Team B5 OS Project Design

Project Specification

We have to implement byte stream file system and disk cache. We have outlined it by dividing our disk into two partitions one handling the metadata and other handling the data for the files. In metadata partition we have kept the relevant information for each file and directory existing along with the Disk Status Map (DSM) to help in handling file operation. Whereas in data partition, files are allotted data blocks according to their size and handled using chaining.

Below, we have provided a detailed structure of each partition.

Metadata partition structure:-

First block will contain the following

- Root directory block number (first 5 Bytes)
- Starting block no. for the data section (5B)
- Disk Status Map - Its size will be determined according to the total number of blocks on disk. It will be string of 0s and 1s where 1 at i^{th} position indicates that i^{th} block is in use whereas 0 indicates that the block is free. (DSM is allocated continuous blocks if it exceeds)
- Following DSM, new block will be given to metadata block for root directory whose number is stored at the very beginning of this block.

Metadata block Structure for directories

- First field is name of the directory which we are assuming to be not more than 15 characters. (If it is less than 15B then we are padding it with redundant character so as to leave the scope for renaming)
- Next field is the type where 1 indicates it to be file and 0 indicates directory (1B)
- Next field contains the owner id for protection purposes. (2 Bytes)
- Next field will contain the permission list whose size will be 100 (according to the number of possible number of total users). i^{th} byte corresponding to the i^{th} owner will be set to 6 whereas other will be set to 0.
- Next field will contain time stamp according to the last modification :dd-mm-yyyy hh-mm-ss (14 B)
- Next field will be the directory entry list for the directory. Each entry will have following structure
    - Valid Bit (1B) which will indicate whether the entry is valid or not.
    - File/Dir Name (15B)
    - Block number (5B) which will be the starting block number in metadata section for the corresponding file/directory

- Last 5B of each directory metadata block will point to the next directory metadata block of current block (chaining)

Metadata block Structure for files

- First field is name of the file which we are assuming to be not more than 15 characters. (If it is less than 15B then we are padding it with redundant character so as to leave the scope for renaming)
- Next field is the type where 1 indicates it to be file and 0 indicates directory (1B)
- Next field contains the owner id for protection purposes. (2 Bytes)
- Next field will contain the permission list whose size will be 100 (according to the number of possible number of total users). i^{th} byte corresponding to the i^{th} owner will be set to 6 whereas other will be set to 0.
- Next field will contain time stamp according to the last modification :dd-mm-yyyy hh-mm-ss (14 B)
- Next field is open count which will keep a count to indicate no. of processes which have opened the same at this instant. (2B)
- Next field will indicate the type of lock (if any) applied by a process. It can be r:read, w:write, c:closed (1B)
- Next field will be the starting block number of the file in data partition (5B)
- Next field will be the block number of the last block corresponding to the file in data partition (5B)
- Next field will contain the last byte in the last block corresponding to the file in data partition (3B)

Data partition structure:-

- Each block will contain the data of the corresponding file. The last 5B of the data block will point to next block of the file.(if any).

Miscellaneous:-

- Open File Table(OFT) is maintained in memory (code). FCB (an entry of OFT) will contain following entry
    - Valid bit, file name, process id, block number of file meta data, permission, current block number in file data, current offset in current block in filedata

Cache Implementation

- Cache is an list of structs. Each struct has a data block and its associated metadata. Along with it will have following fields
    - Fd of the file (points to fcb), block number in the file, block number of this block in the disk, valid bit, dirty bit, reference bit and data field according to the replacement policy.

Syscalls –

- pwd
- mkdir
- ls
- create file
- open file
- close file
- read file
- write file
- rename file
- delete file
- delete directory
- rename directory
- cd
- chmod

Implementation of some syscalls

- pwd - print path name of current directory
    - we will maintain a current directory struct in the code which will store the path of the current directory as well as the starting block number of the directory in the metadata section
    - We will maintain this variable for each process.
- ls – Prints all the file and directories present in current directory according to the permissions of user
    - Using the current directory struct we will access the metadata block of the former and we will traverse the directory entry list sequentially. For each entry we will check the permission for the user, valid bit and print it accordingly.
- mkdir – makes the directry with corresponding name in current directory
    - Checking permission of current directory if allowed
    - A new block is allocated to the directory in metadata block and it is ntialized with appropriate values.

    - An entry corresponding to the same is made and entered in the directory list of current meta data block
    - Updating DSM
- Create file- same as making directory but a block is also allocated in data section and DSM is updated for this block too.
- Rename file/directory
    - Check permissions and status of file as it has to be in closed mode
    - Check permission of parent dir also.
    - Changing the name in the metadata of file and also to change the same in the metadata block of the parent directory.
- Cd
    - Struct that maintain current directory will be updated
    - Both absolute and relative addresses will be handled
- Chmod
    - Change the permission in the corresponding metadata block after checking the permissions and the status of the file (it should be closed)
- Other syscalls will be implemented in the similar fashion

Conclusion

- We will be developing our pranali OS according to the above design. Some features in the design might be changed depending upon the ease in the implementation.

Work to be done by next week

- We aim to finish the above mentioned implementation.

Work Distribution

- 3 people will handle read, write and cache portion of the implementation while other three will implement metadata, remaining syscalls etc.

Current Status

- We have finalized our design and it is same as mentioned above.