



Concepts Systems

The Center of Excellence

Project Title	MailDisk
Registration Number	
Engineering College	
Project Members	
Project Guide	
Academic Year	
Synopsis	

- **The motivation behind the project, and what it does:**

Today as internet is evolving, e-mail is gaining popularity. The email account space provided by the email-providers is also increasing exponentially. Gone are the days when account space of 6MB was considered to be a matter of luxury. Now providers like *rediffmail*, *gmail* provide space up to 1GB. Now this space is currently used only for storing mails. However, nobody utilizes the complete space for storing mails and much of the space is left unutilized from user perspective.

Our project aims at utilizing this space by treating the mail account as secondary storage device. The various traits of mailboxes are instrumental to promote us to utilize it for storage purpose. These traits include:

1. **Reliability:** You don't find some of your mails missing due to hard disk crash at server side. It is the responsibility of the server to handle such cases.
2. **Mobility:** The space is accessible from any point in the world (provided internet connection is available).
3. **Availability:** It's very rare that an email-provider doesn't provide its service to its users.

- **Design of Project:**

1. The crux of our system lies in deciding the operating level at which the system is implemented. Implementing at file system level raises many issues like:
 - a. The mailbox (i.e. hard disk) is accessible (mounted) from multiple sites (multiple machines). To exploit this facility we need to implement a distributed file system.
 - b. At file system level we would have to implement every system call which asks for gigantic amount of efforts. Instead we can use the already existing efficient file systems.
 - c. Database systems cannot make use of mailbox space directly as they require raw block operations. Writing our own file system would have required to take this issue under consideration.Owing to these (and few more) issues, we decide to implement our system at a level below file system, that is, at **block level**.
2. [Figure1](#) shows the basic architecture diagram of our system. The file-system tries to read/write a block to the mail disk. The mail disk device driver contacts the mail sever to read/write the block as a mail (one mail for each block) using SMTP for sending mail (block write) and IMAP protocol for retrieving mail (block read) and completes the request of the file system.
3. We choose to implement our system at block layer. The advantage of this is that any file system can operate on top of us. The upper file system is concerned regarding handling issues like journaling, handling locks, recovery amongst others.
4. Further the efficiency of the system is improved by introducing a buffer cache between our driver and the mail server.

- **The algorithms and methods used in each part:**

In our system, a block write is equivalent to sending a mail using SMTP protocol to the mail server. The mail has its subject as the corresponding block number. Similarly to read a block, a search is carried out based on the subject field with the required block number being the content to be searched. The searched mail is then fetched using IMAP protocol.

A new block write sends a mail to the mail server with some block number as its subject. Now for reading a particular mail and hence during search operation a situation can arise wherein more than one mail has in its subject field the content that is being searched (block number). In this case the most recent mail is fetched and the remaining stale mails are deleted. This process is carried out during **read operation**. The purpose of this being that it **ensures atomicity**. Had this process been carried out during write operation, then chances of inconsistency would arise due to system failure.

- **Future Plans:**

To improve the read performance, we can implement a 'write through' cache between MAILDISK device driver and e-mail server. This would permit speedy read operations of recently used blocks (mails).

Other technique to improve reliability would be to use block replication. For this we can send blocks to multiple email-servers (*rediffmail, gmail, runbox* etc.), a technique similar to *mirroring* in RAID. While reading, the block(mail) first received from either mail servers is utilized.

FIGURE1

