

# Bài tập thực hành

## Môn: TH Lập Trình Mạng

### Chương 2: Cơ bản về lập trình C trên Linux

#### 1. MỤC TIÊU

Bài thực hành đưa ra với mục tiêu cho sinh viên ôn tập lại kiến thức cơ bản về lập trình C trên Linux. Đồng thời, sinh viên có thể làm quen dần với những bài lập trình liên quan đến socket cơ bản với ngôn ngữ C.

#### 2. YÊU CẦU

- Kiến thức cơ bản về lập trình C
- Máy tính cài đặt Hệ điều hành Linux (khuyến khích distro Ubuntu), và các công cụ dịch C (gcc)

#### 3. BÀI THỰC HÀNH

**Bài 1.** Xây dựng một struct như sau để lưu thông tin của tên sv và điểm các môn cùng với điểm trung bình (mean):

```
struct student{
    char name[20];
    int eng;
    int math;
    int phys;
    double mean;
};
```

Nhập vào một mảng data lưu thông tin của những SV và các thông tin điểm như sau:

```
static struct student data[]={
    {"Tuan", 82, 72, 58, 0.0},
    {"Nam", 77, 82, 79, 0.0},
    {"Khanh", 52, 62, 39, 0.0},
    {"Phuong", 61, 82, 88, 0.0}
};
```

- Viết chương trình để tính và ghi lại điểm trung bình vào mảng data cho đúng. Sau đó in kết quả ra màn hình điểm số của mỗi SV và điểm trung bình tương ứng.
  - Mã nguồn chương trình:

```

29 void calMean(STUDENT *data, int count) {
30     STUDENT *p;
31     for (p = data; p < data + count; p++) {
32         p->mean = (p->eng + p->math + p->phys) / 3.0;
33     }
34 }
35
36 void prtStu(STUDENT *data, int count) {
37     STUDENT *p;
38     for (p = data; p < data + count; p++) {
39         printf("Name: %s\n", p->name);
40         printf("English score: %d\n", p->eng);
41         printf("Math score: %d\n", p->math);
42         printf("Physic score: %d\n", p->phys);
43         printf("Mean = %.2f\n", p->mean);
44         printf("-----\n\n");
45     }
46 }

```

- Kết quả:

```

phuca@phuca-VirtualBox:~/network/week2$ ./ex1
Name: Tuan
English score: 82
Math score: 72
Physic score: 58
Mean = 70.67
-----

Name: Nam
English score: 77
Math score: 82
Physic score: 79
Mean = 79.33
-----

Name: Khanh
English score: 52
Math score: 62
Physic score: 39
Mean = 51.00
-----

Name: Phuong
English score: 61
Math score: 82
Physic score: 88
Mean = 77.00
-----

```

b. Với nguyên tắc sắp xếp bảng như sau:

Mean	Grade
$90 \leq x \leq 100$	S
$80 \leq x < 90$	A
$70 \leq x < 80$	B
$60 \leq x < 70$	C
$x < 60$	D

Hãy viết chương trình để tính điểm trung bình của từng SV và hiện lên hạng của SV đó.

- Mã nguồn chương trình:

```
36 void prtStu(STUDENT *data, int count) {
37     STUDENT *p;
38     for (p = data; p < data + count; p++) {
39         printf("Name: %s\n", p->name);
40         printf("English score: %d\n", p->eng);
41         printf("Math score: %d\n", p->math);
42         printf("Physic score: %d\n", p->phys);
43         printf("Mean = %.2f\n", p->mean);
44         printf("Grade: %c\n", calGrade(p->mean));
45         printf("-----\n\n");
46     }
47 }
48
49 char calGrade(double mean) {
50     if (mean >= 90) {
51         return 'S';
52     } else if (mean >= 80) {
53         return 'A';
54     } else if (mean >= 70) {
55         return 'B';
56     } else if (mean >= 60) {
57         return 'C';
58     } else {
59         return 'D';
60     }
61 }
```

- Kết quả:

```

phuca@phuca-VirtualBox:~/network/week2$ ./ex1
Name: Tuan
English score: 82
Math score: 72
Physic score: 58
Mean = 70.67
Grade: B
-----

Name: Nam
English score: 77
Math score: 82
Physic score: 79
Mean = 79.33
Grade: B
-----

Name: Khanh
English score: 52
Math score: 62
Physic score: 39
Mean = 51.00
Grade: D
-----

Name: Phuong
English score: 61
Math score: 82
Physic score: 88
Mean = 77.00
Grade: B
-----

```

- c. Sử dụng typedef để định nghĩa kiểu STUDENT, sử dụng con trỏ p để duyệt các thành phần trong mảng data:

```

typedef struct student{
    char name[20];
    int eng;
    int math;
    int phys;
}STUDENT;

STUDENT data[]={
    {"Tuan", 82, 72, 58},
    {"Nam", 77, 82, 79},
    {"Khanh", 52, 62, 39},
    {"Phuong", 61, 82, 88}
};

STUDENT *p;

```

Câu hỏi 1: Upload mã nguồn chương trình lên

**Bài 2.** Tải 2 tệp server.c và client.c về, dịch để ra chương trình chạy.

```
gcc server.c -o server
```

```
gcc client.c -o client
```

```
phuca@phuca-VirtualBox:~/network/week2$ gcc server.c -o server
phuca@phuca-VirtualBox:~/network/week2$ gcc client.c -o client
```

Chạy hai chương trình đó bằng 2 lệnh sau (chú ý đúng thứ tự):

```
./server
```

```
phuca@phuca-VirtualBox:~/network/week2$ ./server
Server listening on port 8080
Message from client: Hello from client
Hello message sent to client
```

```
./client
```

```
phuca@phuca-VirtualBox:~/network/week2$ ./client
Hello message sent to server
Message from server: Hello from server
```

Câu hỏi 2: Đọc qua mã nguồn của 2 chương trình, kết hợp với kết quả đầu ra của việc chạy 2 chương trình, hãy trình bày nhiệm vụ của từng chương trình server và client.

*Chương trình mô phỏng giao tiếp giữa server và client*

- *Nhiệm vụ của chương trình server:*
  - + *Tạo socket và lắng nghe kết nối từ client.*
  - + *Khi nhận được kết nối, đọc dữ liệu từ client và gửi lại message “Hello from server”.*
  - + *Đóng kết nối.*
- *Nhiệm vụ của chương trình client:*
  - + *Kết nối với server thông qua socket.*
  - + *Gửi message “Hello from client” đến server.*
  - + *In message nhận từ server ra màn hình.*

**Bài 3.** Dựa vào mã nguồn của bài 2, hãy viết hai chương trình server và client, đặt tên là `capitalize_server` và `capitalize_client` trong đó:

- Chương trình server lắng nghe thông điệp từ client.
- Chương trình client sẽ yêu cầu người dùng nhập một xâu, sau đó gửi xâu đó lên cho server.

```

42 // Enter string
43 printf("Enter string: ");
44 fgets(message, BUFFER_SIZE, stdin);
45 message[strcspn(message, "\n")] = 0;
46 send(sock, message, strlen(message), 0);

```

- Sau khi nhận được xâu từ client, server sẽ thực hiện viết hoa hết xâu đó lên và gửi ngược lại cho client.

```

58 // Capitalize
59 capitalize(buffer);
60 send(new_socket, buffer, strlen(buffer), 0);

```

```

69 void capitalize(char *str) {
70     for (int i = 0; str[i]; i++) {
71         str[i] = toupper(str[i]);
72     }
73 }

```

- Client hiển thị xâu đó ra màn hình.

Câu hỏi 3: Upload mã nguồn chương trình lên.

*Kết quả:*

```

phuca@phuca-VirtualBox:~/network/week2$ ./capitalize_server
Server listening on port 8080
Message from client: Dinh Thi Hong Phuc

```

```

phuca@phuca-VirtualBox:~/network/week2$ ./capitalize_client
Enter string: Dinh Thi Hong Phuc
Message from server: DINH THI HONG PHUC

```