

Computer Systems Lab

Stefan M. Freudenberger

Computer Systems, ETH Zürich



File System Lab: Sample Solution

File table

```
struct file_table_entry {
    __u16 cluster;
    __u16 first_cluster;    /* the first cluster of the file */
    __u32 position;         /* the current pos in the file */
    __u32 size;             /* the size of the file */
    __u32 dir_entry_offset; /* pos. of the dir entry: size */
    __u16 dir_entry_cluster; /* pos. of the dir entry: cluster */
};

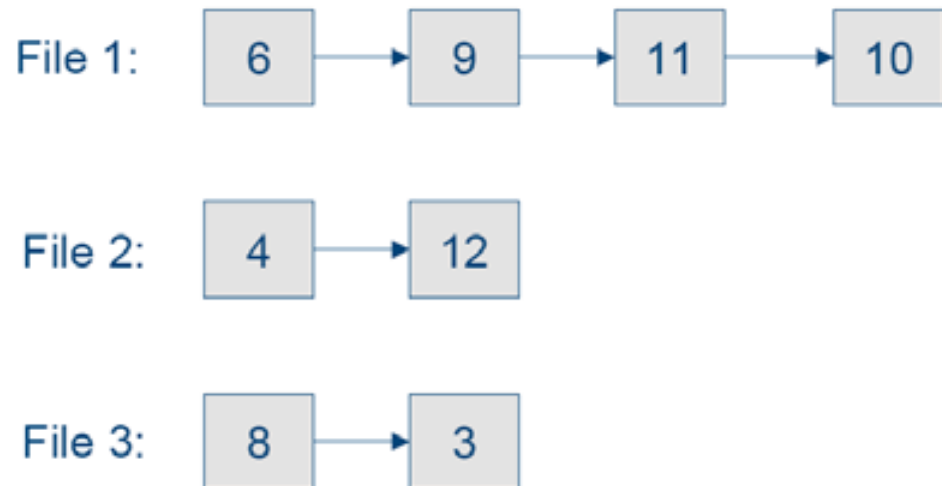
...

fte = (struct file_table_entry*)malloc(
    sizeof(struct file_table_entry) * MAX_FILES
);

...
```

FAT: Table Structure

0	
1	
2	EOF
3	EOF
4	12
5	FREE
6	9
7	BAD
8	3
9	11
10	EOF
11	10
12	EOF
13	FREE
...	



File System Lab: Sample Solution

Reading FAT entries

```
static __u16 get_fat_entry(__u16 idx) {
    /* uv.wx.yz -> xuv, yzw */
    if ((idx & 0x1) == 1) {
        /* odd index */
        next_idx = idx + 1;
        index =
            (FAT[next_idx + next_idx / 2 - 1] << 4) +
            ((FAT[next_idx + next_idx / 2 - 2] >> 4) & 0xF);
    }
    else {
        /* even index */
        index =
            FAT[idx + idx / 2 ] +
            (FAT[idx + idx / 2 + 1] & 0xF) << 8;
    }
    return index;
}
```

File System Lab: Sample Solution

Reading clusters

```
static void get_cluster(__u16 cluster_number, void* buffer)
{
    __u8 sector; /* current sector */
    // #define CLUSTER2SECTOR(cluster)
    //      (((cluster)-FIRST_CLUSTER) * (int)fbs.sec_per_clus)
    // #define DIR_SIZE
    //      ( (fbs.dir_entries * sizeof(struct dos_dir_entry))
    //        / fbs.sector_size)
    // #define ROOT_DIR (fbs.reserved+fbs.fats*fbs.fat_length)
    for (sector = 0; sector < fbs.sec_per_clus; ++sector)
    {
        bios_read((int)(    ROOT_DIR + DIR_SIZE
                           + CLUSTER2SECTOR(cluster_number) + sector),
                  buffer + (sector * fbs.sector_size));
    }
}
```

File System Lab: Sample Solution

```
static void get_date_and_time(__u16* dos_time,
                             __u16* dos_date)
{
    time_t cur_time_t = time(NULL);
    struct tm* cur_time = localtime(&cur_time_t);
    /* 15-09 year (0 = 1980, 127 = 2107)
     * 08-05 month (1 = January, 12 = December)
     * 04-00 day (1 - 31) */
    *dos_date = cur_time->tm_mday;
    *dos_date |= ((cur_time->tm_mon+1) << 5) & 0x01e0;
    *dos_date |= ((cur_time->tm_year-80) << 9) & 0xfe00;
    /* 15-11 hours (0-23)
     * 10-05 minutes (0-59)
     * 04-00 seconds/2 (0-29) */
    *dos_time = cur_time->tm_sec / 2;
    *dos_time |= (cur_time->tm_min << 5) & 0x07d0;
    *dos_time |= (cur_time->tm_hour << 11) & 0xf900;
}
```

File System Lab: Sample Solution

Writing (simplified)

```
int fs_write(int fd, void* buffer, int len)
{
    /* get the file table entry from the file descriptor */
    fte = (struct file_table_entry*)file_table[fd];
    if (fte->position != fte->size) {
        /* the file is not empty */
        fte->position = fte->size;
        fte->cluster = get_cluster_number(fte->first_cluster,
                                         fte->position);
    }

    /* allocate a buffer for the RW operations */
    disk_buffer = (char*)malloc(BUFFER_SIZE);

    [...]
```

File System Lab: Sample Solution: Writing (cont.)

[...]

```
/* compute the current position in the current cluster */
t_rem = len - t_wri;
while (t_rem) {
    get_cluster(fte->cluster, disk_buffer);
    cl_offset = ((int)fte->position) % ((int)CLUSTER_SIZE);
    cl_remain = (CLUSTER_SIZE - cl_offset);
    /* don't write more than requested */
    if (cl_remain > t_rem) { cl_remain = t_rem; }
    /* copy the buffer to the right position */
    memcpy(disk_buffer + cl_offset, buffer + t_wri,
           (size_t)cl_remain);

    /* write back the cluster */
    put_cluster(fte->cluster, disk_buffer);
}
```

[...]

File System Lab: Sample Solution: Writing (cont.)

```
[...]  
  
t_rem -= cl_remain;  
fte->position += cl_remain;  
t_wri += cl_remain;  
fte->size = fte->position;  
if (t_rem) { /* we need another cluster */  
    fte->cluster = append_FAT_cluster(fte->cluster,  
                                     FAT1, FAT2);  
}  
} /* end while */  
  
[...]
```

File System Lab: Sample Solution: Writing (cont.)

[...]

```
/* update dir structures */  
dep = malloc(sizeof(struct dos_dir_entry_pos));  
dep->cluster = fte->dir_entry_cluster;  
dep->offset = fte->dir_entry_offset;  
dep->de = NULL;
```

```
read_directory_entry(&dep);
```

```
get_date_and_time(&dos_time, &dos_date);  
dep->de->size = fte->size;  
dep->de->date = dos_date;  
dep->de->time = dos_time;
```

```
write_directory_entry(dep);
```

```
[...]
```

```
return (int)t_wri;
```

```
}
```