

- exception.cc:

```
case SC_Open:
    doSC_Open();
    break;
```

- schandle.cc :

```
/*
```

Input: user space address

file name (reg4)

file type (reg5)

0 - standard file

1 - read only

2 - encrypted

Output: -1 - error

OpenFileID -success

Purpose: do Open a file

```
*/
```

```
int doSC_Open()
```

```
{
```

```
    int virtAddr = machine->ReadRegister(4);
```

```
    int type = machine->ReadRegister(5);
```

```
    if (type < 0 || type > 2)
```

```
    {
```

```
        printf("\n SC_OpenError: unexpected file type: %d",type);
```

```
        return -1;
```

```
    }
```

```
    int id = currentThread->fTab->FindFreeSlot();
```

```
    if (id < 0)
```

```
    {
```

```
        printf("\n SC_OpenError: No free slot.");
```

```
        return -1;
```

```
    }
```

```
    char *filename = User2System(virtAddr,MaxFileLength+1);
```

```
    if (filename == NULL)
```

```
    {
```

```
        printf("\n Not enough memory in system");
```

```
        machine->WriteRegister(2,-1);
```

```

    delete filename;
    return -1;
}

if (strlen(filename) == 0 || (strlen(filename) >= MaxFileLength+1))
{
    printf("\n Too many characters in filename: %s",filename);
    machine->WriteRegister(2,-1);
    delete filename;
    return -1;
}

OpenFile* of = fileSystem->Open(filename);

if (of == NULL){
    printf("\n Error opening file: %s",filename);
    machine->WriteRegister(2,-1);
    delete filename;
    return -1;
}

int rs = currentThread->fTab->fdOpen(virtAddr,type,id,of);

machine->WriteRegister(2,rs);

return rs;
}

```

- fdtable.cc :

```

/*
Input: user space address
      filetype (0-standard, 1- read only, 2- encrypt)
Output:OpenFileID - success, -1 - fail

Purpose:open a file
*/
int  FDTable::fdOpen(int virtAddr,int t,int i,OpenFile *of)
{
    int vA = virtAddr;
    int type = t;
    int id = i;

```

```

FDRw *rw;
FDRo *ro;
FDEnc *enc;
switch(type)
{
case WRFILE:
    rw = new FDRw(of);
    fTable[id] = (FDBase*) rw;
    break;
case ROFILE:
    .....
case ENCRYPTFILE:
    .....
default:
    printf("\n FDTable:OpenFileError - unknow file type %d",type);
    break;
}

bm->Mark(id);

return id;
}

```

- **fdbase.cc**

- **fdrw.cc**

```

FDRw::FDRw(OpenFile *f)
{
    file = f;
}

```