CS551: Operating System: Design & Implementation.

Project 1: Sub-Shell of a shell in Minix.

Design Document

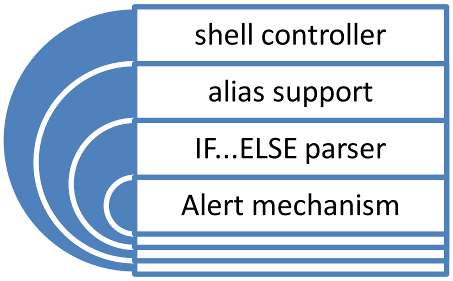
Sunny Changediya (A20353568)

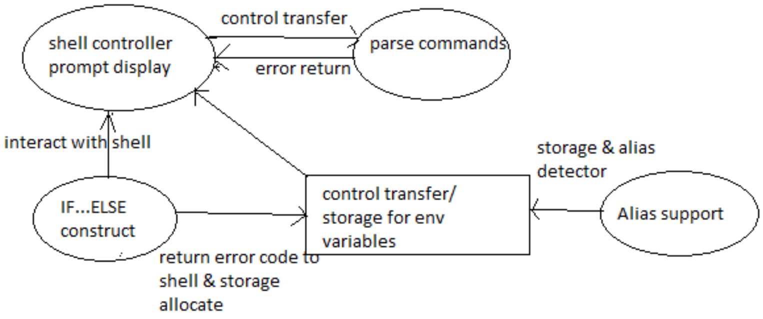
Parth Desai (A2035)

Contents:

Building Blocks and flow diagram:

There are four main building blocks for New Shell.




shell controller prompt display
control transfer
parse commands
error return
interact with shell
IF...ELSE construct
return error code to shell & storage allocate
control transfer/ storage for env variables
storage & alias detector
Alias support

There are total 4 sub systems in this application that have interaction/abstraction with each other.

- **Shell controller:** This is responsible for reading profile file and controlling signal handling along with prompt displaying.
  It also parses the commands and redirects them to in-built shell command binaries. It controls basic shell commands.
- **Alias Support:** This is partially depends on shell controller. It detects the alias general construct and then parses them into global construct so that aliases will be available for period of shell.
- **IF…ELSE Block:** This is responsible for IF…ELSE construct support in shell. It is completely independent module and supports basic IF…ELSE looping structures.
- **Alert mechanism:** This is a small new feature shell supports. If command takes more than 5 sec to execute then alarm will invoke user to abort or continue with execution.

Interaction Mechanism:

We have kept our design as singular, but we have made it modular as per requirement. Each module is sharing some common functions of shell while rests of the modules are separate. Each module is using some kind of similar functions and making it memory reliable construct. We are also freeing memory at the end so all constructs will be freed at same time.

Alias construct limitation:

Alias can be used for large number of keywords.

1. Single normal command like **$$> alias ls2='ls –la'.**
2. Cross aliasing also supported & shows original result. It is able to parse up to any level of aliases in the command where multiple aliases will not be accepted and
   - ✓ Ex: alias ls1='ls –lrt'
   - ✓ Ex: alias ls='ls1'. It will show result of ls –lrt as supported in original shell terminals.Alias structure alone is responsible for looking up all the information related to command and fill in command aliasing structure.

IF…ELSE Support:

- It supports IF…ELSE structure very efficiently. Error checks are added.
- The structure supported is " IF expr=value;then echo msg1;else echo msg2;". Later support for reliability can be added as spaces in between can encounter error in structure.

- It supports basic operators like "=", ">" & "<" only. We have eliminated operators like "-eq, -gt etc."

Application Limitation:

As for now we are supporting alias, IF..ELSE, basic commands & In-built commands. Support for IF..Else variable replacement can be added later. Also Aliases remove & unaliasing can be added in later development. Much global memory allocation has been performed. This can be reduced by implementing stable data structures like linked-list, files and in-memory allocation.

Exception handling:

- Shell controller doesn't find profile file, it will exit at startup only by printing error message on console.
- Shell controller doesn't find home directory, it will exit at start with error message on console.
- If PATH not found, shell will error on command not exists.
- General IF…ELSE error handling has been performed. Whether IF…ELSE structure present or not. If not then shell won't go and check for everything. On start only it will prompt for IF>>>ELSE structure.
- Ctl +C, exit signal/command will be handled and shell will exit with all resources cleanup.
- Automatic memory free done on each command execution so next command will take fresh start.