

## Code Hướng dẫn Thực hành Hệ điều hành Ptit

1) Giả sử thư mục gốc của hệ thống file FAT 16 sử dụng tên file độ dài tối đa 8 kí tự đã được đọc vào bộ nhớ tại địa chỉ “void\*root”.Viết đoạn chương trình trên C/C++ thực hiện 2 việc: in tên và độ dài các file trong thư mục gốc, tìm một tên file cho trước trong thư mục gốc và số thứ tự cluster đầu tiên của file đó.

//In tên và độ dài các file trong thư mục gốc

```
printf("List file of the root \n");
```

```
printf("Filename\t Size\n");
```

```
for (int i=0;i<boot.ROOT_size;i++)
```

```
{
```

```
    if (root[i].name[0]==' '|| root[i].name[0]==0) continue;
```

```
    for (int j=0; j<8 && root[i].name[j]!=' ';j++)
```

```
        printf("%c", root[i].name[j]);
```

```
    if (root[i].ext[0]!=' ')
```

```
    {
```

```
        printf(".");
```

```
        for (int u=0;u<3 && root[i].ext[u]!=' ';u++)
```

```
            printf("%c",root[i].ext[u]);
```

```
    }else printf("  ");
```

```
    printf (" \t %ld\n", root[i].size);
```

```
}
```

//Tìm một file có tên cho trước trong thư mục gốc và số thứ tự cluster đầu tiên của file đó

```

//find a file with name given in char filename[]

int k;

char str[9];

int first_cluster=-1;

for (i=0;i<boot.ROOT_size;i++)

{
    //Copy root[i].name to str to make null-terminated string

    for(k=0;k<8 && root[i].name[k]!='';k++)

        str[k]=root[i].name[k];

    str[k]=0;

    //Comparing

    char filename[8];

    printf("Enter a file name:");

    scanf("%s",filename);

    if (strcmp(str,filename)==0)

    {

        first_cluster=root[i].first_cluster;

        break;

    }

}

```

2) Giả sử bảng FAT đã được đọc vào bộ nhớ tại địa chỉ “void\*fat”, viết đoạn chương trình trên C/C++ để liệt kê tất cả các cluster trong số N cluster đầu tiên. Giả sử 1 file bắt đầu từ cluster n, viết đoạn chương trình liệt kê các cluster thuộc về file đó?

```
//Liệt kê các cluster trống trong N cluster đầu tiên
```

```
for (i=2; i<=N; i++) //first 2 cluster is not used
```

```
    if (fat[i]==0)
```

```
        printf("%d ",i);
```

```
//=====
```

```
//Liệt kê các cluster thuộc về file bắt đầu từ cluster thứ n
```

```
int cur=n;
```

```
while (cur<0xFFF8)
```

```
{
```

```
    printf("->%d ",cur);
```

```
    cur=fat[cur];
```

```
}
```

3) Tên nhà sản xuất

```
printf("Ten nha san xuat:");
```

```
for(int i=0;i<8;i++)
```

```
    printf("%c",boot.OEM[i]);
```

4) Xây dựng struct BOOT

```
struct BOOT{ //for FAT16
```

```
    char jmp[3]; //lệnh JUMP, chỉ thị cho CPU bỏ qua phần thông tin và nhảy  
    //thực hiện phần mã của hệ điều hành nếu đây là đĩa mềm HDH
```

```
    char OEM[8]; //ten hang san xuat, bổ sung dấu " " ở cuối cho đủ 8bit
```

```

    int bytes_per_sector;

    char sector_per_cluster;

    int reserved; //so luong sector dành cho vùng đĩa đầu đến trước FAT, bao
gom boot sector và các sector dự phòng

    char FAT_cnt;//số lượng bảng FAT, thường bằng 2

    int ROOT_size;//số khoản mục tối đa trong thư mục gốc ROOT

    int total_sectors;

    char media;//mô tả loại đĩa

    int FAT_size;//kích thước FAT tính bằng sector

    int sector_per_track;

    int head_cnt;//số lượng đầu đọc

    long hidden_sectors;//số sectors ẩn

    long total_sectors_long;//tổng số sector trên đĩa trong trường hợp có nhiều
hơn 65535

    char unknown[3];

    long serial;//số seri của đĩa, được tạo lúc format đĩa

    char volume[11];//nhãn của đĩa được tạo khi format

    char FAT_type[8];//tên hệ thống file FAT ví dụ: "FAT12 ", "FAT16 "

    char loader[448];//mã mỗi HDH

    char mark[2];//dấu hiệu boot sector

}

```

## 5) Xây dựng struct ROOT

```

struct ROOT{

    char name[8];//tên file, thêm dấu trống ở cuối nếu ngắn hơn 8 bytes

```

bytes

```
char ext[3]; //phan mo rong, them bang dau trang o cuoi neu ngan hon 3
```

```
char attr; //byte thuoc tinh cua file
```

```
char reserved[10]; //du phong
```

```
char time[2];
```

```
char date[2];
```

```
int first_cluster;
```

```
long size;
```

```
}
```