(f) Reset memory
(g) Set page replacement policy
(h) Display working set
(i) Redo previous reference results
(j) Exit

- enter the size of the process in the input window.
- choose replacement policy through "set replacement policy"
  — Either get the reference string from the system or man
- The detailed page table status and also the hit ratio for the
  policy chosen can be obtained.
  — If the policy chosen was WSClock, we can also display the
- For a comparative study, use the same reference string
  other policy.
- Reset memory to clear the current memory status.
- Reference results list out detailed reference information.

(vii) Tools used: The page replacement simulator is developed using C
language in Linux environment with Ncurses graphics library for the
(viii) Bug reports and feedback: All bugs are detected and feedback is

CASE STUDY 4: DESIGN STRATEGY OF MINI SHELL[6]

Statement of the problem: This Mini Shell is designed to simulate the
UNIX kernel. In our simulation we have tried to include as many
possible of the bash shell. This simulation not only works well for all the
commands, but we have tried to include as many internal commands as
major hurdle is how to execute these (internal) commands.

Approach taken and elaboration: The basic program structure
Figure 25.6. Below we describe the main programs developed to implement
Design steps:

1. To execute all the UNIX external commands. To do this system, call
   used.
2. To make provision for all kinds of redirection possible, i.e. input <
   and append(>>).
3. To make provision for piping (single level), in process creating
   processes and executing commands separated by pipe(|).
4. To ensure additional functionalities that are not executed by
   environment variables, cd(all variations), history and exit comm

[6]Student Project Team: Abhigyan Agarwal, Abhijeet Rao R., and Abhinav Garg

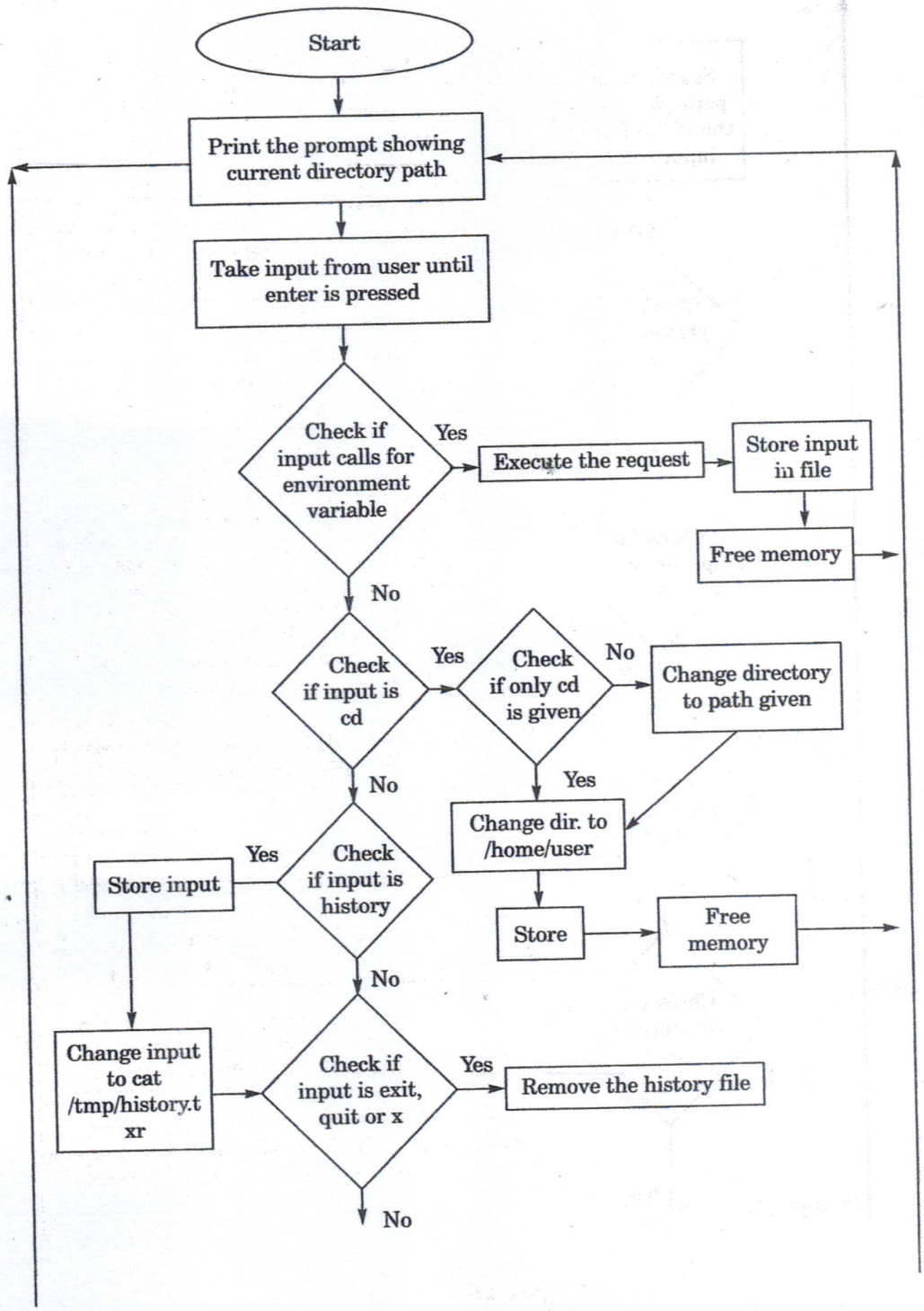

Figure 28.7 Contd.

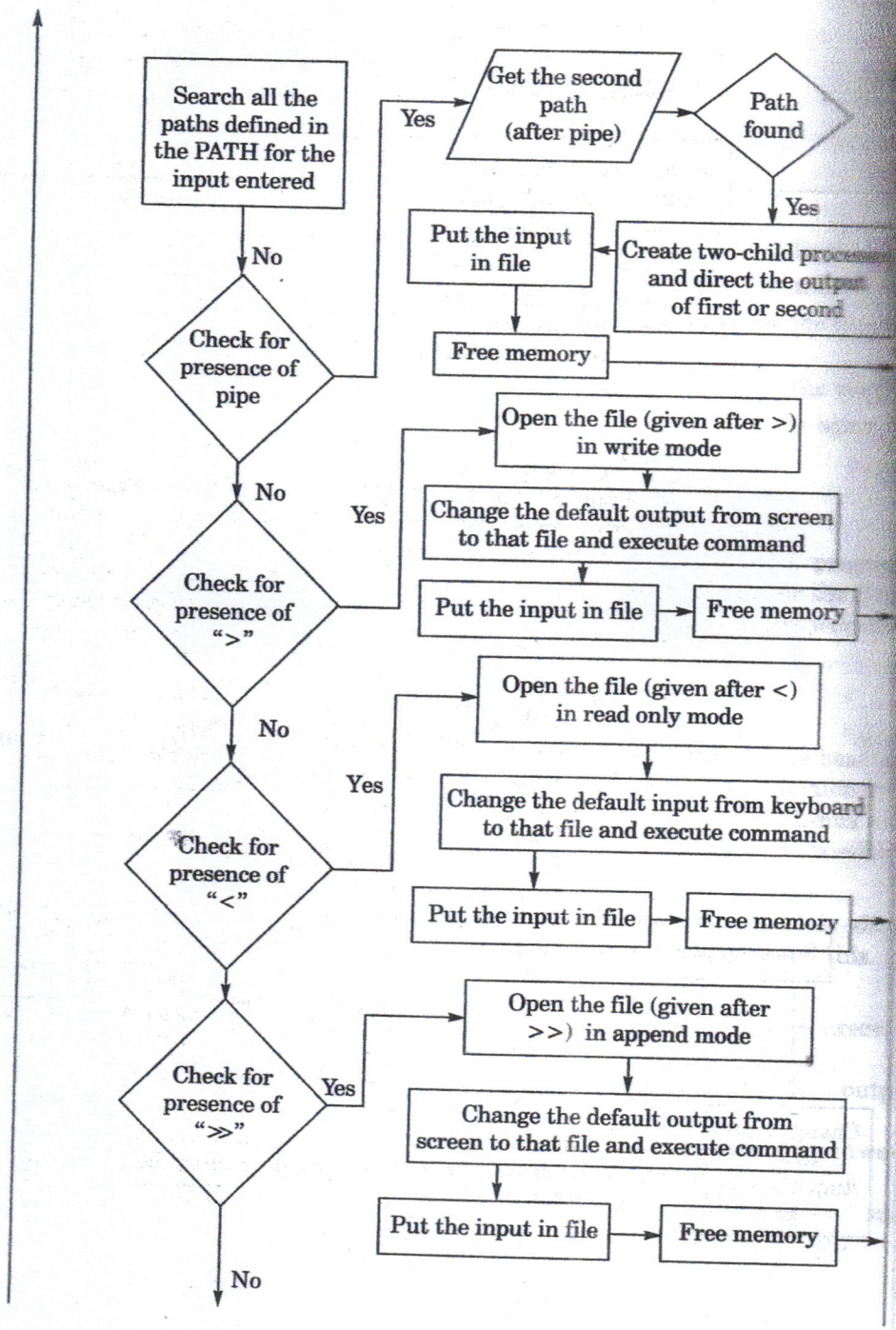

Figure 28.7 Contd.

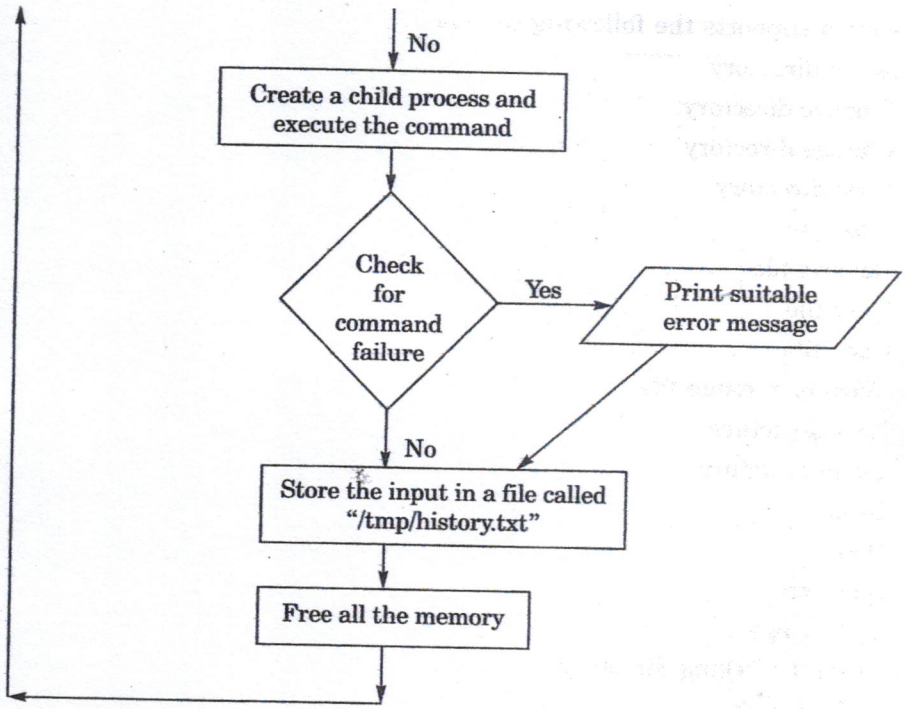

Figure 28.7

CASE STUDY 5: INODE BASED FILE SYSTEM[7]

Abstract

The project deals with the simulation of file systems in UNIX.

Default block size that this system supports is 32 bytes. For simulation and proof of concept realisation, only two data blocks are taken (in UNIX inodes it is 10 data blocks + indirection blocks).

The system supports directory hierarchies.

The operations that are to be supported on the file system as per specification includes:

1. Make directory
2. Remove directory
3. Create file
4. Delete file
5. View file contents
6. View all files in a folder (with properties like inode no., size, owner)
7. Copy file contents to a different file
8. Traversing through the directory hierarchy

[7]Student Project Team: Venkata Thippanna Rao. P.R., Venu Madhav, A., and Vidvadhar. N