человек, переписавший тест, попасть в вывод?

A-06-04;PashkovskyA;2007 September;1;2

A-06-04;PashkovskyA;2007 September;3;3

A-06-07;PodshivalovaBZ;2010 September;1;2

A-06-07;PodshivalovaBZ;2010 September;1;2

A-06-07;PodshivalovaBZ;2010 September;0;2

A-06-07;PodshivalovaBZ;2010 September;0;2

A-06-07;PodshivalovaBZ;2010 September;1;2

A-06-07;PodshivalovaBZ;2010 September;0;2

A-06-07;PodshivalovaBZ;2010 September;0;2

A-06-07;PodshivalovaBZ;2010 September;3;3

изменено 21:13

Пункт 3.1

В список студентов, пропустивших N-ую лекцию, но сдавших К-ый тест на 5, должен попасть тот, кто сдал с 1-го раза или переписал на 5?

21:27

Степан Лойко

Передал вопрос 21:29

Олег

Пункт 3.2

Должны ли мы указывать год или выводить все годы сразу? Что должно быть в выводе только ФИО или ФИО + № группы, кол-во баллов, оценку?

21:33

Степан Лойко

Ждем ответа 21:35

Степан Лойко

Переслано от Dmitry Y

нет, нужны только те, кто в итоге не пересдал или вообще не сдавал

23:22

Степан Лойко

Переслано от Dmitry Y

3.1. Можно брать тех, кто пересдал

3.2. Нужны год, ФИО, группа, количество правильных ответов

23:41

16 ноября

Степан Лойко

Переслано от Dmitry Y

Добрый! Вы можете добавить нового студента в файл группы, и у него не будет тестов

Ответить

19:06

-o- Commits on Nov 17, 2025

Загрузка тестов



g0set committed 6 hours ago

-o- Commits on Nov 16, 2025

Провел тестирование программ



drewshich committed yesterday

Добавил файл для запуска скриптов



OlegA0922 committed yesterday

Закончил все задачи



OlegA0922 committed yesterday

Обновил данные в файлах для тестирования



drewshich committed yesterday

Сделал первые 2 задания



OlegA0922 committed yesterday

Initial commit



texnik400 authored yesterday