Lab 4: Screen Captures

Table of Contents

Section 3: Lab Challenge and Analysis

Part 1: Analysis and Discussion

BASH (Bourne-Again Shell):
- **Type**: It is a Unix shell and scripting language.

Key Features:
- **Command Execution**: BASH is primarily a command-line shell, allowing users to execute commands directly.
- **Scripting Language**: BASH also functions as a scripting language, allowing users to write scripts to automate tasks.
- **Variables**: It supports variables for storing and manipulating data.
- **Control Flow**: BASH provides control structures like loops
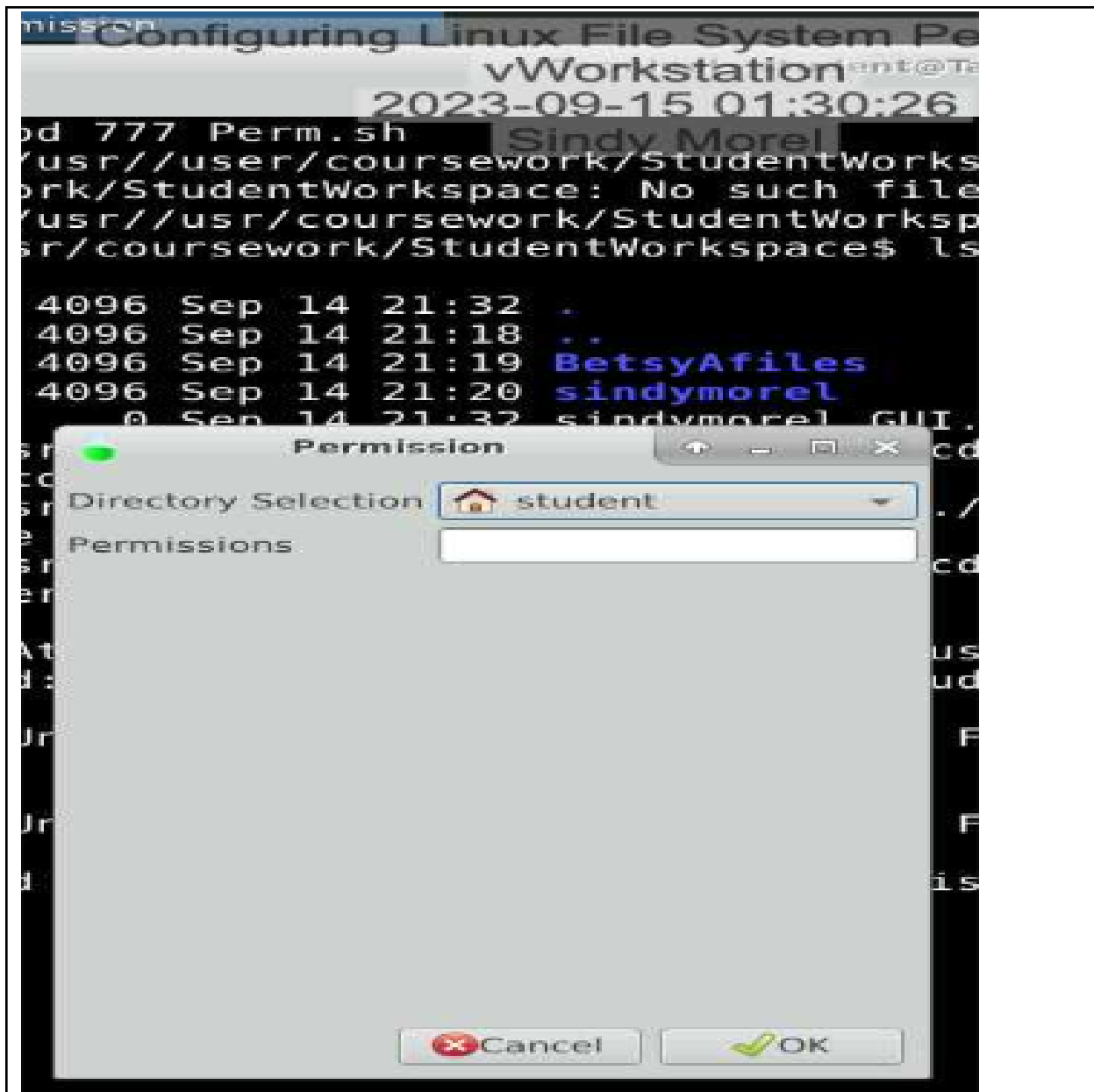
AWK:
- **Type**: It is a Text processing and pattern scanning language.

Key Features:
- **Text Processing**: AWK is designed for text processing and operates on structured data.
- **Pattern Matching**: It allows users to specify patterns to match within the input data.
- **Data Extraction**: AWK is commonly used to extract specific fields or data columns from files.
- **Calculations**: Users can perform mathematical calculations and operations on data.
- **Looping and Conditionals**: It supports loops and conditional statements

YAD (Yet Another Dialog):
- **Type**: GUI (Graphical User Interface) dialog builder for shell scripts.

Key Features:
- **GUI for Shell Scripts**: YAD allows shell script developers to create graphical user interfaces (GUIs) for their scripts.
- **Dialog Boxes**: It provides the ability to create various types of dialog boxes, such as input forms, file selection dialogs, and message boxes.
- **User Interaction**: YAD facilitates user interaction with scripts through buttons, input fields, and other graphical elements.
- **Script Integration**: Developers can incorporate YAD dialogs into their shell scripts to

and conditional statements.

- **Pipe and Redirection**: It supports pipes and redirection for input/output manipulation.
- **Environment Variables**: BASH uses environment variables to store configuration and system information.
- **Job Control**: Allows users to manage multiple processes and jobs.
- **Custom Functions**: Users can define custom functions for code modularity.
- **Extensible**: Users can extend BASH's functionality by creating custom scripts and functions.

for complex data processing.

- **Report Generation**: AWK is often used to generate reports from structured data.
- **Built-in Variables**: It provides built-in variables that hold information about the input data, making data manipulation easier.
- **Custom Functions**: Users can define custom functions to encapsulate reusable logic.

enhance user experience.

- **Cross-Platform**: YAD is primarily used in Linux environments but can be installed on different Unix-like systems.
- **Customization**: Users can customize the appearance and behavior of YAD dialogs.
- **Script Automation**: YAD can be used to automate tasks that require user input or configuration.

These tools, BASH, AWK, and YAD, play important roles in Unix and Linux environments, offering a range of capabilities from command-line interaction and data processing to graphical user interface creation. They are widely used by system administrators and script developers to perform various tasks efficiently.

Part 2: Tools and Commands

Changing –field="Path" \ to –field="Directory Selection":DIR \ changes the requirement of having to type out the path to being able to drop down and select a path instead. The "Path" field will now display a "Browse" button that allows the user to select a directory interactively. This change gives users the option to choose the directory path rather than typing it manually.

Part 3: Challenge Exercise

TargetLinux01 - 172.30.0.11 - Remote Desktop Connection

Terminal - student@Tar...

Configuring Linux File System Permissions
vWorkstation
2023-09-15 02:17:50
Sindy Morel

student@TargetLinux01: ~

File   Edit   View   Terminal   Tabs   Help

```bash
#!/bin/bash
#Create Application Dynamic - Frame Windows and Fields
frmdata=$(yad --title "Permission" --form --width 300 --height 400 \
    --field="Directory Selection":DIR \
    --field="Permissions" \
    --directory \
    )
#Process User Input
frmSelect=$(echo $frmdata | awk 'BEGIN {FS="|"}{ print $1}')
frmPerm=$(echo $frmdata | awk 'BEGIN {FS="|"}{ print $2}')
#Logic Process check for empty variables
# -z = if variable is empty
# -n = If variable is not empty
#
log_file="sindymorel.log"

set -x
if [ -z "$frmSelect" ]; then
    echo "Directory Not Selected"

elif [ -n "$frmSelect" ]; then
    echo "Directory $frmSelect selected"
fi

if [ -z '"$frmPerm" | "$frmSelect"' ]; then
    cecho "User Input Missing Perm: $frmPerm Select: $frmSelect"

elif [ -n '"$frmPerm" | "$frmSelect"' ]; then
    chmod $frmPerm $frmSelect
fi
~
```

Configuring Linux File System Permissions
vWorkstation
Sindy Morel

rk/LecureSlides
f '/usr/usr/coursework/LecureSlides': Operation no
rm.sh

nable to open server bookmarks: Failed to open fil

nable to open serve

without a transier

d'

elect"' ']'
elect"' ']'

nformation.
usr//usr/coursework
r/coursework/Studer

4096 Sep 14 21:32 .
4096 Sep 14 21:18 .
4096 Sep 14 21:19 B
4096 Sep 14 21:20 s
   0 Sep 14 21:32 s
r/coursework/Studer
 or directory
r/coursework/Studer

Permission

Directory Selection 🏠 student ▾

Permissions [   ]

❌ Cancel   ✔ OK

Configuring Linux File System Permissions
vWorkstation
Sindy Morel

rk/LecureSlides
f '/usr/usr/coursework/LecureSlides': Operation no
rm.sh

nable to open server bookmarks: Failed to open fil

nable to open serve

without a transier

d'

elect"' ']'
elect"' ']'

nformation.
usr//usr/coursework
r/coursework/Studer

4096 Sep 14 21:32 .
4096 Sep 14 21:18 .
4096 Sep 14 21:19 B
4096 Sep 14 21:20 s
   0 Sep 14 21:32 s
r/coursework/Studer
 or directory
r/coursework/Studer

Permission

Directory Selection 📁 BetsyAfiles ▾

Permissions [o-rx]

❌ Cancel   ✔ OK

```
+ '[' -n '"$frmPerm" | "$frmSelect"' ']'
+ chmod
chmod: missing operand
Try 'chmod --help' for more information.
student@TargetLinux01:~$ cd /usr//usr/coursework/StudentWorkspace
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$ ls -al
total 16
drwxr-xr-x 4 student student 4096 Sep 14 21:32 .
drwxr-xr-x 5 root    root    4096 Sep 14 21:18 ..
drwxr-xr-x 2 betsya  root    4096 Sep 14 21:19 BetsyAfiles
drwxr-xr-x 2 root    root    4096 Sep 14 21:20 sindymorel
-rw-r--r-- 1 root    root       0 Sep 14 21:32 sindymorel_GUI.sh
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$ ./Perm.sh
bash: ./Perm.sh: No such file or directory
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$ Perm.sh
Perm.sh: command not found
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$ cd
student@TargetLinux01:~$ vi Perm.sh
student@TargetLinux01:~$ ./Perm.sh

(yad:1704): Gtk-WARNING **: Unable to open server bookmarks: Failed to open file '/home/student/.con
rmission denied

(yad:1704): Gtk-WARNING **: Unable to open server bookmarks: Failed to open file '/home/student/.con
rmission denied
Gtk-Message: GtkDialog mapped without a transient parent. This is discouraged.
+ '[' -z /usr/usr/coursework/StudentWorkspace/BetsyAfiles ']'
+ '[' -n /usr/usr/coursework/StudentWorkspace/BetsyAfiles ']'
+ echo 'Directory /usr/usr/coursework/StudentWorkspace/BetsyAfiles selected'
Directory /usr/usr/coursework/StudentWorkspace/BetsyAfiles selected
+ '[' -z '"$frmPerm" | "$frmSelect"' ']'
+ '[' -n '"$frmPerm" | "$frmSelect"' ']'
+ chmod o-rx /usr/usr/coursework/StudentWorkspace/BetsyAfiles
chmod: changing permissions of '/usr/usr/coursework/StudentWorkspace/BetsyAfiles': Operation not per
student@TargetLinux01:~$ cd /usr//usr/coursework/StudentWorkspace
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$ ls -al
total 16
drwxr-xr-x 4 student student 4096 Sep 14 21:32 .
drwxr-xr-x 5 root    root    4096 Sep 14 21:18 ..
drwxr-xr-x 2 betsya  root    4096 Sep 14 21:19 BetsyAfiles
drwxr-xr-x 2 root    root    4096 Sep 14 21:20 sindymorel
-rw-r--r-- 1 root    root       0 Sep 14 21:32 sindymorel_GUI.sh
student@TargetLinux01:/usr/usr/coursework/StudentWorkspace$
```