**Design Document**

*Xiaowei An ( A )*

*Ambalkar Sachin( A )*

*Ying ma (A20313747)*

1. **Introduction:**

The name of our shell is Myshell. Myshell provides and interactive shell that lets you use the programs provided with the Minix system. Besides, Myshell can support the execution of list of commands in a order and also support integer variables. We implement a calculator application that can use the variables as input and output.

1. **Architecture:**

|  |  |  |
| --- | --- | --- |
| **User level** | Level 3 |  |
| **Command interpreter service** | Level 2 | **Exception handling service** |
| **Shell service** | Level 1 |
| **Process service** | Level 0 |

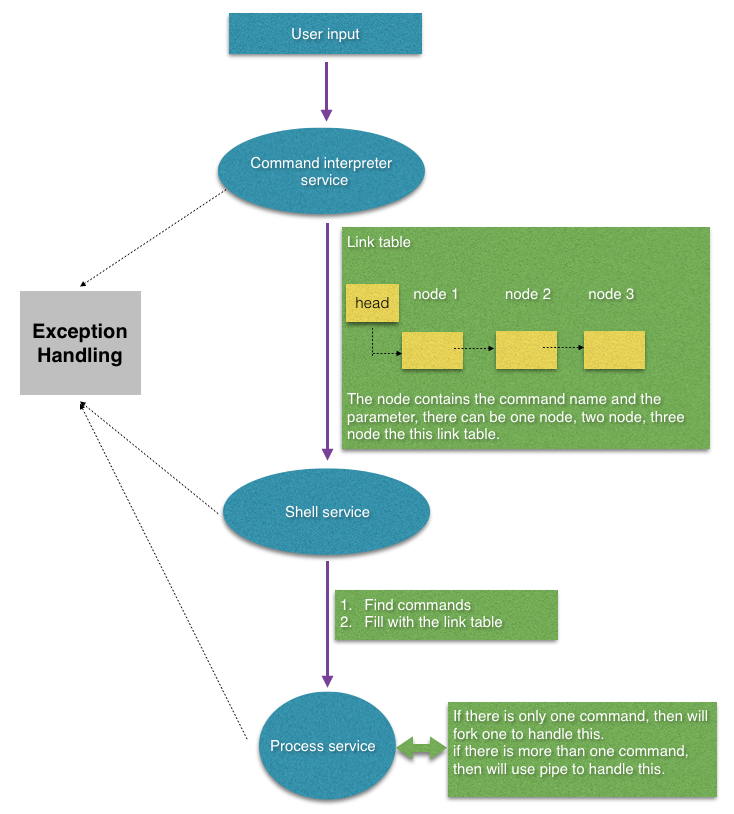
We divide our project into 4 levels and another module called Exception handling service. The Level 2, Level 1, Level 0 can call the Exception handling service any time to deal with the error occurs.

For the user level (Level 3), you can input the commands you want to deal with.

For the Command interpreter service (Level 2), we can process the input you passed in and offer a interface to the next level. If there is any error occur in this level, it will call the exception handling service.

For the Shell service level (Level 1), we add some data structure to the commands you input and then pass to the next level. Besides, the Shell service level also check the commands you input is exist or not if it doesn’t exist, it will pass to the exception handling service.

For the Process service level, we process the data passed from Shell service level and render the answer to the user.

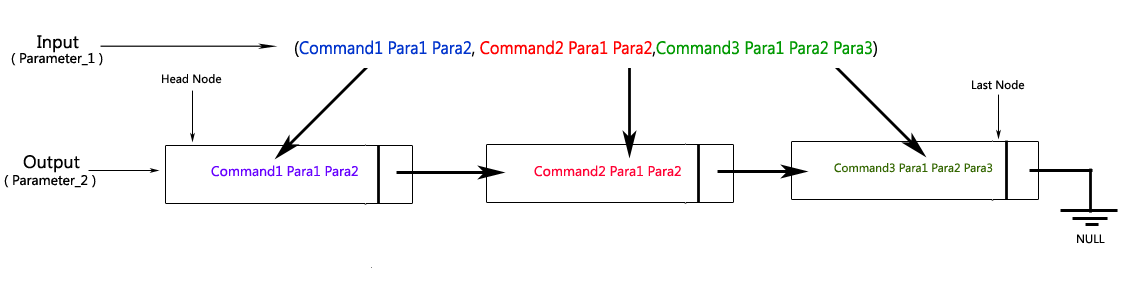


Explained two main functions from ‘ interpretor\_services.h’ file.

Function Details:

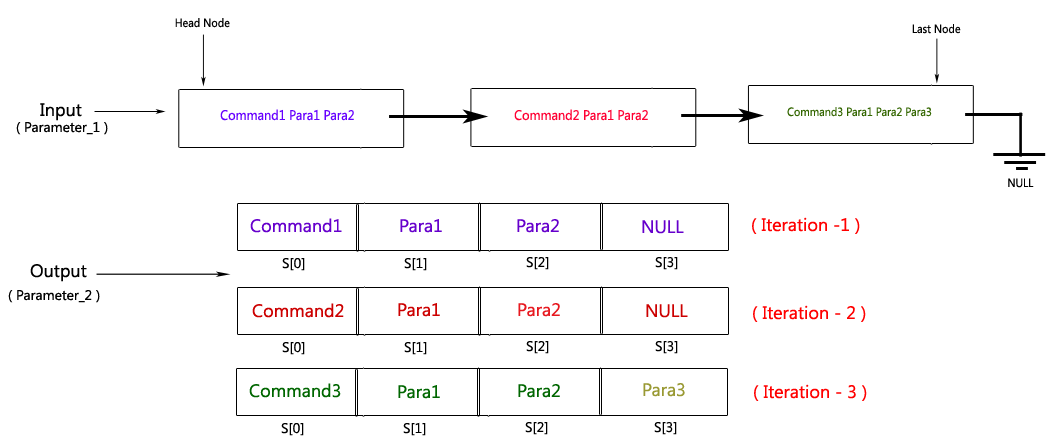
1. extern void **get\_command\_link** ( **char input[COMMAND\_SIZE]** , **ptr \*head** )
2. Parameter\_1 : char input[COMMAND\_SIZE]
3. Parameter\_2 : ptr \*head

Command interpreter is implemented in this function. Parameter\_1 is providing user entered data as input, and all interpreted values stored in Parameter\_2.  
Following Diagram gives pictorial view for the function:



1. extern void **get\_command** ( **ptr \*head**, **char\*s[]** , **char\*path** )
2. Parameter\_1: ptr \*head
3. Parameter\_2: char\*s[]
4. Parameter\_3: char\*path

This function gives interpreted data in array format. It separates command and parameter from string and saves in array named S[]. Output of this function will serves as input for system command.



1. **Exceptional handling:**

Every level will have different error occurs. If the error occurs, then they will call the same interface which supplied by the exceptional handling module. Although the name of the function is same but the exceptional handling module will distinguish these error by the different parameters passed in from different level.When the Exceptional handler is called, it will output some error information and it will use the function pointer to return to the original status to avoid the crash.

1. **Conclusion:**

In this project, we have implement a shell and we know some basic operation. This project has given us a thorough understanding of the structure a shell application.