Operating Systems

Lesson 2

Plan

- Assignment (due in 2 weeks)
  - Description
  - Hints
- Windows Files
  - File object
  - Disks and clusters
- Sample
  - Enumerate file in the folder
  - Unicode signature

Electronic Submission (Reminder)

- Create a directory called **os08b** in your home directory
- For each homework assignment, create directory within ~/os08b, called **/ex01**, **/ex02**, etc.
- For each assignment create two subdirectories **code** and **bin**
- Under **code**
  - source code (.cpp, .h)
  - project files (.sln, .vcproj)
- Under bin
  - executables (.exe)
- Make sure you give read permissions to everyone for the whole path
- **Important**: write your username and id on the hardcopy

Assignment #1

Assignment #1(due in 2 week)

- Write “dir2file” utility
  - Accept path to a folder and path to an output file
  - The utility should produce a file with list of files in the input folder and exit immediately
  - Do not include(ignore) system files , sub-folders and “dotted” folders in the output
  - New output file shall be created/ overwritten each time the utility will run
  - Output file will have a line per file (see format below)
  - Unicode (e.g. Hebrew) file names and paths should be processed
  - Assume file sizes are less then 2Mbytes
  - Assume folder of the output file exists (but not necessary file itself)
  - All file operations are through system calls only (no CRT calls)
- Input Format/Usage
  - Dir2File .exe “c:\\my folder\\” “c:\\”
- Output Format
  - FileName <TAB>FileSize<TAB>FileSizeOnDisk<TAB>FileHashValue\r\n
  - Example: **File.txt 1245 4096 15**
  - Note that first field does not include full file path
  - Hash value is a numeric value of file middle byte (2nd byte for file of length 4 bytes and 3rd for length 7 bytes)
  - FileSizeOnDisk will be discussed today
- Will be checked automatically so please stick to input/output format

Assignment #1: (Tips)

- Use _stprintf to format a string in memory before writing to a file
- Use "%s\t%d\t%d\t%d\r\n" format string for _stprintf
- Use TCHAR and LPTSTR and _T to support Unicode
- Check return values through assert
- Divide your code into functions and debug them one by one
- Intelligently comment your code
- Design test sets and test your code before submission
- Use "command arguments" entry in Visual Studio project's debug property window

Assignment #1

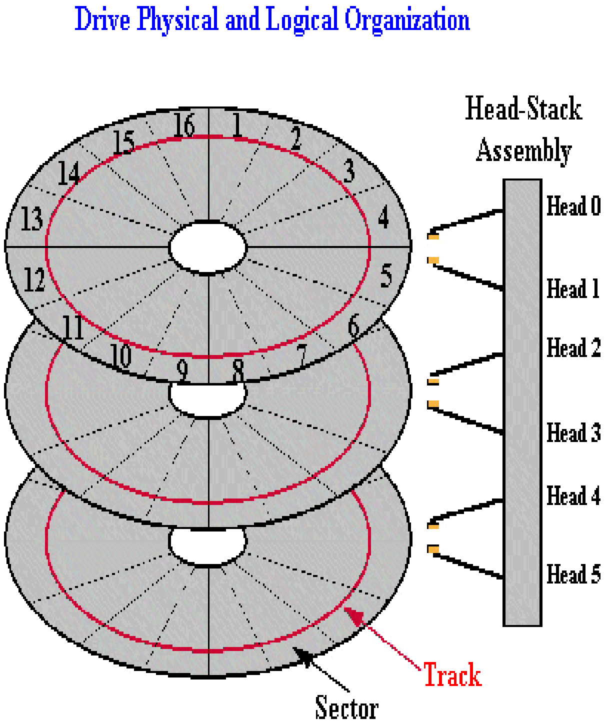
Assignment #1 (Hints)

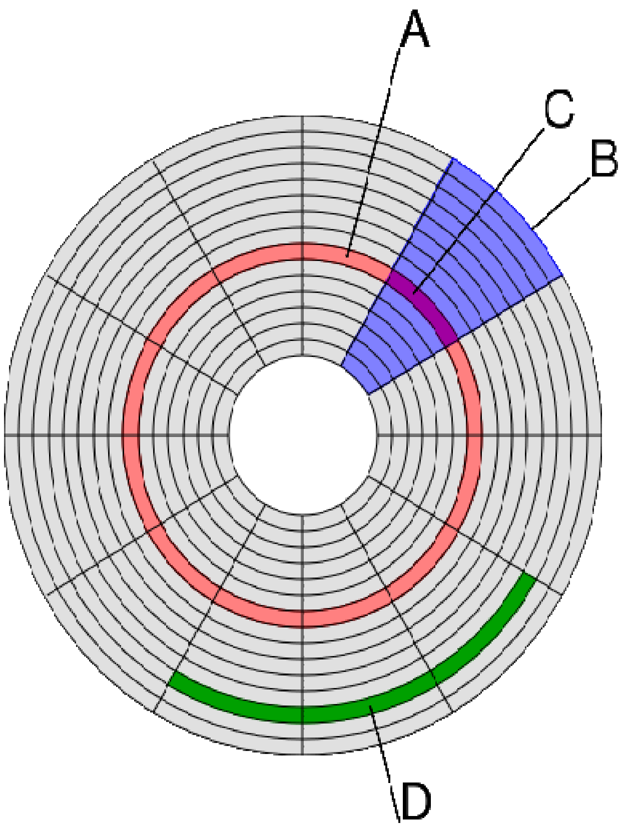
- Use safe functions
  - _tsplitpath_s, _tcscat, _tcscpy,_stprintf
  - strsafe.h (StringCchCopy, StringCchCat, ...)
- ZeroMemory (structure initialization)
- CreatFile/ReadFile/WriteFile/CloseHandle
- FindFirstFile/FindNextFile/FindClose
- SetFilePointer
- GetDiskFreeSpace

Windows File

- An object (API requires handle)
- Has a name: mydoc.doc
- Has a full path: c:\my folder\mydoc.doc
- Has a size
- Windows keeps file times
  - Create/Write/Access time
- File Attributes
  - Read-only/System/Hidden
- Has a content (bytes)
- Sequential or direct access
  - Sequential access (from start to end)
  - Direct access (read at specific position)

Hard disk


Drive Physical and Logical Organization
Head-Stack Assembly
Head 0
Head 1
Head 2
Head 3
Head 4
Head 5
16
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
Track
Sector


A
C
B
D

Cluster

- Minimal unit of physical storage management
- A File stored at one or more
- Half-used cluster not used by other files
- Storage required to store a file is number of cluster it occupies multiplied by cluster size in bytes.
- Different File Systems have different cluster sizes. (e.g. NTFS/FAT32)
- Use Windows file properties dialog to see how much space a file occupies

More hints for assignment

- Use **GetDiskFreeSpace** to disk calculate cluster size
- Use **_tsplitpath_s** to construct disk root folder (remember different disks might have different file systems)
- Before writing to the output file , put 2 bytes UNICODE signature
- Don’t forget to close a file.
- Use **SetFilePointer** to set a reader pointer at the desired position