# OPERATING SYSTEM LAB (COM -312)

## Simulating Linux Terminal File Explorer functionality using special keys, basic commands, and open files features.


**CSE, MODEL INSTITUTE OF ENGINEERING AND TECHNOLOGY**

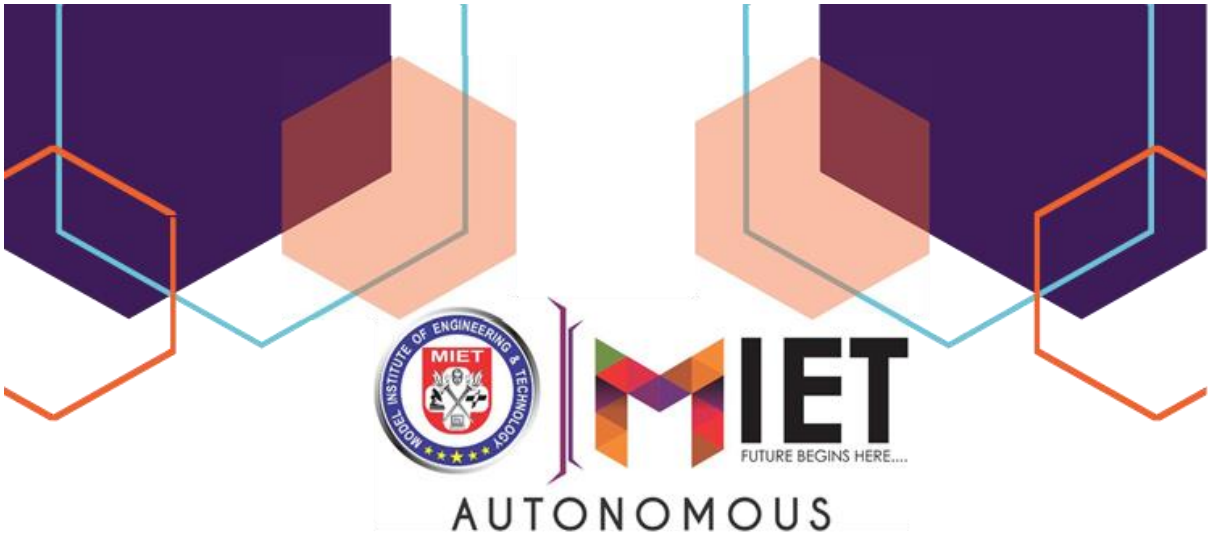
**BACHELOR OF ENGINEERING**

**In**

**Computer Science & Engineering**

**SUBMITTED BY**

**Ankush Raina (2021A1R059)**
**Aditya Zalpuri (2021A1R057)**
**Shradha Sharma (2022A1L016)**
**Kanav Sharma (2021A1T055)**
**Sparsh Sharma (2021AR056)**


SUBMITTED TO

Department of Computer Science & Engineering
Model Institute of Engineering and Technology (Autonomous)
Jammu, India

# ACKNOWLEDGEMENT

Under CRIE (Center for Research, Innovation and Entrepreneurship) we both worked under the guidance of mentors of CRIE, who guided us through a way leading to professionalism and practical hands-on work experience. It is indeed with a great pleasure and immense sense of gratitude that we acknowledge the help of these individuals. We are highly indebted to our Director Ankur Gupta, "MODEL INSTITUTE OF ENGINEERING AND TECHNOLOGY", for the facilities provided to accomplish this main project. We would like to thank our Prof.Ashok Kumar, Head of the Department of Computer Science and Engineering, MIET. For this constructive criticism throughout our project. We feel elated in manifesting our sense of gratitude for our internal project guide. Asst. Saurabh Sharma, Department of Computer Science and Engineering, MIET.He has been a constant source of inspiration for us, and we are very deeply thankful to him.

# FACULTY AND MEMBER

Dr. Ankur Gupta - **He is the Director at the ModeI Institute of Engineering and Technology, Jammu, India, besides being a Professor in the Department of Computer Science and Engineering. Prior to joining academia, he worked as Technical Team Lead at Hewlett Packard, developing software in the network management and e-Commerce domains. He has 2 patents to his name and 20 patents pending. He obtained his B.E, Hons. He is a senior member of the ACM, senior member IEEE and a life-member of the Computer Society of India. He has received competitive grants over Rs 2 crores from various funding agencies and faculty awards from IBM and EMC.**

Prof. Ashok Kumar - **He has over 21 years of experience in industry, academics, research and academic administration. He did his Doctorate from IKG Punjab Technical University, Jalandhar; Master of Engineering from Punjab Engineering College, Chandigarh and Bachelor's degree from NIT, Calicut, Kerala. He is Fellow of Institution of Electronics and Telecommunication Engineers (IETE). His research interest areas are Optical Networks, Electronics Product Design, Wireless Sensor Networks and Digital Signal Processing... He is reviewer of many reputed national and international journals. He has also coordinated many national and international conferences.**

Mr. Saurabh Sharma - **He is working as Assistant Professor in the Department of Computer Science at Model Institute of Engineering and Technology, Jammu and has over 10 years of experience in academics and research. He has done his M. Tech CSE in Web Mining from Maharishi Markandeshwar University, Mullana, Ambala (HR) in the year 2011.His research interest areas are Web Mining, Data Mining, Machine Learning, ANN, Pattern Classification and Object Identification. He has, to his credit, 25 research papers published in journals of national and international repute and he has guided 11 M. Tech Dissertations.**

# CONTENTS

# ABSTRACT

The visualization of hierarchies is very important for digital information management and presentation systems. Especially in the context of Personal Information Management, file explorers play a very important role. Currently the most common file explorer visualizations are Windows Explorer and the simple zoomable visualization offered by Microsoft Windows. This work explores the issue of file explorer visualization through a user study based on interviews and an experiment.

It provides a graphical user interface for accessing the file systems. It is also the component of the operating system that presents many users interface items on the screen such as the taskbar and deskto


## 1. Introduction

File Explorer is a file browser which enables us to organise our files or folders in the way we would like it to be organised. It has various functions and as today's technology goes on getting better and better many more functions are being included.

Some of its main functions are as follows:

**1.1** File Management.

**1.2** Create or delete a file.

**1.3** Move or copy a file.

In 1995, Microsoft first released test versions of a shell refresh, named the Shell Technology Preview, and often referred to informally as "New Shell". The update was designed to replace the Windows 3.x Program Manager/File Manager based shell with Windows Explorer. The release provided capabilities quite similar to that of the Windows "Chicago" (codename for Windows 95)shell during its late beta phases, however, was intended to be nothing more than a test release

## 2. Objective

### A. Normal Mode

1. Files should be displayed alphabetically using explorer.
2. Scrolling: 1 file scrolled at a time.
3. Opening files should be done in their default apps.
4. Back and forward implemented the same as we observe in the Linux GUI file explorer app.
5. Assumed: application home should be given while running the program.

### B. Command Mode

1. If changes are made in the current dir (shown currently on terminal), the changes would be updated (on terminal) when the user comes out of the command mode by pressing Special Key.

2. Goto would update the terminal with a new path immediately.
3. All paths would be relative to Application home.

Our aim is to develop a program that displays the file system on the terminal and helps users navigate through it using special keys, open files and execute basic file commands. We must simulate the basic functionality of a Linux terminal using a shell script

## 3. Advantages of File Explorer

**3.1** It centralises important Documents in one place:

Documents are often siloed across desktop computers, laptops, tablets, mobile phones, flash drives, and email inboxes. File explorer systems can bring an organisation's documents together in one place for easy accessibility. Instead of spending precious time trying to find that one document buried in an archived email thread, you can find what you need and get back to work.

**3.2** It reduces Work:

A staggering 83% of employees must recreate pre-existing documents because they can't locate them on their corporate network. While some documents may take just a few minutes to prepare, many require numerous man-hours to complete. Even if the rework time is low per document, that time can quickly add up across documents and employees. File explorer systems help keep documents organised, so your workforce can quickly find what they need and avoid reinventing the wheel.

**3.3** It cuts down on emails:

Embracing file explorer has cut down on our email volume, which has improved efficiency and helped us avoid frustrations.

## 4. Disadvantages of File Explorer

**4.1** Data Redundancy:

Often, within an organization, files and applications are created by different programmers from various departments over extended periods of time. This can lead to data redundancy, a situation that occurs in a database when a field needs to be updated in more than one table.

**4.2** Inconsistency:

Because of data redundancy this often leads to data inconsistency. Which means that the same copies of data located in various places contain different values. For preventing this, there should be paper listing among different files.

**4.3** Accessibility:

Accessing data in file explorer system is not an effortless process. It is not convenient as it should be. Whenever a user needs to access an information using different approaches, they must execute a special program

# 5. Flowcharts

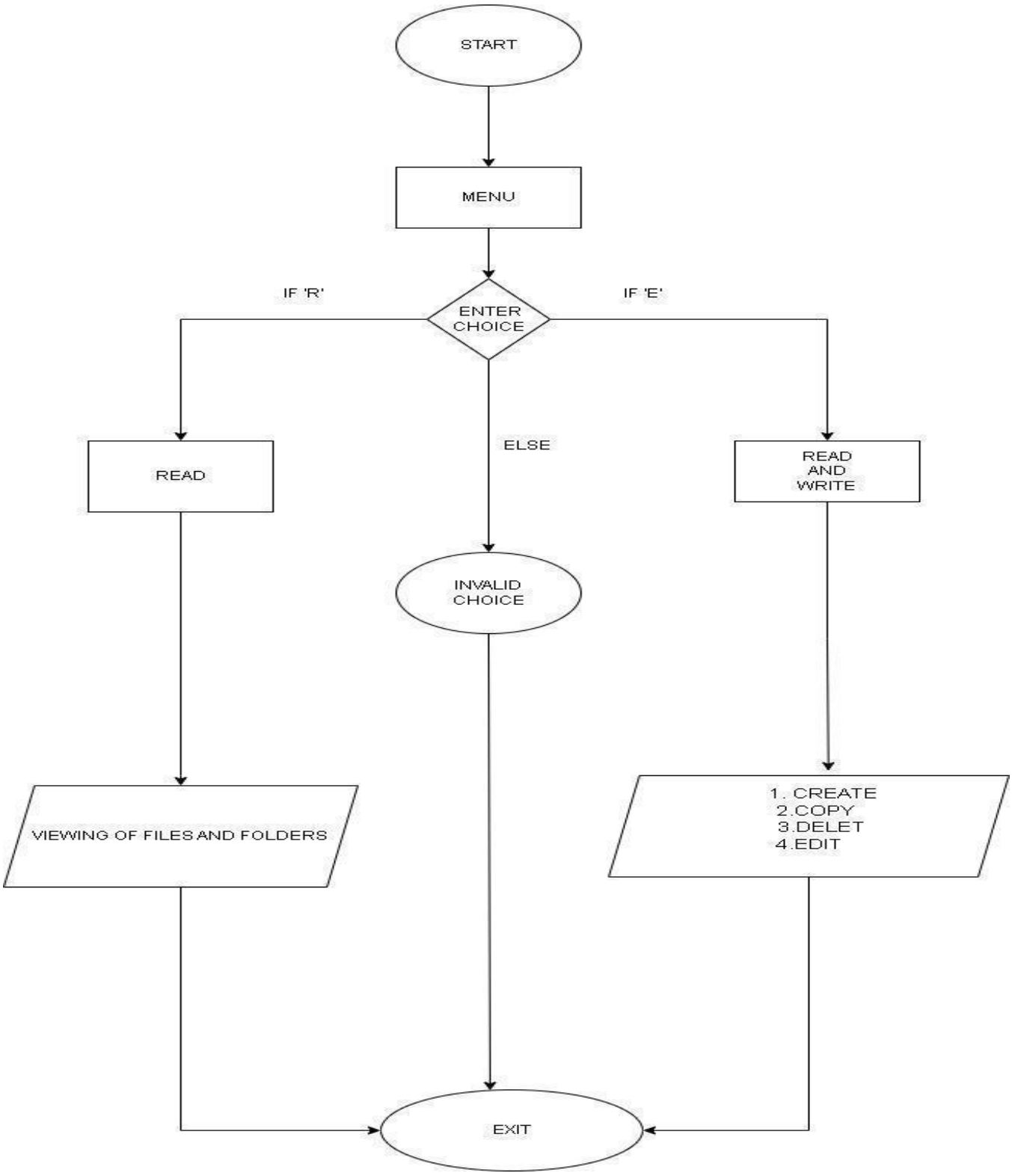



**Fig.A**

## 6. WORKING

- **The File Explorer displays:**

1. **File/Folder**

- **Provides Functions Like:**

1. **Delete**
2. **Display**
3. **Edit The File**
4. **Move To the Directory**
5. **Scroll Up and Do**

# 7. CODE

# NORMAL MODE

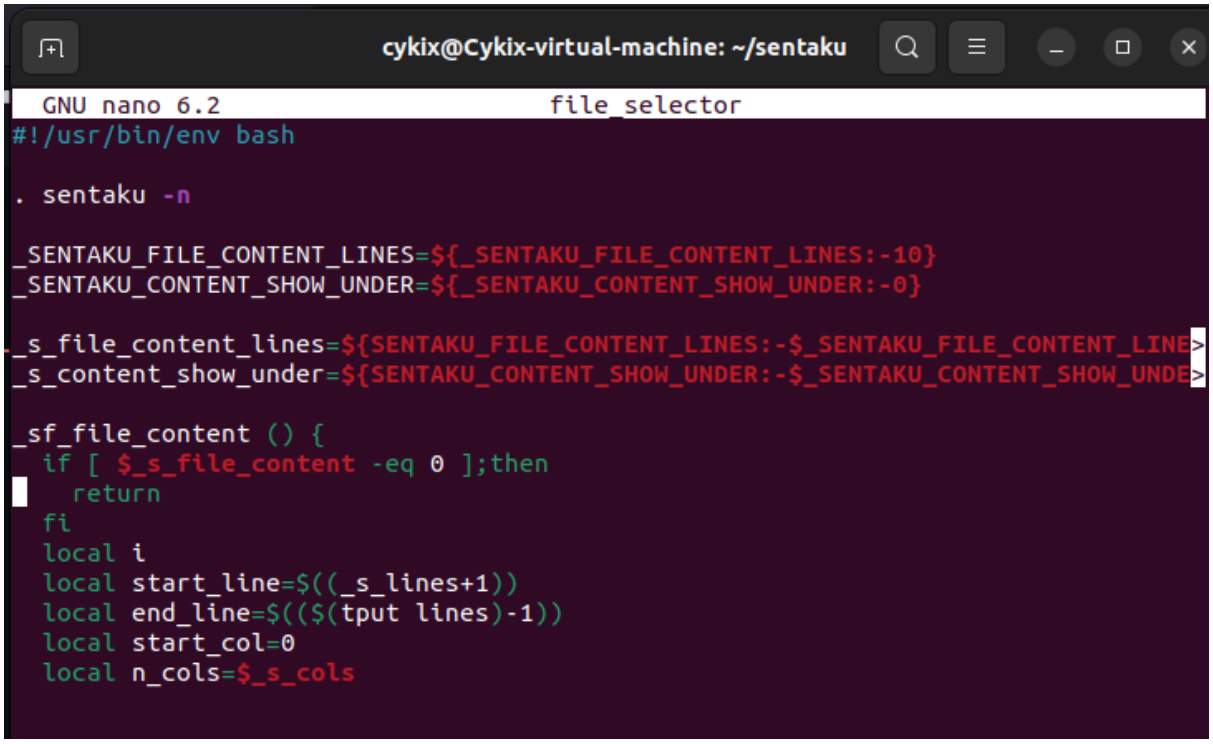



Fig 1:

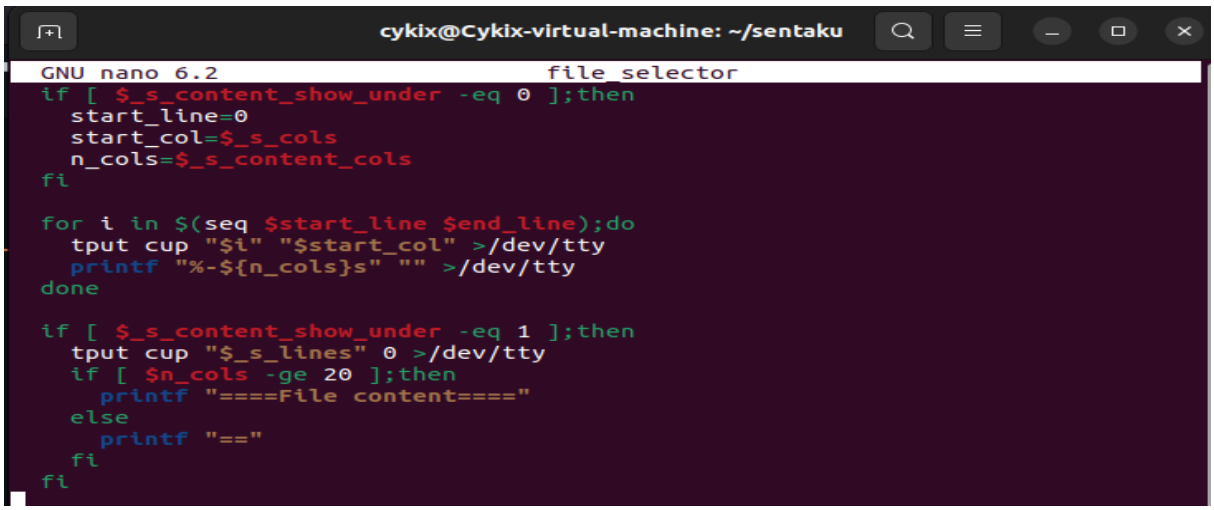



Fig 2:

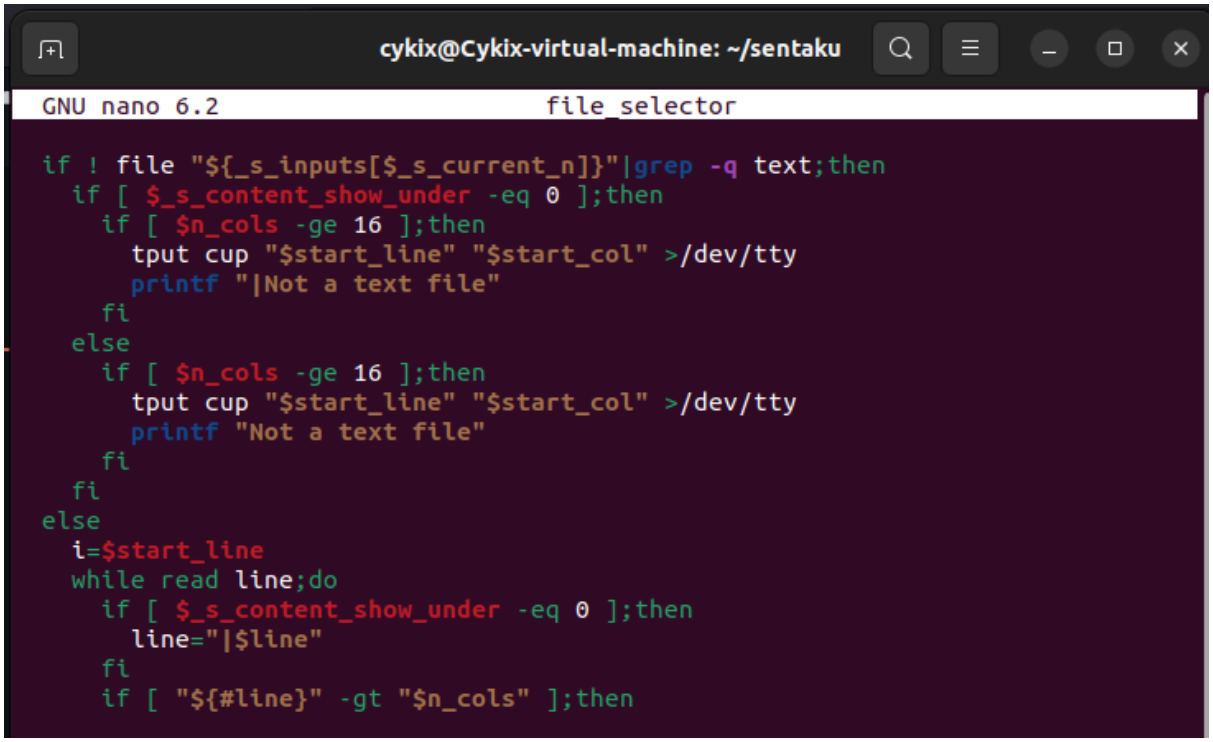

cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
if ! file "${_s_inputs[$_s_current_n]}"|grep -q text;then
if [ $_s_content_show_under -eq 0 ];then
if [ $n_cols -ge 16 ];then
tput cup "$start_line" "$start_col" >/dev/tty
printf "|Not a text file"
fi
else
if [ $n_cols -ge 16 ];then
tput cup "$start_line" "$start_col" >/dev/tty
printf "Not a text file"
fi
fi
else
i=$start_line
while read line;do
if [ $_s_content_show_under -eq 0 ];then
line="|$line"
fi
if [ "${#line}" -gt "$n_cols" ];then

Fig 3:

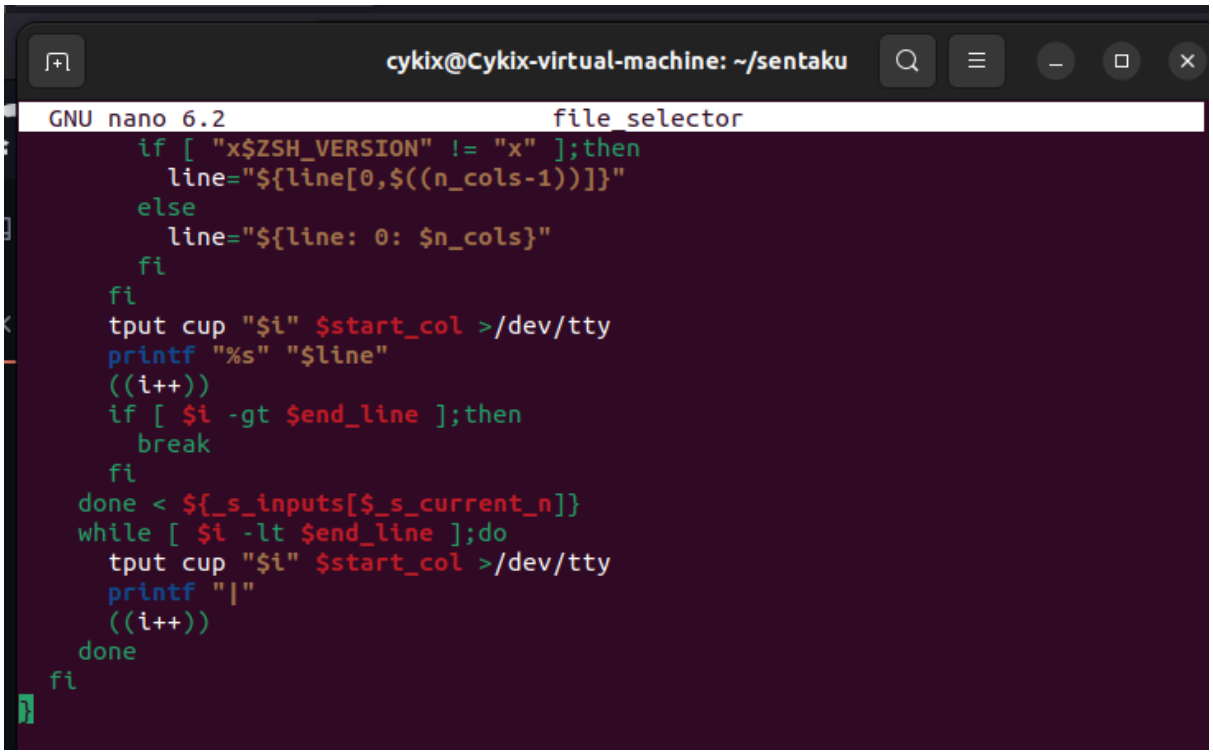

cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
if [ "x$ZSH_VERSION" != "x" ];then
line="${line[0,$((n_cols-1))]}"
else
line="${line: 0: $n_cols}"
fi
fi
tput cup "$i" $start_col >/dev/tty
printf "%s" "$line"
((i++))
if [ $i -gt $end_line ];then
break
fi
done < ${_s_inputs[$_s_current_n]}
while [ $i -lt $end_line ];do
tput cup "$i" $start_col >/dev/tty
printf "|"
((i++))
done
fi
}

Fig 4:

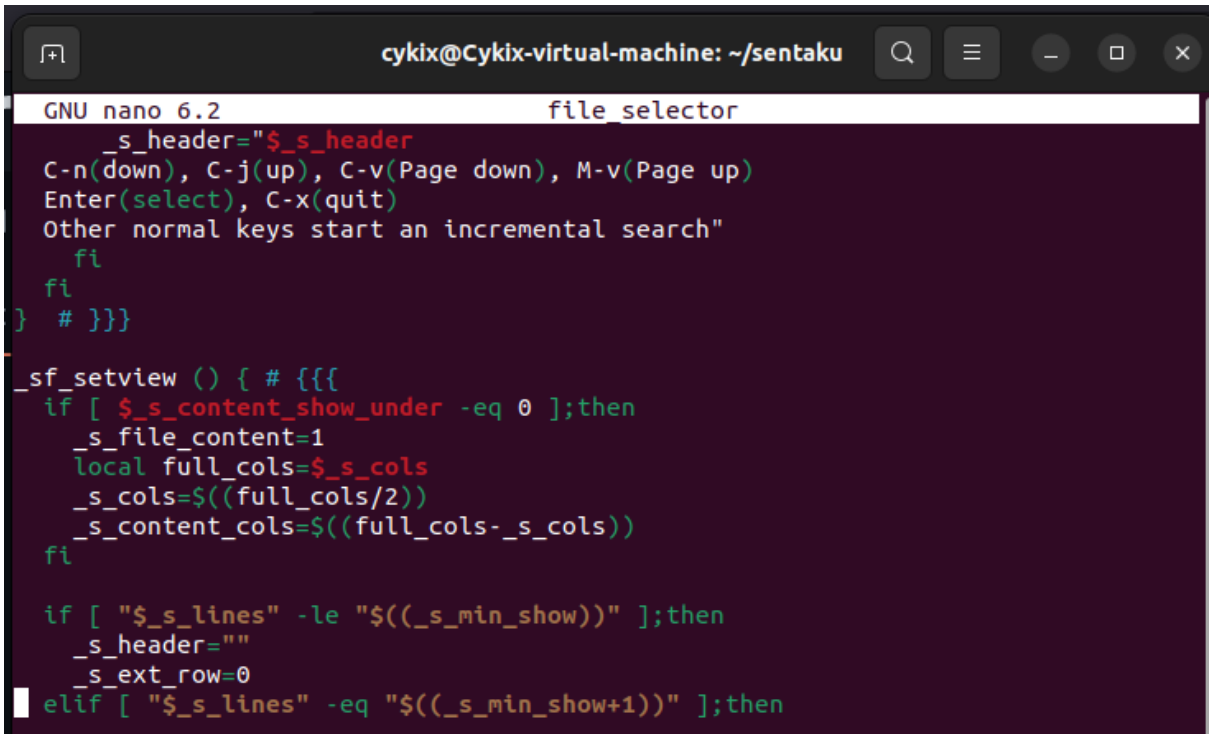

cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
_s_header="$_s_header
C-n(down), C-j(up), C-v(Page down), M-v(Page up)
Enter(select), C-x(quit)
Other normal keys start an incremental search"
fi
fi
} # }}}
_sf_setview () { # {{{
if [ $_s_content_show_under -eq 0 ];then
_s_file_content=1
local full_cols=$_s_cols
_s_cols=$((full_cols/2))
_s_content_cols=$((full_cols-_s_cols))
fi
if [ "$_s_lines" -le "$((_s_min_show))" ];then
_s_header=""
_s_ext_row=0
elif [ "$_s_lines" -eq "$((_s_min_show+1))" ];then

Fig 5:

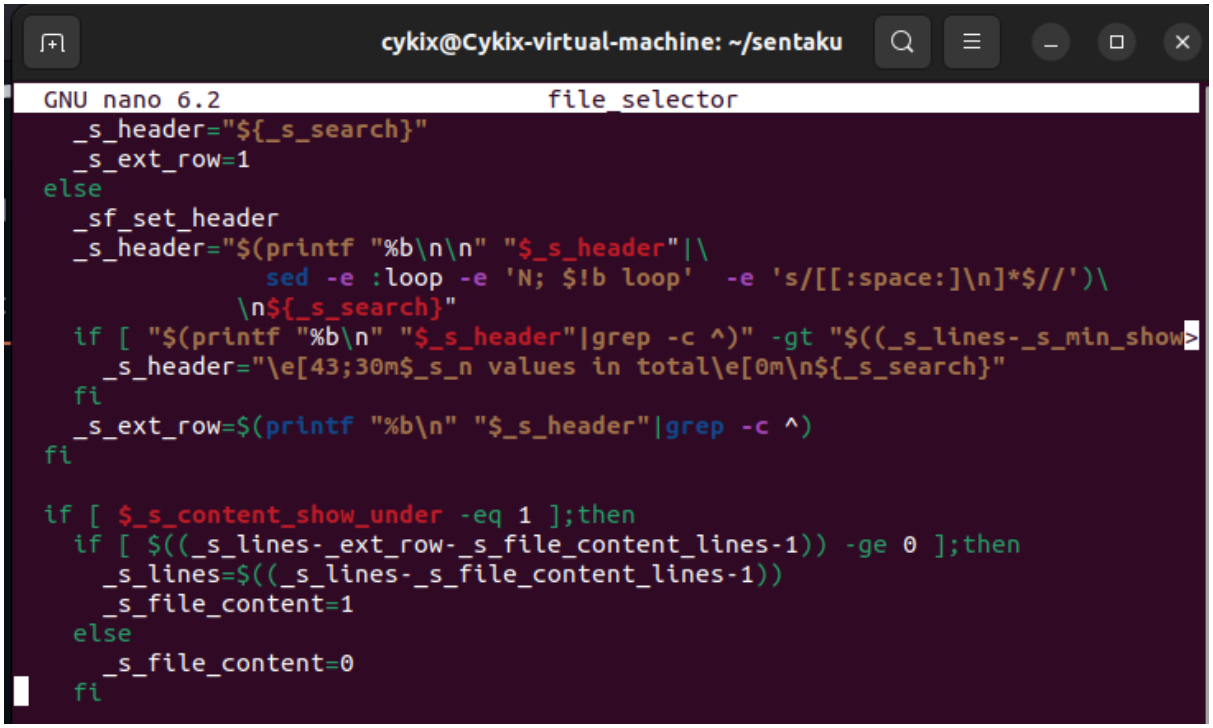

cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
_s_header="${_s_search}"
_s_ext_row=1
else
_sf_set_header
_s_header="$(printf "%b\n\n" "$_s_header"|\
sed -e :loop -e 'N; $!b loop' -e 's/[[:space:]\n]*$//')\
\n${_s_search}"
if [ "$(printf "%b\n" "$_s_header"|grep -c ^)" -gt "$((_s_lines-_s_min_show>
_s_header="\e[43;30m$_s_n values in total\e[0m\n${_s_search}"
fi
_s_ext_row=$(printf "%b\n" "$_s_header"|grep -c ^)
fi
if [ $_s_content_show_under -eq 1 ];then
if [ $((_s_lines-_ext_row-_s_file_content_lines-1)) -ge 0 ];then
_s_lines=$((_s_lines-_s_file_content_lines-1))
_s_file_content=1
else
_s_file_content=0
fi

Fig 6:

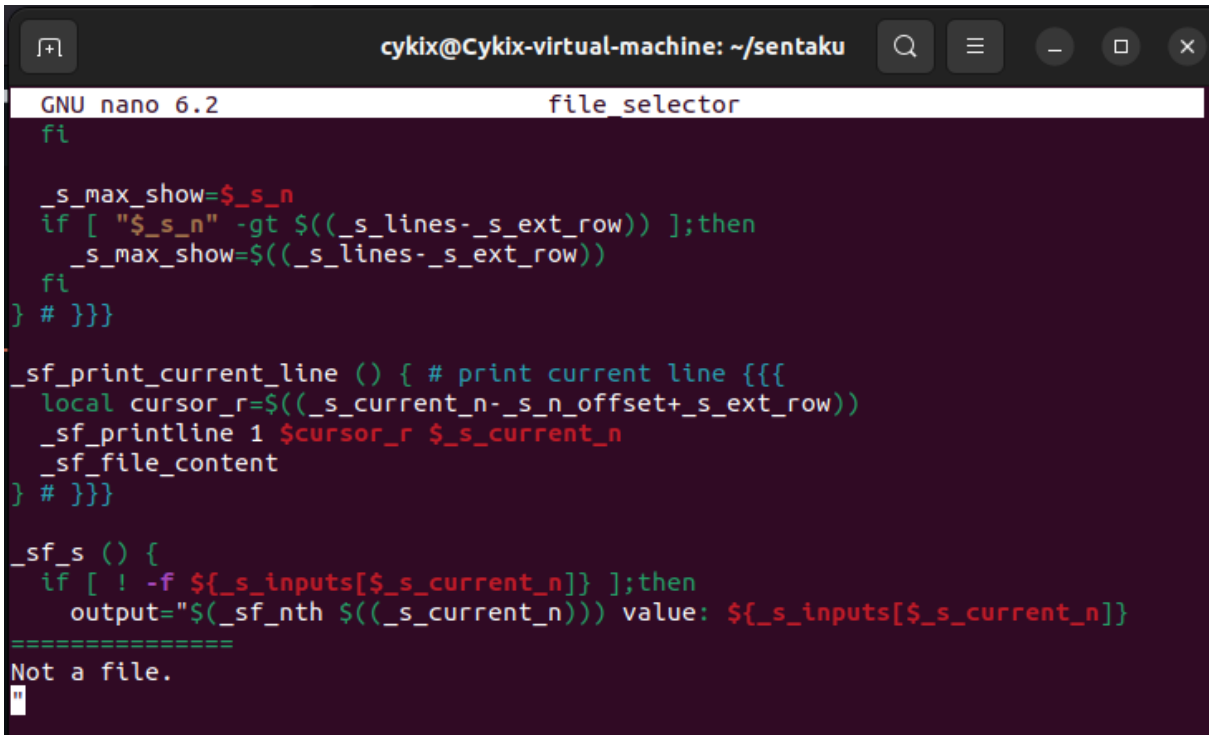
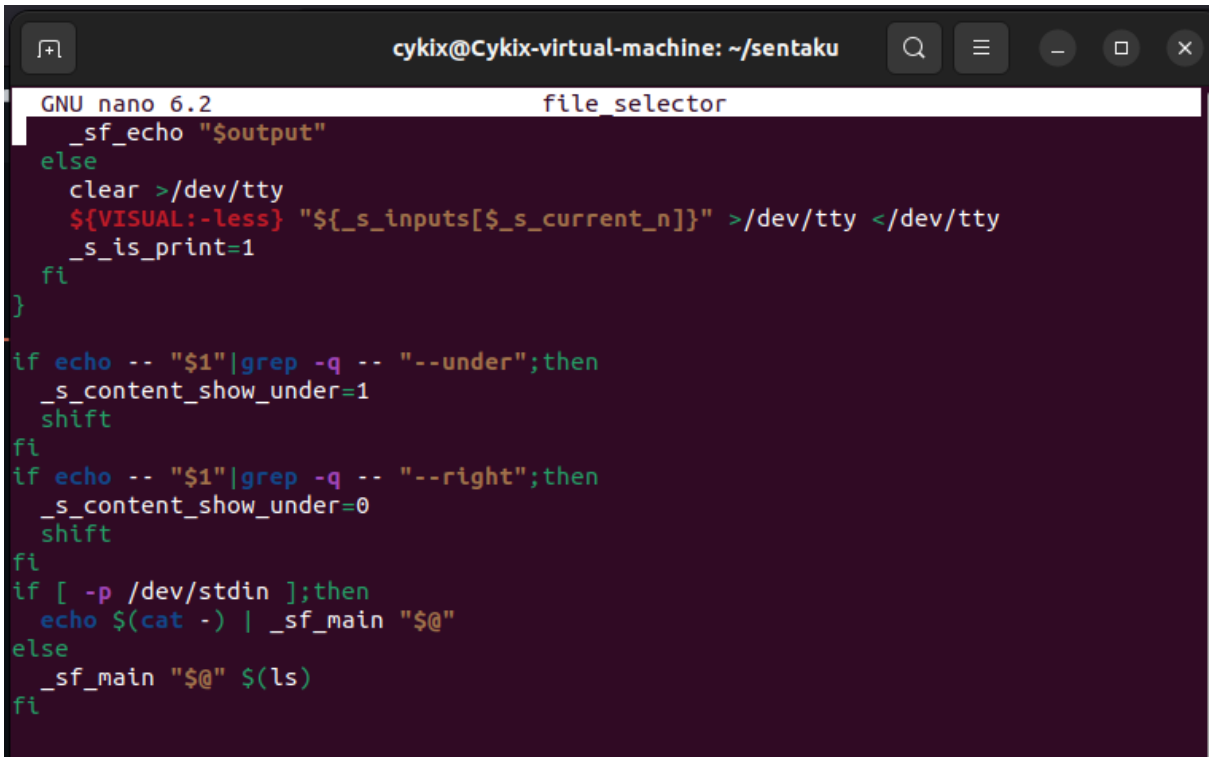

cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
fi
_s_max_show=$_s_n
if [ "$_s_n" -gt $((_s_lines-_s_ext_row)) ];then
_s_max_show=$((_s_lines-_s_ext_row))
fi
} # }}}
_sf_print_current_line () { # print current line {{{
local cursor_r=$((_s_current_n-_s_n_offset+_s_ext_row))
_sf_printline 1 $cursor_r $_s_current_n
_sf_file_content
} # }}}
_sf_s () {
if [ ! -f ${_s_inputs[$_s_current_n]} ];then
output="$(_sf_nth $((_s_current_n))) value: ${_s_inputs[$_s_current_n]}
==========
Not a file.
"

Fig 7:


cykix@Cykix-virtual-machine: ~/sentaku
GNU nano 6.2 file_selector
_sf_echo "$output"
else
clear >/dev/tty
${VISUAL:-less} "${_s_inputs[$_s_current_n]}" >/dev/tty </dev/tty
_s_is_print=1
fi
}
if echo -- "$1"|grep -q -- "--under";then
_s_content_show_under=1
shift
fi
if echo -- "$1"|grep -q -- "--right";then
_s_content_show_under=0
shift
fi
if [ -p /dev/stdin ];then
echo $(cat -) | _sf_main "$@"
else
_sf_main "$@" $(ls)
fi

# COMMAND MODE

```
#!/usr/bin/env bash

# Example of explorer
. sentaku -n

_SENTAKU_SEPARATOR=$'\n'
_SENTAKU_EDITOR=""

# New variable
_s_a=0

# New help
_s_help="
Usage: ex_explorer.sh [-aHNl] [-f <file>] [-s <sep>]

Arguments:
   -a          Show hidden files/directories.
   -H          Header is shown at sentaku window.
   -N          No nubmers are shown.
   -l          Show last words instead of starting words for longer lines.
   -h          Print this HELP and exit.
"

_sf_get_values () { # {{{
  # Get variables
  local orig_ifs=$IFS
  IFS="$_s_s"
  if [ $_s_a -eq 1 ];then
    _s_inputs=($(ls -a))
  else
    _s_inputs=(".." $(ls))
  fi
  IFS=$orig_ifs
  _s_n=${#_s_inputs[@]}
} # }}}

_sf_printline () { # useage: _sf_printline is_selected n_line n_input {{{
  local show=${_s_inputs[$3]}
  tput cup $2 0 >/dev/tty
  if [ $1 -eq 1 ];then
    printf "\e[7m" >/dev/tty
  fi
  if [ -d "$show" ];then
    printf "\e[33;1m" >/dev/tty
```

Fig 9:

```
  fi
  local n_show=$_s_cols
  local num=""
  if [ $_s_nonumber -eq 0 ];then
    n_show=$((_s_cols-5))
    num=$(printf "%3d: " $3)
  fi
  if [ ${#show} -gt $n_show ];then
    if [ $_s_showlast -eq 0 ];then
      if [ "$ZSH_NAME" = "zsh" ];then # need for zsh version < 5
        printf "$num${show[0,$((n_show-1))]}" >/dev/tty
      else
        printf "$num${show: 0: $n_show}" >/dev/tty
      fi
    else
      if [ "$ZSH_NAME" = "zsh" ];then # need for zsh version < 5
        printf "$num${show[$((${#show}-$n_show)),-1]}" >/dev/tty
      else
        printf "$num${show: $((${#show}-$n_show))}" >/dev/tty
      fi
    fi
  else
    printf "$num${show}" >/dev/tty
  fi
  printf "\e[m" >/dev/tty
  tput cup $2 0 >/dev/tty
} # }}}

_sf_execute () { # {{{
  :
} # }}}

_sf_check_args () { # {{{
  # Get arguments
  _s_continue=0
  while [ $# -gt 0 ];do
    case $1 in
      "-a" ) _s_a=1;shift;;
      "-H" ) _s_noheader=1;shift;;
      "-N" ) _s_nonumber=1;shift;;
      "-l" ) _s_showlast=1;shift;;
      "-h" )
        _sf_echo "$_s_help"
        return 0
```

Fig 10:

```
        ;;
      * )
        _sf_echo "$_s_help"
        return 1
        ;;
    esac
  done
  _s_continue=1
  return 0
} # }}}

_sf_finalize_user () { # {{{
  unset _s_a
} # }}}

_sf_select () { # {{{
 if [ -d "${_s_inputs[$_s_current_n]}" ];then
   cd "${_s_inputs[$_s_current_n]}"
   _sf_get_values
   _sf_reset
 else
   _sf_echo "${_s_inputs[$_s_current_n]} is not a directory"
 fi
} # }}}

_sf_set_header () { # {{{
  _s_header=""
  if [ $_s_noheader != 1 -a $_s_lines -gt 10 ];then
    local curdir=$(pwd)
    if [ $((${#curdir}+1)) -gt $_s_cols ];then
      if [ "$ZSH_NAME" = "zsh" ];then
        curdir=${curdir[$((${#curdir}-${_s_cols}+1)),-1]}
      else
        curdir=${curdir: $((${#curdir}-${_s_cols}+1))}
      fi
    fi
    if [ $_s_cols -ge 66 ];then
      _s_header=" $curdir
 [n]j(n-down), [n]k(n-up), gg(top), G(bottom), [n]gg/G, (go to n),
 ^D(Half page down), ^U(Half page up), ^F(Page down), ^B(Page Up),
 s(show detail), d(delete), l(open with less), e(edit the file)
 Enter(select, move to the directory), q(quit)"
    elif [ $_s_cols -ge 42 ];then
      _s_header=" $curdir
```

Fig 11:

```
_sf_d () {
  clear >/dev/tty
  local yes=0
  while : ;do
    echo "Delete ${_s_inputs[$_s_current_n]}?: (y/n)"
    _sf_read
    if [ "$_s_read" = "y" ];then
      yes=1
      break
    elif [ "$_s_read" = "n" ];then
      break
    fi
  done
  if [ $yes -eq 1 ];then
    rm -rf ${_s_inputs[$_s_current_n]}
  fi
  _sf_get_values
}

_sf_s () { # {{{
  _sf_echo $(ls -l "${_s_inputs[$_s_current_n]}")
} # }}}

_sf_l () { # {{{
  clear >/dev/tty
  less ${_s_inputs[$_s_current_n]} >/dev/tty </dev/tty
} # }}}

_sf_e () { # {{{
  local e=${_SENTAKU_EDITOR:-${EDITOR}}
  e=${e:-vim}
  $e ${_s_inputs[$_s_current_n]} >/dev/tty </dev/tty
  _sf_quit
} # }}}

_sf_main "$@"
```

Fig 12:

# 8. IMPLEMENTATION

# Normal Mode(View only)

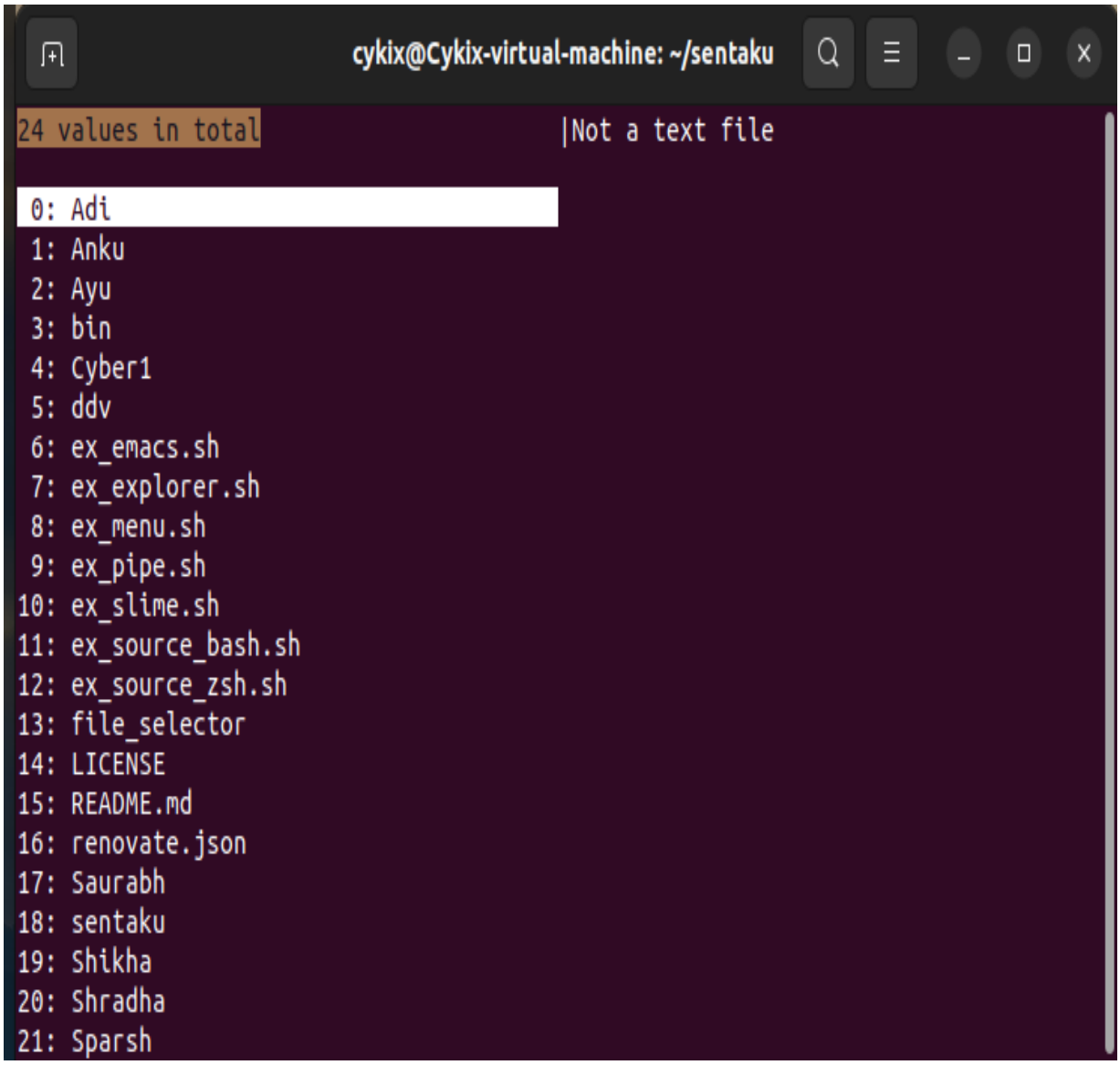



**Fig. 1:**

# COMMAND MODE (EDITING MODE)

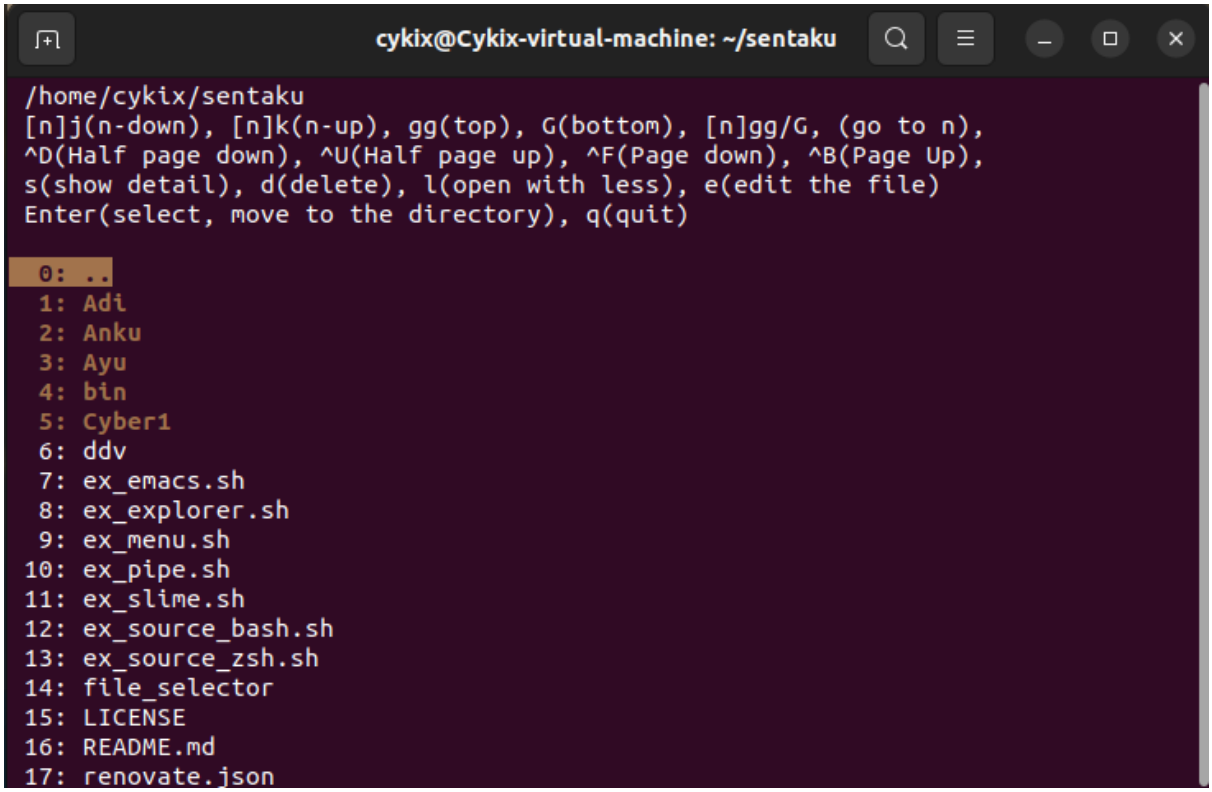



**Fig. 2:**

## 7. References

**6.1** https://www.hitechwhizz.com/2021/02/7-advantages-and-disadvantages-limitations-benefits-of-file-management-system.html

**6.2** https://en.wikipedia.org/wiki/File_Explorer

**6.3** https://github.com/rcmdnk