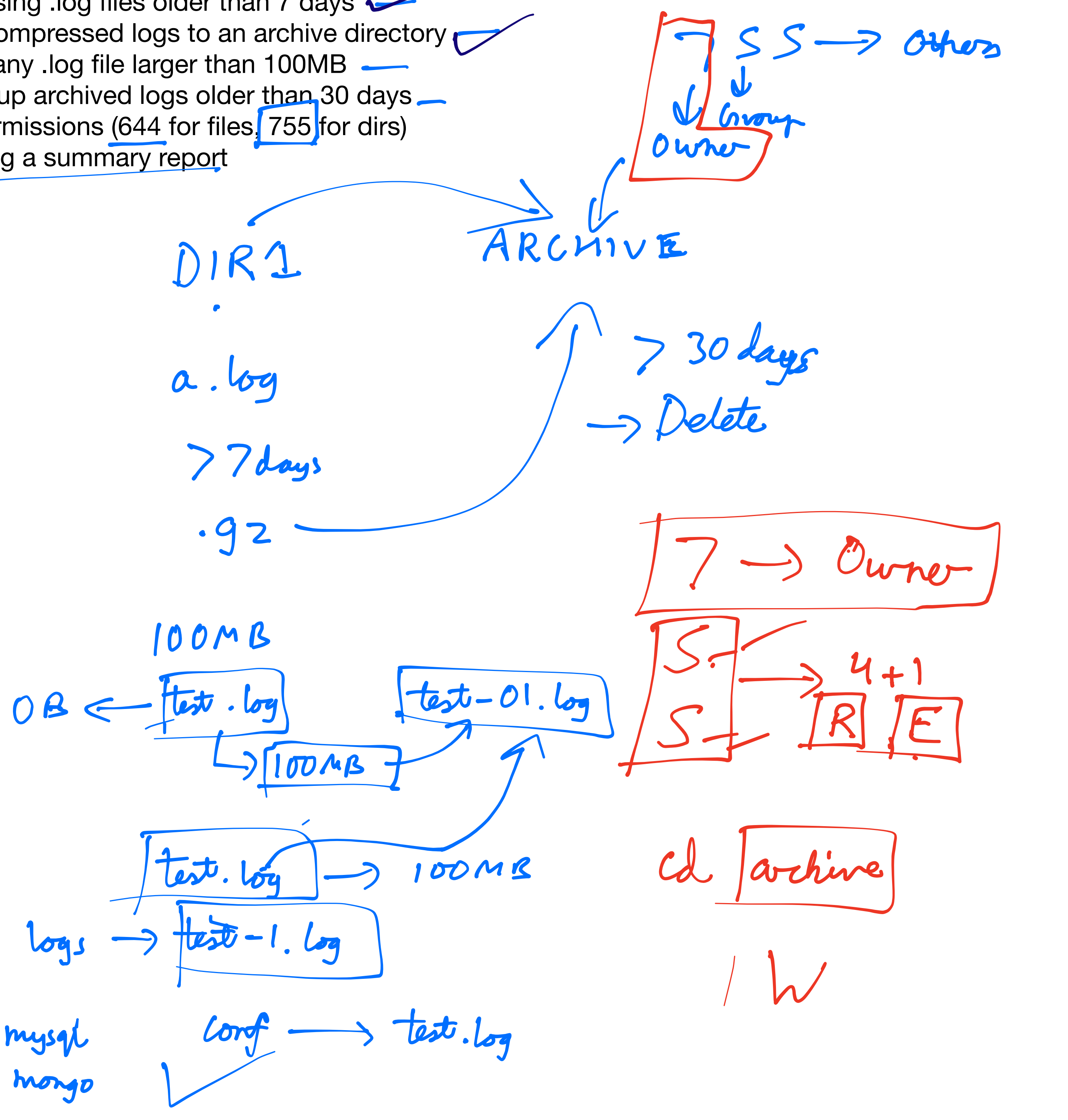
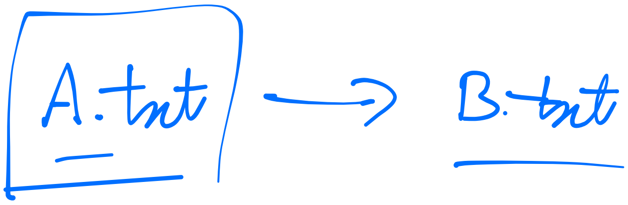
Jordan manages logs for a growing app infrastructure. The logs are getting large, scattered, and unmanageable. Your task is to write a Bash script that automates:

1. Creating the log and archive directory structure
2. Compressing .log files older than 7 days
3. Moving compressed logs to an archive directory
4. Rotating any .log file larger than 100MB
5. Cleaning up archived logs older than 30 days
6. Fixing permissions (644 for files, 755 for dirs)
7. Generating a summary report





1) Modularization

2) Compute Optimization

mv A.txt B.txt

cp A.txt B.txt

1. Create directories

mkdir -p

2. Compress old log files > 7days

find gzip

3. Move compressed files to archive directory

f

1) find .gz files
2) Move to archive dir —exec

4. Rotate file ≥ 100MB

1) wc -c ← 1) Find files with size >100MB
2) stat -c %s 2) Move them

5. Delete archived logs ≥ 30 days

1) find log files > 30days → find
2) delete them → rm ; find -delete

6. Fix perms of files & directories

chmod

7. Generate summary report

date

```
du -sh
```

2. Step 1 : find files older than 7days

Step2 : gzip them

Method 1

```
while IFS= read -r filename do
    gzip $filename
done < find "$logdir" -name "*.log" -mtime +7
```

Method 2

mv {} DIR

```
find "$logdir" -hame "*.log" -mtime +7 -exec gzip {} \
```

Execute

signals command termination

✡ `stat -c %s`

Tells the size of the file

c → custom format

%s → Specify in bytes

SKELETON

```
#!/bin/bash
set -euo pipefail
trap 'echo "Error at line $LINENO";
      exit 1' ERR

LOG_DIR = "/var/app/logs"
ARCHIVE_DIR = "/var/app/archive"
```

```
MAX_SIZE = 100000000
RETENTION_DAYS = 30

create_directories ()
{

  }

compres_logs ()
{

 }

move_to_archive ()
{

}
```

```
}

rotate ()
{

}

delete_old_files ()
{

}

fix_perms ()
{
}

generate_report ()
{

}

main ()
{
    Create_directories
    compress_logs
    move_to_archive
    rotate
    delete_old_files
```

```
fix_perms
generate_report
```

```
main "$@"
```

1

all args passed by user

```
root@ip-172-31-27-45:~# cat script.sh
#!/bin/bash
set -euo pipefail
trap 'echo "Error at line $LINENO"; exit 1' ERR

LOG_DIR="/var/app/logs"
ARCHIVE_DIR="/var/app/archive"
RETENTION_DAYS=30
MAX_SIZE=100000000

create_directories(){
  mkdir -p "$LOG_DIR" "$ARCHIVE_DIR"
  echo "Directories Created"
}

compress_old_logs(){
  find "$LOG_DIR" -type f -name "*.log"  -exec gzip {} \;
  echo "Old Logs compressed"
}

move_to_archive(){
   find "$LOG_DIR" -type f -name "*.gz" -exec mv {} $ARCHIVE_DIR/ \;

}

rotate(){
  find "$LOG_DIR" -type f -name "*.log" | while read -r file ; do
    size=$(stat -c%s "$file")
    echo $size
    if [ "$size" -gt "$MAX_SIZE" ]; then
       format=$(date +%Y%m%d_%H_%M_%S)
       mv "$file" "${file}.${format}"
       touch $file
       echo "Rotated Log File"
    fi
  done
}

delete_old_archives(){
        find "$ARCHIVE_DIR" -type f -name "*.gz" -mtime +$RETENTION_DAYS -delete
        echo "Deleted Old Archives"
}

fix_perms(){
   find "$LOG_DIR" -type f -exec chmod 644 {} \;
   chmod 755 "$LOG_DIR"
   #find "$LOG_DIR" -type d -exec chmod 755 {} \ ;
   find "$ARCHIVE_DIR" -type f -exec chmod 644 {} \;
   #find "$ARCHIVE_DIR" -type d -exec chmod 755 {} \ ;
   chmod 755 "$ARCHIVE_DIR"
   echo "File Perms Fixed"
}

generate_report(){
  echo "New Report $(date)"
  echo "Log Size "
  du -sh "$LOG_DIR"
  echo "Archive Dir Size"
  du -sh "$ARCHIVE_DIR"
}

main(){
  create_directories
  #compress_old_logs
  #move_to_archive
  rotate
  delete_old_archives
  fix_perms
  generate_report
}

main "$@"
```