# Задание VI. Обработка последовательной файловой структуры на языке Си

Разработать последовательную структуру данных для представления простейшей базы данных на файлах в СП Си в соответствии с заданным вариантом. Составить программу генерации внешнего нетекстового файла заданной структуры, содержащего представительный набор записей (15-20). Распечатать содержимое сгенерированного файла в виде таблицы и выполнить над ним заданное действие для 2–3 значений параметров запроса ***p*** и распечатать результат.

Действие по выборке данных из файла оформить в виде *отдельной программы* с параметрами запроса, вводимыми из стандартного входного текстового файла, или получаемыми из командной строки UNIX. Второй способ задания параметров обязателен для работ, оцениваемых на хорошо и отлично. Параметры задаются с помощью ключей –f (распечатка файла) или –p <parameter> (параметры конкретного варианта задания). Получение параметров из командной строки производится с помощью стандартных библиотечных функций argc и argv.

Структуры данных и константы, совместно используемые программами, следует вынести в отдельный заголовочный файл.

В процессе отладки и тестирования рекомендуется использовать команды обработки текстовых файлов ОС UNIX и переадресацию ввода-вывода. Сгенерированные и отформатированные тестовые данные необходимо заранее поместить в текстовые файлы и распечатывать при протоколировании. Рекомендуется подобрать реальные или правдоподобные тестовые данные. Число наборов тестовых данных должно быть не менее трёх. Имя файла с бинарными данными является обязательным параметром второй программы.

Отчёт должен содержать оценку пространственной и временной сложности использованного алгоритма. В состав отчета также рекомендуется включить графическую иллюстрацию структуры файла и запроса на выборку.

В качестве дополнительного задания (принимается в качестве идеи решения задачи и способа тестирования):

– описать структуру файла как реляционную таблицу и сформулировать действие в виде запроса на структурированном языке запросов SQL, или на Прологе [10, 11]. Так, задание варианта 48 может быть специфицировано на языке SQL следующим образом:

**Описание таблицы:** CREATE TABLE ABIT (SURNAME CHAR (80), INITIALS CHAR(2), MATH INTEGER, PHYS INTEGER, LIT LOGICAL);
**Запрос:** SELECT SURNAME, INITIALS FROM ABIT WHERE (LIT=TRUE) AND (MATH+PHYS> (SELECT AVG(MATH+PHYS) FROM ABIT WHERE LIT=TRUE));

– добавить проверку правильности работы процедуры запроса в протоколе, путем сравнения её результатов с результатами, получаемыми из исходных текстовых файлов командами UNIX.

## Варианты заданий

### Содержимое и структура файла

**1 –11**. Сведения о составе комплектующих личных ПЭВМ в студенческой группе: фамилия владельца, число и тип процессоров, объём памяти, тип видеоконтроллера (встроенный, внешний, AGP, PCI) и объём видеопамяти, тип (SCSI/IDE, ATA/SATA), число и ёмкость винчестеров, количество интегрированных контроллеров и внешних (периферийных) устройств, операционная система.

**12 – 21**. Информация об успеваемости студентов данной группы по всем предметам: фамилия, инициалы, пол, номер группы, отметки по экзаменам и зачетам.

**22 – 31**. Сведения о вступительных экзаменах абитуриентов: фамилия, инициалы, пол, номер школы, наличие медали, оценки в баллах и зачет/незачет по сочинению.

**32 – 39**. Информация о пассажирах аэропорта: фамилия, инициалы, количество вещей, общий вес вещей, пункт назначения, время вылета, наличие пересадок, сведения о детях.

**40 – 47**. Общая информация о выпускниках школы студента: фамилия, инициалы, пол, номер класса, буква класса, в каком ВУЗе учится, где работает, в каком полку служит и т.п.

По усмотрению преподавателя задачи могут быть сформулированы, в соответствии с номером группы, для сотрудников фирмы (1), преподавателей кафедры (2), больных в больнице (3), жильцов дома (4), рейтинговых таблиц спортсменов (5), хит-парадов (6), осужденных в местах заключения (7), залогодателей ломбарда (8), клиентов службы знакомств (9), покойников на кладбище (10), покупателей интернет-магазина (11), абонентов телефонных компаний (12), владельцев автомобилей (13) и т.д.

Тестовые данные не должны нарушать действующее законодательство о персональных данных.

**Действия** (* обозначены более сложные задания)**:**

1. Найти всех владельцев двухпроцессорных компьютеров, имеющих не более $p$ внешних устройств.
2.* Напечатать список однофамильцев, имеющих однотипные компьютеры.
3.* Распечатать типичные конфигурации компьютеров в группе (более $p$ владельцев).
4. Отпечатать список студентов, компьютеры которых нуждаются в апгрейде (более $p$ устройств).
5.* Для всех студентов, имеющих более одного компьютера, распечатать сведения о самом мощном из них.
6. Распечатать сведения обо всех компьютерах-серверах и рабочих станциях.
7. Составить аннотированный список неукомплектованных компьютеров (некомплект – $p$ устройств).
8. Составить список мультимедийных компьютеров и бездисковых рабочих станций.
9. Составить список плохо сконфигурированных компьютеров.
10. Составить список компьютеров с фирменными комплектующими.
11. Перечислить все компьютеры студентов группы, платформа которых отлична от WINTEL.
12. Выяснить, сколько студенток группы $p$ получают стипендию.
13. Выяснить, сколько студенток группы $p$ имеют ровно одну пятёрку.
14. Выяснить, сколько студентов группы $p$ имеют больше двух троек.
15. Напечатать список потенциальных стипендиатов – студентов, у которых одна тройка, а все остальные оценки четвёрки и пятёрки или все пятёрки и одна четвёрка.
16. Найти фамилии лучших студенток курса (не имеющих отметок ниже четырех и по сумме баллов не уступающих другим студентам своей группы).
17. Выяснить, в какой группе студентки имеют максимальный средний балл.
18.* Выяснить, в какой группе разность между максимальным и минимальным средним баллом студентов максимальна.
19.* Выяснить, в какой группе учится максимальное число студентов с минимальным на курсе средним баллом.
20.* Выяснить, в какой группе учится максимальное число студенток с максимальным на курсе средним баллом.
21.* Напечатать список $p$ лучших студентов курса (с наивысшими средними баллами).
22. Найти абитуриентов-медалистов, не набравших проходной балл $p$.
23. Найти абитуриентов-медалистов, получивших неудовлетворительную оценку по математике.
24. Найти абитуриентов, имеющих заданную сумму баллов $p$.
25. Найти абитуриентов, имеющих сумму баллов от $p_1$ до $p_2$.
26. Найти абитуриентов-немедалистов, суммарный балл которых выше среднего.
27. Найти абитуриенток, получивших по двум предметам одинаковые оценки.
28. Найти абитуриенток, имеющих по всем предметам разные оценки.
29. Найти абитуриентов, получивших максимальную оценку по одному предмету, но не набравших проходного балла $p$.
30. Найти абитуриенток, получивших одинаковые оценки по всем предметам, но не набравшим проходного балла $p$.
31.* Найти абитуриентов, имеющих полупроходной балл, при наличии $p$ мест на факультете.
32. Найти пассажиров, вес багажа которых отличается от максимального веса менее чем на $p$ кг.
33.* Найти пассажира, средний вес вещей багажа которого отличается не более чем на $p$ кг от среднего веса вещей пассажиров для каждого рейса.
34. Найти пассажиров, имеющих более $p$ вещей.
35. Найти пассажиров, число вещей которых превосходит среднее число вещей не менее, чем на $p$ штук.
36.* Определить, имеются ли два пассажира, багаж которых совпадает по числу вещей и различается по весу не более чем на $p$ кг.
37. Выяснить, имеется ли пассажир, багаж которого состоит из $p_1$ вещей весом не менее $p_2$ кг.
38.* Дать сведения о пассажирах, число вещей которых не меньше, чем в любом другом багаже, а вес вещей не больше, чем в любом другом багаже с этим же числом вещей.
39. Выяснить, имеется ли пассажир, багаж которого превышает багаж каждого из остальных пассажиров и по числу вещей и по весу.
40.* Выяснить, имеются ли в школе однофамильцы.
41.* Выяснить, имеются ли однофамильцы в каких-либо параллельных классах.
42.* Выяснить, имеются ли однофамильцы в каком-нибудь классе.
43.* Выяснить, в каком классе учится максимальное число учениц.
44. Выяснить, на сколько учеников в $p$-х классах школы больше, чем в десятых.
45.* Найти среднее число учениц в классах школы.
46.* Найти классы, в которых число учеников больше числа учениц.
47. Найти классы, выпускники которых либо поступили в вузы, либо призваны на военную службу.

**Пример**

- Тип данных: имя и фамилия больного, температура.
- Задание: вывести имена и фамилии больных, температура которых ниже средней по больнице.
- Проект состоит из трех файлов: person.h, persons_dump.c, cool_persons.c

person.h:

```
#ifndef __person_h__
#define __person_h__

typedef struct {
  char name[50];
  int temp;
} person;

#endif
```

persons_dump.c

```
#include <stdio.h>
#include <string.h>
#include <errno.h>

#include "person.h"

void usage()
{
  printf("Usage: program filename\n");
}

int readperson(person *p)
{
  return scanf("%[^\t]\t%d\n", p->name, &p->temp) == 2;
}

int main(int argc, char* argv[])
{
  if (argc != 2) {
    usage();
    return 1;
  }
  person p;
  FILE *out = fopen(argv[1], "w");
  if (!out) {
    perror("Can't open file");
    return 2;
  }
  while (readperson(&p))
    fwrite(&p, sizeof(p), 1, out);
  return 0;
}
```

cool_persons.c

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

  #include "person.h"

  /*
  Программа просматривает данные бинарного файла пациентов больницы
  и выводит имена и фамилии больных, температура которых меньше средней
  по лечебному учреждению
  */

 void usage()
 {
   printf("Usage: program filename\n");
 }

int main(int argc, char* argv[])
{
  if (argc != 2) {
    usage();
    return 1;
  }
  person p;
  FILE *in = fopen(argv[1], "r");
  if (!in) {
    perror("Can't open file");
    return 2;
  }
  int temp_sum = 0;
  int n = 0;
  while (fread(&p, sizeof(p), 1, in) == 1) {
    temp_sum += p.temp;
    ++n;
  }
  fseek(in, 0, SEEK_SET);
  if (n == 0) {
    printf("No people, average temperature is not defined\n");
    return 3;
  }
  double avg = (double)temp_sum / n;
  while (fread(&p, sizeof(p), 1, in) == 1)
    if (p.temp < avg)
     printf("%s\n", p.name);
 return 0;
}
```

## Литература к заданию VI


1. *Кристиан К. Введение в операционную систему* UNIX. *–М.: Финансы и статистика, 1985.*
2. *Беляков И.Н., Рабовер Ю.И., Фридман А.Л. Мобильная операционная система: Справочник. –М.: Радио и связь, 1991.*
3. *Баурн С. Операционная система UNIX. –М.: Мир, 1986.*
4. *Зайцев В.Е. и др. CD-хрестоматия по курсу информатики. –М.: МАИ, 1997-2004.*
5. *Дейт К. Дж.. Введение в системы баз данных. – М.: Наука, 1981.*
6. *Каймин В.А., Титов В.К., и др. Информатика. Учебное пособие и сборник задач с решениями (для школьников). — М.: Бридж, 1994.*

**Вопросы для самостоятельного изучения к заданию VII курсового проекта**

**РАЗРЕЖЕННЫЕ МАТРИЦЫ**

1. Представление массивов в памяти ЭВМ.
2. Адресация элементов массивов и ее использование для представления структур данных.
3. Передача параметров-массивов и параметров-записей.
4. Ошибки адресации массивов и их последствия при выполнении программ в операционных системах с защитой памяти и без неё.
5. Приемы обработки и ввода/вывода массивов на скалярных ЭВМ.
6. Представление обычных и вариантных записей (структур) в памяти ЭВМ.
7. Приемы обработки и ввода/вывода записей.
8. Разреженные матрицы. Их представление в памяти ЭВМ.
9. Особенности хранения в памяти ЭВМ треугольных, симметричных и квазидиагональных матриц.
10. Приемы хранения и обработки разреженных матриц на языке Си.

## Задание VII. Разреженные матрицы.

Составить программу на языке Си с процедурами и/или функциями для обработки *прямоугольных* разреженных матриц с элементами целого (группы 6, 8), вещественного (группы 2-5), или комплексного (группы 1, 7) типов, которая:

1. вводит матрицы различного размера, представленные во входном текстовом файле в обычном формате (по строкам), с одновременным размещением ненулевых элементов в разреженной матрице в соответствии с заданной схемой;
2. печатает введенные матрицы во внутреннем представлении согласно заданной схеме размещения **и** в обычном (естественном) виде;
3. выполняет необходимые преобразования разреженных матриц (или вычисления над ними) путем обращения к соответствующим процедурам и/или функциям;
4. печатает результат преобразования (вычисления) согласно заданной схеме размещения и в обычном виде.

В процедурах и функциях предусмотреть проверки и печать сообщений в случаях ошибок в задании параметров. Для отладки использовать матрицы, содержащие 5–10% ненулевых элементов с максимальным числом элементов 100.

Вариант схемы размещения матрицы определяется по формуле $((N + 3) \% 4) + 1$, где $N$ — номер студента по списку в группе. Вариант преобразования определяется по формуле $((N - 1) \% 11) + 1$. Вариант физического представления (1 — отображение на массив, 2 — отображение на динамические структуры) определяются по формуле $([1.5\times((3+M) \% 9)]+N) \% 2+1$, где $M$ — номер группы. В случае использования динамических структур индексы заменяются соответствующими ссылками.

**Варианты схемы размещения матрицы:** все матрицы $m \times n$ хранятся *по строкам*, в порядке возрастания индексов ненулевых элементов.

1. Цепочка ненулевых элементов в векторе $A$ со строчным индексированием (индексы в массиве $M$ равны 0, если соответствующая строка матрицы содержит только нули)

| М: | Индекс начала 1-ой строки в массиве А | Индекс начала 2-й строки | ... | Индекс начала N-ой Строки |
|---|---|---|---|---|

| А: | Номер столбца | Значение | Индекс следующего ненулевого элемента этой строки (или 0) | Номер столбца | Значение | Индекс следующего ненулевого элемента этой строки (или 0) | ... |
|---|---|---|---|---|---|---|---|

Индекс, равный нулю, означает отсутствие ненулевых элементов в строке (или в ее остатке).
Если матрицы не изменяются программой, возможна экономия памяти за счет отказа от хранения в массиве $A$ индексов следующего элемента столбца (когда элементы идут подряд). Вставка и удаление при этом способе возможны, но чересчур д*о*роги: число перестановок элементов составит $O(N)$ вместо $O(1)$.

2. Один вектор:

Ненулевому элементу соответствуют две ячейки: первая содержит номер столбца, вторая содержит значение элемента. Нуль в первой ячейке означает конец строки, а вторая ячейка содержит в этом случае номер следующей хранимой строки. Нули в обеих ячейках являются признаком конца перечня ненулевых элементов разреженной матрицы.

| 0 | Номер строки | Номер столбца | Значение | Номер столбца | Значение | ... |
|---|---|---|---|---|---|---|

...

| 0 | Номер строки | Номер столбца | Значение | ... | 0 | 0 |
|---|---|---|---|---|---|---|