

Advance Bash Scripting

Background Process

- To Run bash script in a background just add '&' after the script to move the process from foreground to background

```
sleep 30 #it is foreground process
```

```
sleep 30 & #it is background process
```

- this will keep the sleep thread for 30 seconds in a background along with that it will create **JOB ID** and Process ID(**PID**)
- OUTPUT

```
[1] 1234
```

here [1] is the JOB ID and 1234 is PID

- to get the list of all jobs in a background process

```
jobs
```

- to see the running process

```
ps aux | grep sleep
```

- to send process | script to foreground

```
fg %1
```

- here fg -foreground and %1 is the JOB ID
- to send process to background again
- ctr+z

```
bg
```

- to kill the process

- **to terminate a background job using JOB ID**

kill %1

- **here %1 is JOB ID**

RUNNING A PROCESS IN BACKGROUND AND KEEPING IT ALIVE

- **nohup- No Hang Up**
- when we close the terminal process terminated by default , to prevent this

nohup ./script.sh &

- output is saved in nohup.out bt default
- use disown

./script.sh & disown

- this will remove the process from the job table , so it wont be terminated when the session ends

NOTE:

1. **disown has nothing to do with the file cleanup**
2. **nohup.out is just where nohup writes output unless you redirect it manually**

nohup ./script.sh & disown

TASK: Running Background Script:

- create a script that generates Random files and folders, also the script writes the data in the files
 1. STEP:1 use shebang
 2. STEP:2 write your own logic using BASH
 3. STEP:3 make it executable using chmod + <script_name>.sh
 4. STEP:4 run the script with &

Solution:

```
#!/bin/bash

# Number of folders and files to create
FOLDER_COUNT=3
FILES_PER_FOLDER=2

# Base directory
BASE_DIR="./random_data"

# Create base directory
mkdir -p "$BASE_DIR"

# Loop to create folders and files
for ((i = 1; i <= FOLDER_COUNT; i++)); do
    FOLDER_NAME="folder_$RANDOM"
    FULL_PATH="$BASE_DIR/$FOLDER_NAME"

    mkdir -p "$FULL_PATH"
    echo "Created folder: $FULL_PATH"

    for ((j = 1; j <= FILES_PER_FOLDER; j++)); do
        FILE_PATH="$FULL_PATH/file_$j.txt"
        echo "This is file #$j inside $FOLDER_NAME" > "$FILE_PATH"
        echo "Created file: $FILE_PATH"
    done
done
```

AUTOMATIC USER MANAGEMENT

- We can create or use a script to add multiple users, set password and assign permission

```
#!/bin/bash

USERS=("nikunj" "ankita" "shubham" "mahesh" "satish" "aafrin")

for user in "${USERS[@]}"; do

    if id "$user" &> /dev/null; then
        echo "User $user already exists"
    else
```

```
sudo useradd -m -s /bin/bash "$user"
echo "User $user created"
fi

done
```

TASK: create a script that reads the users from a file and check for the user exist or not

1. Reads the user name from a file(user.txt)
2. Create the Users if they don't exist
3. Prints a Simple message for Each

```
#!/bin/bash username_file="users.txt" if [ ! -f "$username_file" ]; then echo
"Error: $username_file not found." exit 1 fi for username in $(cat
"$username_file"); do if id "$username" &>/dev/null; then echo "User $username
already exists." continue fi sudo useradd -m -s /bin/bash "$username" echo
"User $username created." done echo "User creation completed."
```