Week 12

1. Write a shell script to back up a directory to a specified location.

SOURCE\_DIR="src"

BACKUP\_DIR="dest"

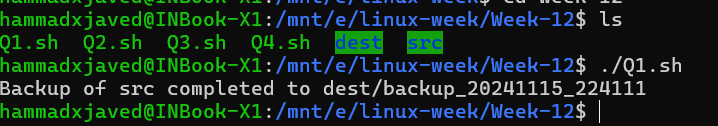
TIMESTAMP=$(date +"%Y%m%d\_%H%M%S")

DEST="$BACKUP\_DIR/backup\_$TIMESTAMP"

mkdir -p "$DEST"

cp -r "$SOURCE\_DIR"/\* "$DEST"

echo "Backup of $SOURCE\_DIR completed to $DEST"



2. Write a shell script to monitor disk usage and send an alert if usage exceeds a

threshold.

THRESHOLD=80

USAGE=$(df / | grep / | awk '{print $5}' | sed 's/%//g')

if [ "$USAGE" -gt "$THRESHOLD" ]; then

    echo "Disk usage is above $THRESHOLD% at ${USAGE}%."

else

    echo "Disk usage is below threshold at ${USAGE}%."

fi



3. Write a shell script to automate the creation of user accounts.

usage() {

    echo "Usage: $0 -u <username> -p <password>"

    exit 1

}

if [ "$(id -u)" -ne 0 ]; then

    echo "This script must be run as root"

    exit 1

fi

while getopts "u:p:" opt; do

    case $opt in

        u) username=$OPTARG ;;

        p) password=$OPTARG ;;

        \*) usage ;;

    esac

done

if [ -z "$username" ] || [ -z "$password" ]; then

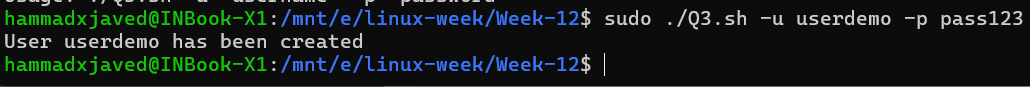
    usage

fi

useradd -m -s /bin/bash "$username"

echo "$username:$password" | chpasswd

echo "User $username has been created"



4. Write a shell script to search for a specific pattern in a file and display the results.

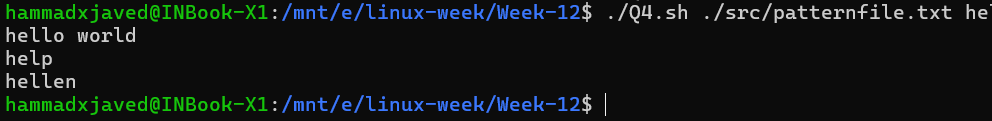
if [ "$#" -ne 2 ]; then

    echo "Usage: $0 <file> <pattern>"

    exit 1

fi

grep "$2" "$1"



5. Consider two features x and y based on the following function:

y = x12 + 3x2 + c, where c can be prepared based on 1000 random values between 0

and 1

Now generate 1000 random values between 0 and 1 for x1 and x2. Calculate y based

on above function. Now train Polynomial Regression model and check the score for

the same.

import numpy as np

from sklearn.preprocessing import PolynomialFeatures

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

x1 = np.random.rand(1000, 1)

x2 = np.random.rand(1000, 1)

c = np.random.rand(1000, 1)

y = (x1 \*\* 2) + (3 \* x2) + c

X = np.hstack((x1, x2))

poly = PolynomialFeatures(degree=2)

X\_poly = poly.fit\_transform(X)

model = LinearRegression()

model.fit(X\_poly, y)

y\_pred = model.predict(X\_poly)

score = r2\_score(y, y\_pred)

print("R^2 Score:", score)

