Introduction_to_linux

Assignment 2

Student Info

- Name: Ntwari Ashimwe Fiacre
- Student ID: 27438
- Assignment: 2

Q1. Which directories might an attacker modify?

From q1_answer.txt

- /etc : system configuration files (passwd, sshd_config, cron jobs). Attackers change configs here.
- /bin : essential binaries (shells, coreutils) which if replaced can hide activity or run malicious code.
- /usr : userland programs; /usr/bin contains many user commands that could be swapped.
- /var : variable data, especially /var/log (intrusion evidence) and /var/spool, /var/tmp (persistence).
- /tmp : world-writable temp; used for uploads or transient scripts (persistence).
- /opt : third-party optional software — often used for vendor apps; attackers may drop tools here.
- /boot : kernel images and bootloader config; tampering here can persist across reboots.
- /home : user files and ssh keys; attackers place backdoors or harvest credentials.

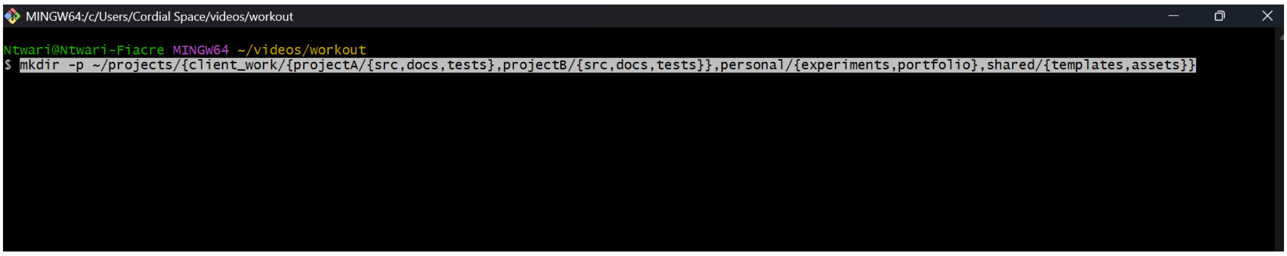
Reasoning: configuration files under /etc control services. Binaries in /bin and /usr/bin are executed by the system; replacing them provides stealth. Logs in /var/log reveal intrusion evidence; /tmp is writable and convenient for attackers; /boot alters system startup; user ssh keys and scripts live in /home.

Q2. Create nested directories structure efficiently

From q2_answer.txt

```
mkdir -p
~/projects/{client_work/{projectA/{src,docs,tests},projectB/{src,docs,tests}},personal/{experiments,portfolio},shared/{templates,assets}}

# This creates:
# - ~/projects/client_work/projectA/{src,docs,tests}
# - ~/projects/client_work/projectB/{src,docs,tests}
# - ~/projects/personal/experiments
# - ~/projects/personal/portfolio
# - ~/projects/shared/templates
# - ~/projects/shared/assets
```

Screenshots:

- 

MINGW64:/c/Users/Cordial Space/videos/workout

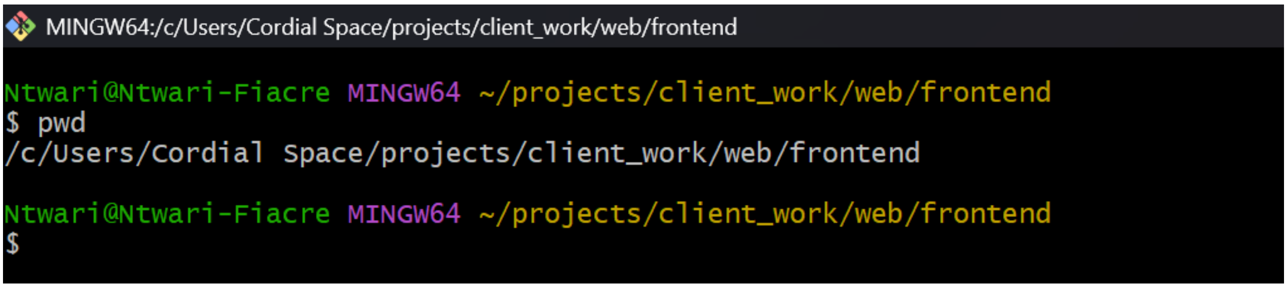
```
Ntwari@Ntwari-Fiacre MINGW64 ~/videos/workout
$ mkdir -p ~/projects/{client_work/{projectA/{src,docs,tests},projectB/{src,docs,tests}},personal/{experiments,portfolio},shared/{templates,assets}}
```

Q3. Navigate without absolute paths (limit cd usage)

From q3.sh

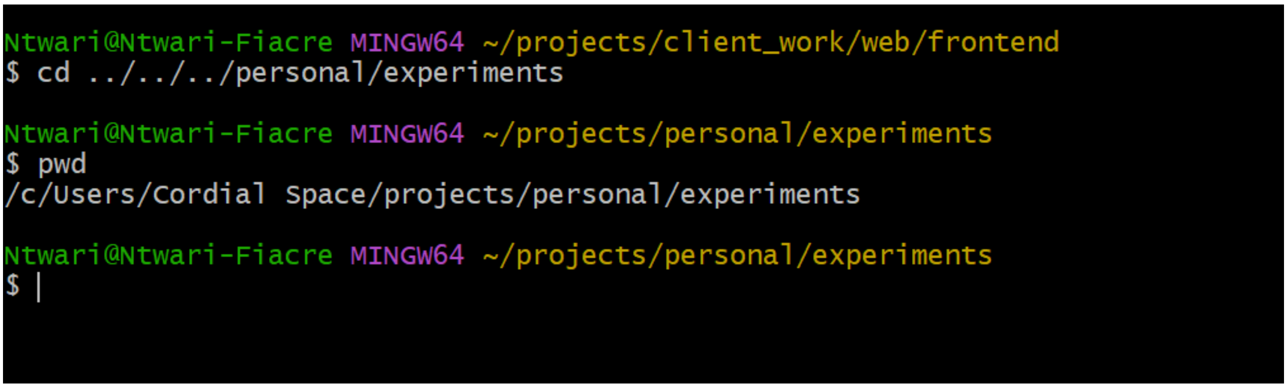
```
# Starting dir was ~/projects/client_work/web/frontend
pwd       # prove starting location
cd ../../../../personal/experiments
pwd       # prove we are in ~/projects/personal/experiments
cd ../../shared/templates
pwd       # prove we are in ~/projects/shared/templates
cd ../../../client_work/web/frontend
pwd       # prove we are back to start

# Alternative using pushd/popd is also valid.
```

Screenshots:

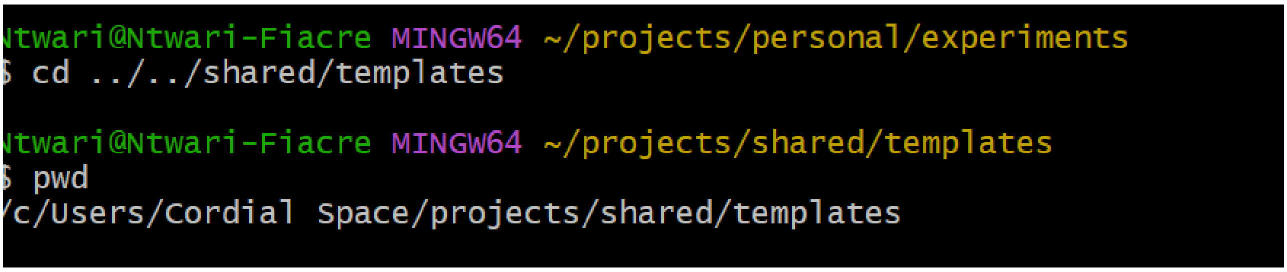
- 

MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend

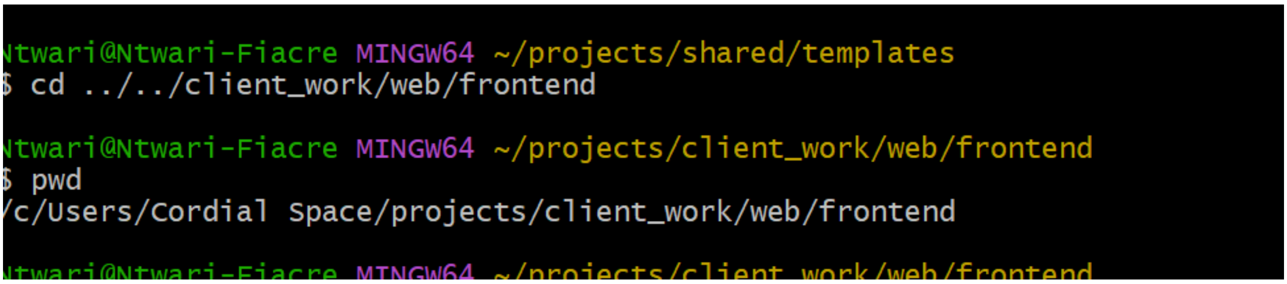
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend
$ pwd
/c/Users/Cordial Space/projects/client_work/web/frontend

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend
$
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend
$ cd ../../../personal/experiments

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/personal/experiments
$ pwd
/c/Users/Cordial Space/projects/personal/experiments

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/personal/experiments
$ |
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/personal/experiments
$ cd ../../shared/templates

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/shared/templates
$ pwd
/c/Users/Cordial Space/projects/shared/templates
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/shared/templates
$ cd ../../client_work/web/frontend

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend
$ pwd
/c/Users/Cordial Space/projects/client_work/web/frontend

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend
```

Q4. Generate a realistic web project structure with minimal commands

From q4.txt

The trick here is to generate the whole realistic web project structure with the fewest commands possible using brace expansion and loops.

Here’s the clean and efficient Git Bash solution:

Step 1: Create project directory

```
mkdir -p web_project && cd web_project
```

Step 2: Create 15 HTML files

Pattern: index.html, about.html, contact.html, page_001.html … page_012.html

```
touch index.html about.html contact.html page_{001..012}.html
```

Step 3: Create 8 CSS files

Names are fixed list:

```
touch {main,reset,theme_light,theme_dark,mobile,tablet,desktop,print}.css
```

Step 4: Create 6 JavaScript files

Requirement: names must include script, util, and config variations. Example:

```
touch {app_script,loader_script,ui_script,util_dom,util_helpers,config_app}.js
```

Step 5: Create 20 backup files (5 each starting with a, b, c, d with mixed extensions)

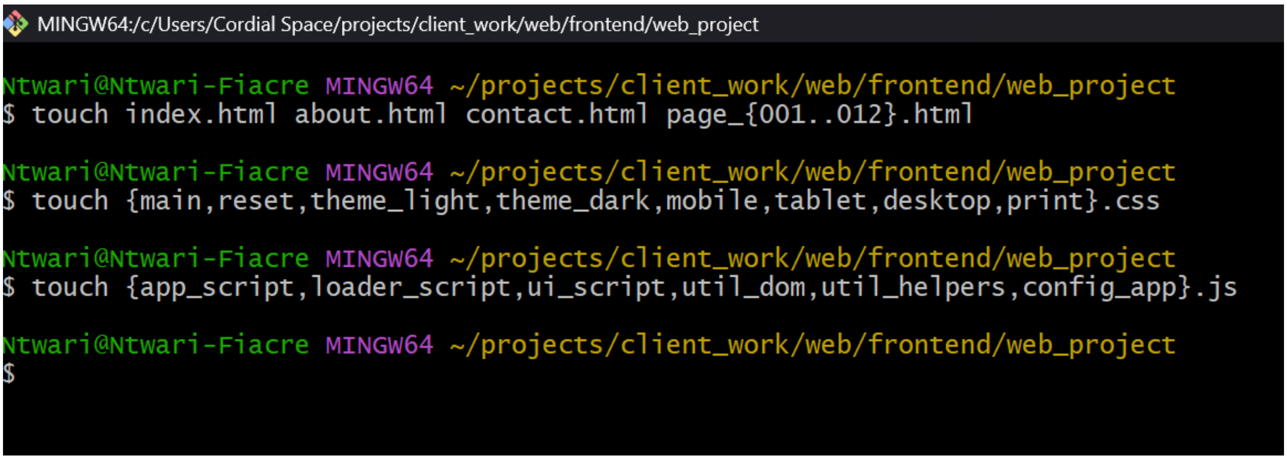
```
for L in a b c d; do
  touch ${L}1.bak ${L}2.old ${L}3.zip ${L}4.tar ${L}5.txt
```

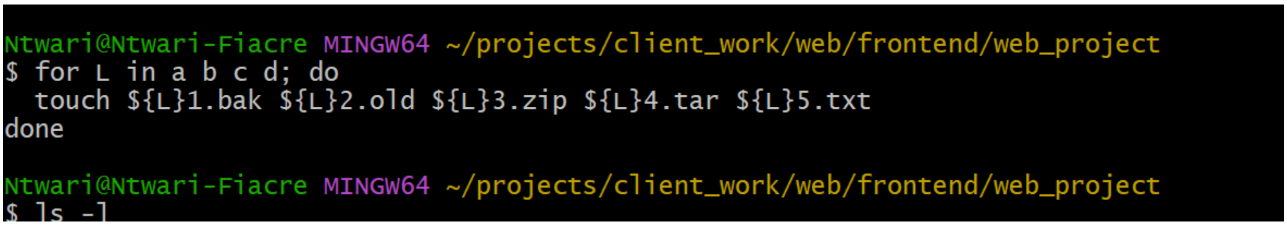
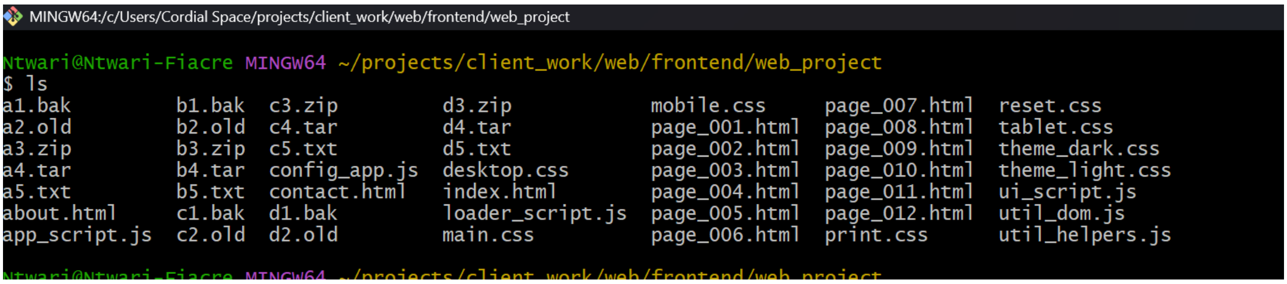
```
done
```

Step 6: Verify structure

```
ls -1
```

Screenshots:

- 
  MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project

  ```
  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ touch index.html about.html contact.html page_{001..012}.html

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ touch {main,reset,theme_light,theme_dark,mobile,tablet,desktop,print}.css

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ touch {app_script,loader_script,ui_script,util_dom,util_helpers,config_app}.js

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $
  ```
- 
  ```
  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ for L in a b c d; do
    touch ${L}1.bak ${L}2.old ${L}3.zip ${L}4.tar ${L}5.txt
  done

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ ls -l
  ```
- 
  MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project

  ```
  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  $ ls
  a1.bak        b1.bak  c3.zip         d3.zip            mobile.css     page_007.html  reset.css
  a2.old        b2.old  c4.tar         d4.tar            page_001.html  page_008.html  tablet.css
  a3.zip        b3.zip  c5.txt         d5.txt            page_002.html  page_009.html  theme_dark.css
  a4.tar        b4.tar  config_app.js  desktop.css       page_003.html  page_010.html  theme_light.css
  a5.txt        b5.txt  contact.html   index.html        page_004.html  page_011.html  ui_script.js
  about.html    c1.bak  d1.bak         loader_script.js  page_005.html  page_012.html  util_dom.js
  app_script.js c2.old  d2.old         main.css          page_006.html  print.css      util_helpers.js

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
  ```

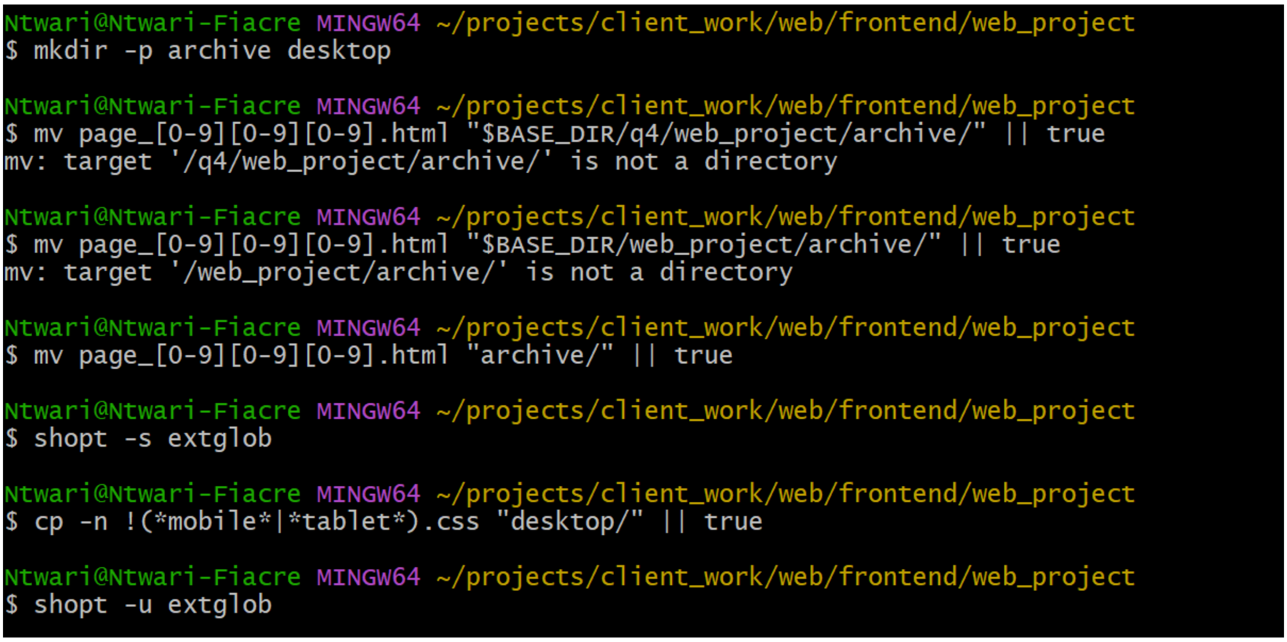
Q5. Pattern matching and selective operations

From q5.txt

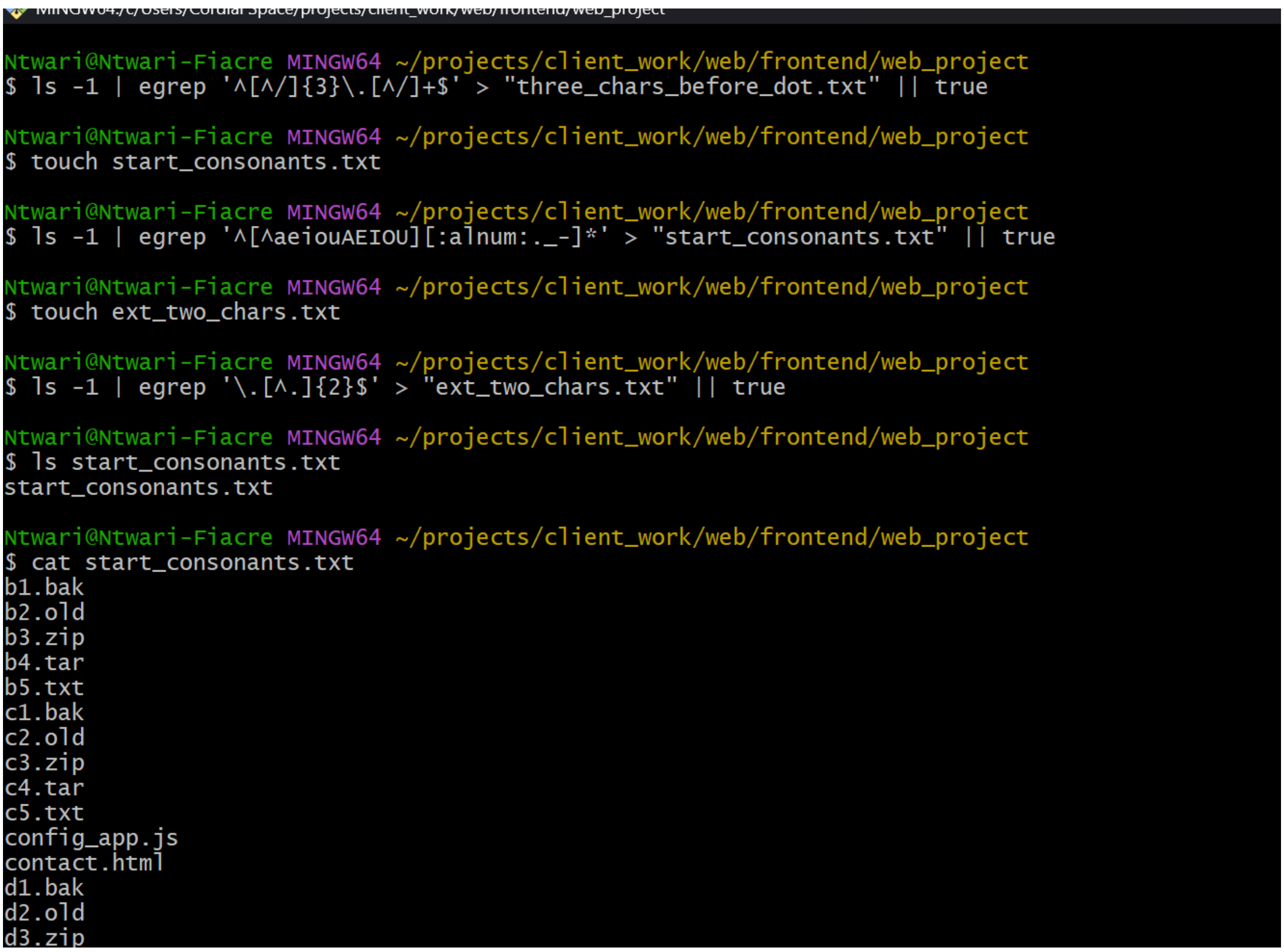
```
# Move files ending in numbers (before extension)
mv *[0-9].* archive/

# Copy CSS except mobile/tablet (requires extglob)
shopt -s extglob
cp !(*mobile*|*tablet*).css desktop/

# List files with 3 characters before dot
ls ???.*
```

```
# Find files starting with a consonant
ls [!aeiouAEIOU]*

# Find files with 2-char extension
ls *.[[:alpha:]][[:alpha:]]
```

Screenshots:

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ mkdir -p archive desktop

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ mv page_[0-9][0-9][0-9].html "$BASE_DIR/q4/web_project/archive/" || true
mv: target '/q4/web_project/archive/' is not a directory

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ mv page_[0-9][0-9][0-9].html "$BASE_DIR/web_project/archive/" || true
mv: target '/web_project/archive/' is not a directory

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ mv page_[0-9][0-9][0-9].html "archive/" || true

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ shopt -s extglob

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ cp -n !(*mobile*|*tablet*).css "desktop/" || true

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ shopt -u extglob
```

- 

MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project

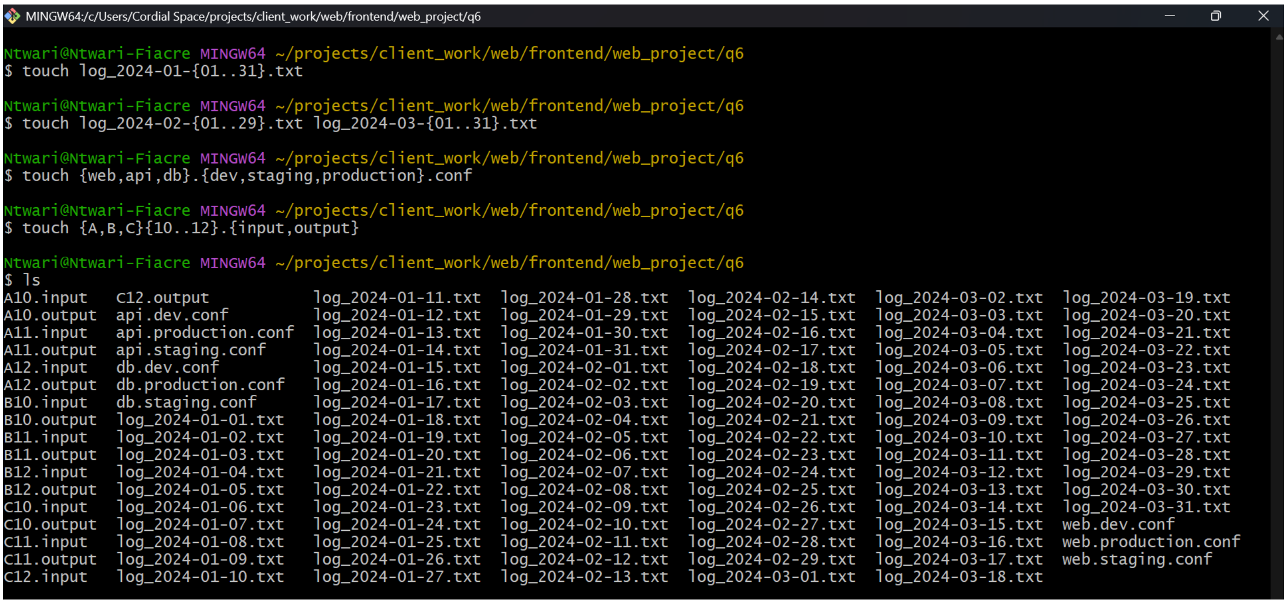
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls -1 | egrep '^[^/]{3}\.[^/]+$' > "three_chars_before_dot.txt" || true

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ touch start_consonants.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls -1 | egrep '^[^aeiouAEIOU][:alnum:._-]*' > "start_consonants.txt" || true

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ touch ext_two_chars.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls -1 | egrep '\.[^.]{2}$' > "ext_two_chars.txt" || true

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls start_consonants.txt
start_consonants.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ cat start_consonants.txt
b1.bak
b2.old
b3.zip
b4.tar
b5.txt
c1.bak
c2.old
c3.zip
c4.tar
c5.txt
config_app.js
contact.html
d1.bak
d2.old
d3.zip
```

Q6. Create logs, configs, and test files efficiently

From q6.txt

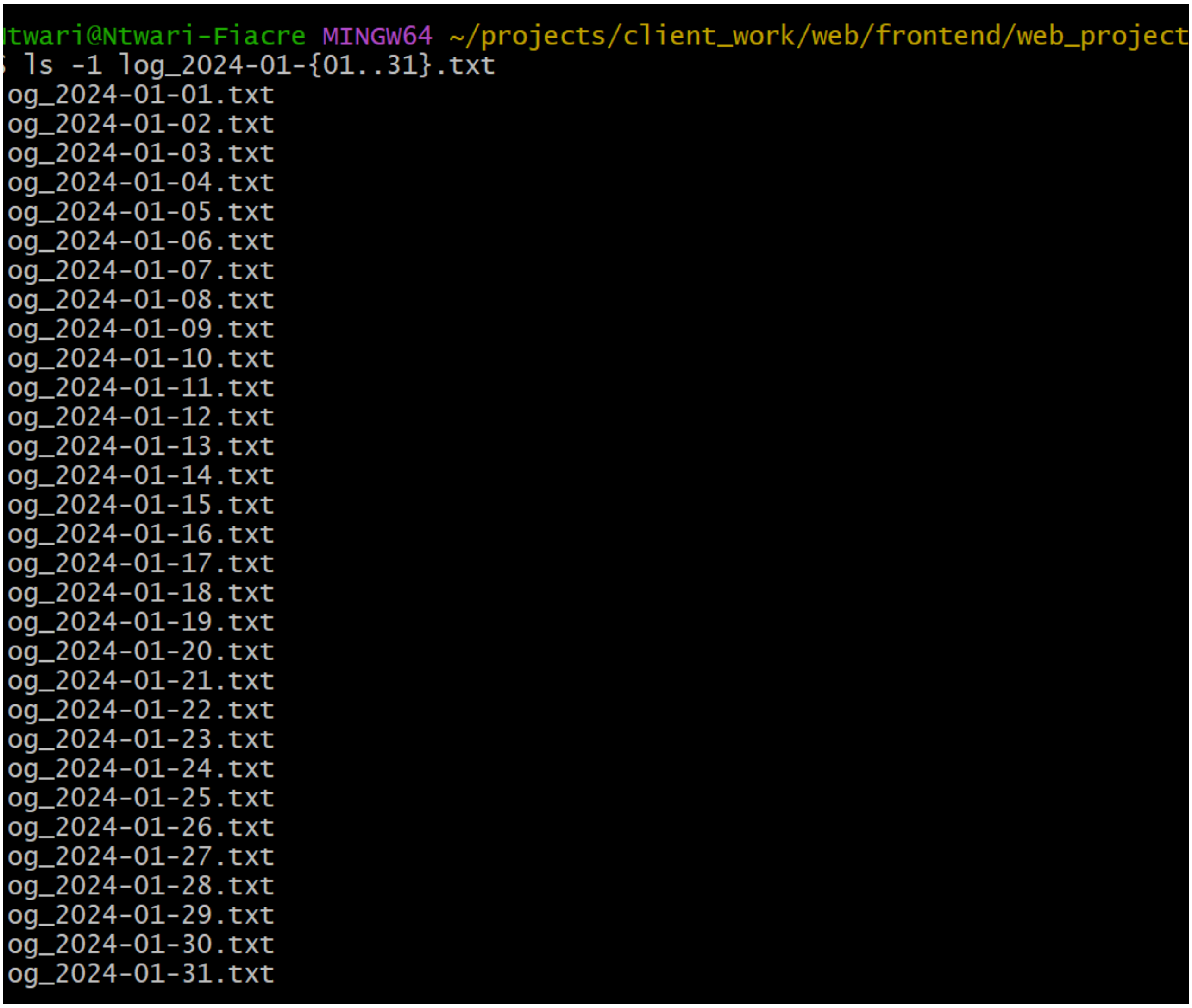
```
# Logs (Jan, Feb, Mar 2024)
touch log_2024-01-{01..31}.txt
touch log_2024-02-{01..29}.txt
touch log_2024-03-{01..31}.txt

# Configs (3 services × 3 envs)
touch {web,api,db}.{dev,staging,production}.conf

# Test files (A–C × 10–12 × input/output)
touch {A,B,C}{10..12}.{input,output}
```

Screenshots:

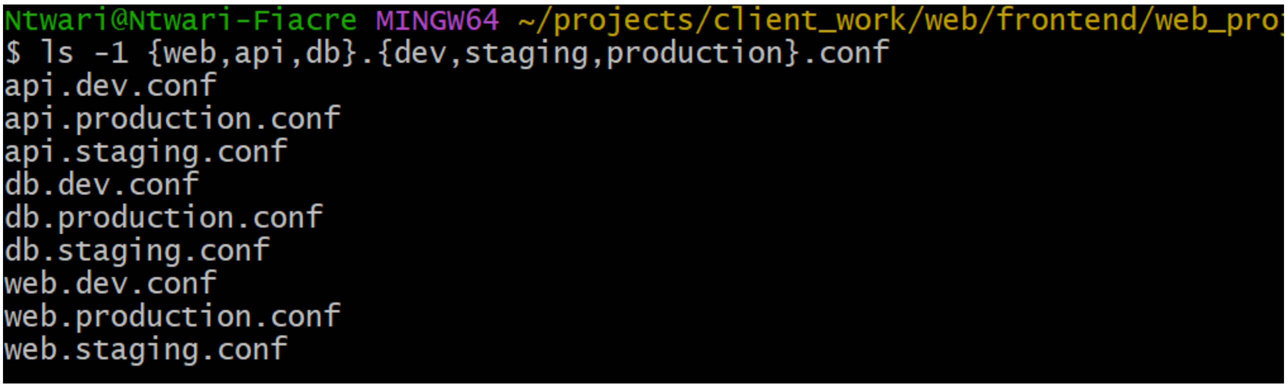
- 

MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project/q6

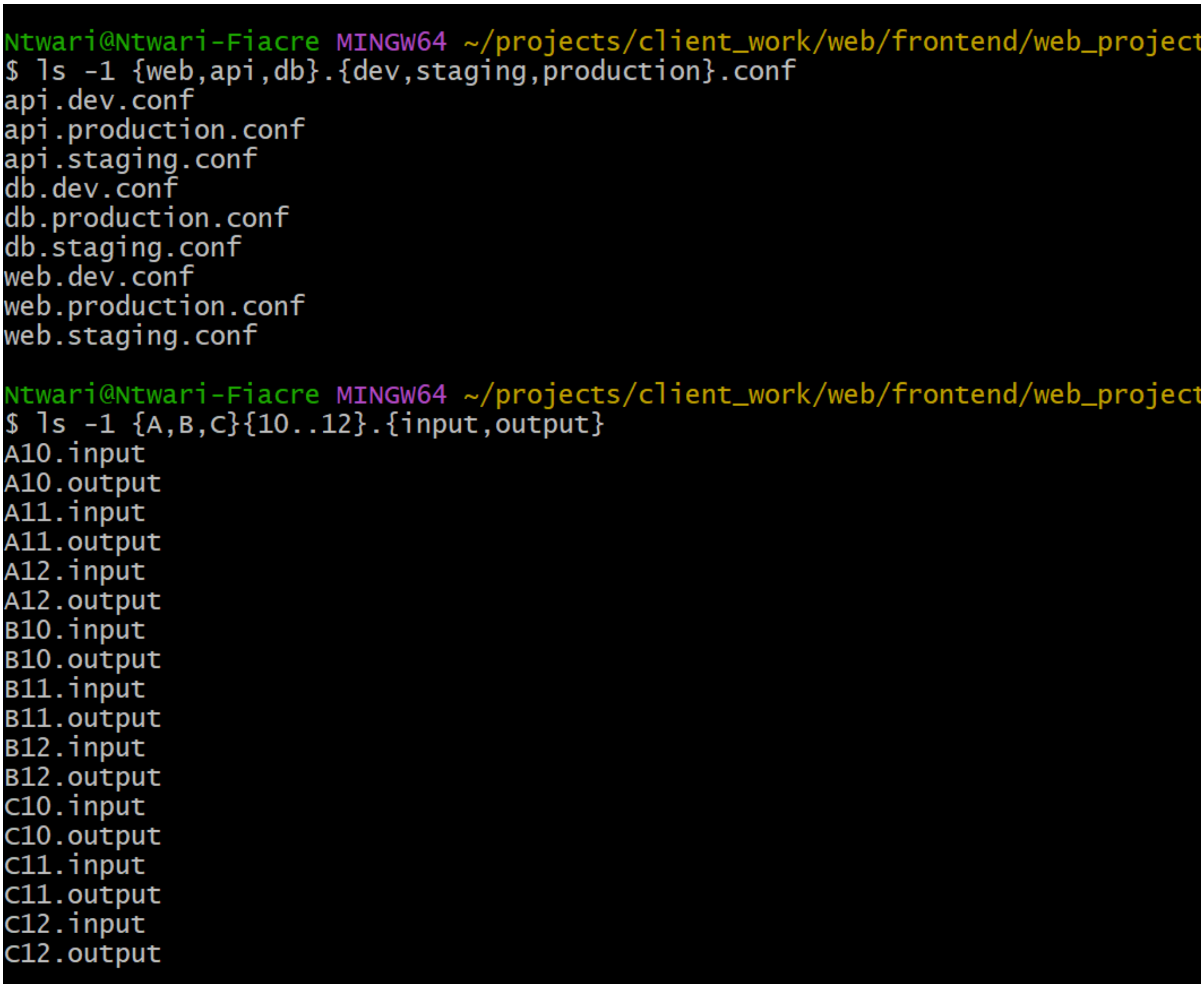
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q6
$ touch log_2024-01-{01..31}.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q6
$ touch log_2024-02-{01..29}.txt log_2024-03-{01..31}.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q6
$ touch {web,api,db}.{dev,staging,production}.conf

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q6
$ touch {A,B,C}{10..12}.{input,output}

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q6
$ ls
A10.input   C12.output           log_2024-01-11.txt  log_2024-01-28.txt  log_2024-02-14.txt  log_2024-03-02.txt  log_2024-03-19.txt
A10.output  api.dev.conf         log_2024-01-12.txt  log_2024-01-29.txt  log_2024-02-15.txt  log_2024-03-03.txt  log_2024-03-20.txt
A11.input   api.production.conf  log_2024-01-13.txt  log_2024-01-30.txt  log_2024-02-16.txt  log_2024-03-04.txt  log_2024-03-21.txt
A11.output  api.staging.conf     log_2024-01-14.txt  log_2024-01-31.txt  log_2024-02-17.txt  log_2024-03-05.txt  log_2024-03-22.txt
A12.input   db.dev.conf          log_2024-01-15.txt  log_2024-02-01.txt  log_2024-02-18.txt  log_2024-03-06.txt  log_2024-03-23.txt
A12.output  db.production.conf   log_2024-01-16.txt  log_2024-02-02.txt  log_2024-02-19.txt  log_2024-03-07.txt  log_2024-03-24.txt
B10.input   db.staging.conf      log_2024-01-17.txt  log_2024-02-03.txt  log_2024-02-20.txt  log_2024-03-08.txt  log_2024-03-25.txt
B10.output  log_2024-01-01.txt   log_2024-01-18.txt  log_2024-02-04.txt  log_2024-02-21.txt  log_2024-03-09.txt  log_2024-03-26.txt
B11.input   log_2024-01-02.txt   log_2024-01-19.txt  log_2024-02-05.txt  log_2024-02-22.txt  log_2024-03-10.txt  log_2024-03-27.txt
B11.output  log_2024-01-03.txt   log_2024-01-20.txt  log_2024-02-06.txt  log_2024-02-23.txt  log_2024-03-11.txt  log_2024-03-28.txt
B12.input   log_2024-01-04.txt   log_2024-01-21.txt  log_2024-02-07.txt  log_2024-02-24.txt  log_2024-03-12.txt  log_2024-03-29.txt
B12.output  log_2024-01-05.txt   log_2024-01-22.txt  log_2024-02-08.txt  log_2024-02-25.txt  log_2024-03-13.txt  log_2024-03-30.txt
C10.input   log_2024-01-06.txt   log_2024-01-23.txt  log_2024-02-09.txt  log_2024-02-26.txt  log_2024-03-14.txt  log_2024-03-31.txt
C10.output  log_2024-01-07.txt   log_2024-01-24.txt  log_2024-02-10.txt  log_2024-02-27.txt  log_2024-03-15.txt  web.dev.conf
C11.input   log_2024-01-08.txt   log_2024-01-25.txt  log_2024-02-11.txt  log_2024-02-28.txt  log_2024-03-16.txt  web.production.conf
C11.output  log_2024-01-09.txt   log_2024-01-26.txt  log_2024-02-12.txt  log_2024-02-29.txt  log_2024-03-17.txt  web.staging.conf
C12.input   log_2024-01-10.txt   log_2024-01-27.txt  log_2024-02-13.txt  log_2024-03-01.txt  log_2024-03-18.txt
```

- 

```
twari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
 ls -1 log_2024-01-{01..31}.txt
og_2024-01-01.txt
og_2024-01-02.txt
og_2024-01-03.txt
og_2024-01-04.txt
og_2024-01-05.txt
og_2024-01-06.txt
og_2024-01-07.txt
og_2024-01-08.txt
og_2024-01-09.txt
og_2024-01-10.txt
og_2024-01-11.txt
og_2024-01-12.txt
og_2024-01-13.txt
og_2024-01-14.txt
og_2024-01-15.txt
og_2024-01-16.txt
og_2024-01-17.txt
og_2024-01-18.txt
og_2024-01-19.txt
og_2024-01-20.txt
og_2024-01-21.txt
og_2024-01-22.txt
og_2024-01-23.txt
og_2024-01-24.txt
og_2024-01-25.txt
og_2024-01-26.txt
og_2024-01-27.txt
og_2024-01-28.txt
og_2024-01-29.txt
og_2024-01-30.txt
og_2024-01-31.txt
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_proj
$ ls -1 {web,api,db}.{dev,staging,production}.conf
api.dev.conf
api.production.conf
api.staging.conf
db.dev.conf
db.production.conf
db.staging.conf
web.dev.conf
web.production.conf
web.staging.conf
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls -1 {web,api,db}.{dev,staging,production}.conf
api.dev.conf
api.production.conf
api.staging.conf
db.dev.conf
db.production.conf
db.staging.conf
web.dev.conf
web.production.conf
web.staging.conf

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project
$ ls -1 {A,B,C}{10..12}.{input,output}
A10.input
A10.output
A11.input
A11.output
A12.input
A12.output
B10.input
B10.output
B11.input
B11.output
B12.input
B12.output
C10.input
C10.output
C11.input
C11.output
C12.input
C12.output
```

Q7. Why do diff/cmp/comm treat similar-looking files as different?

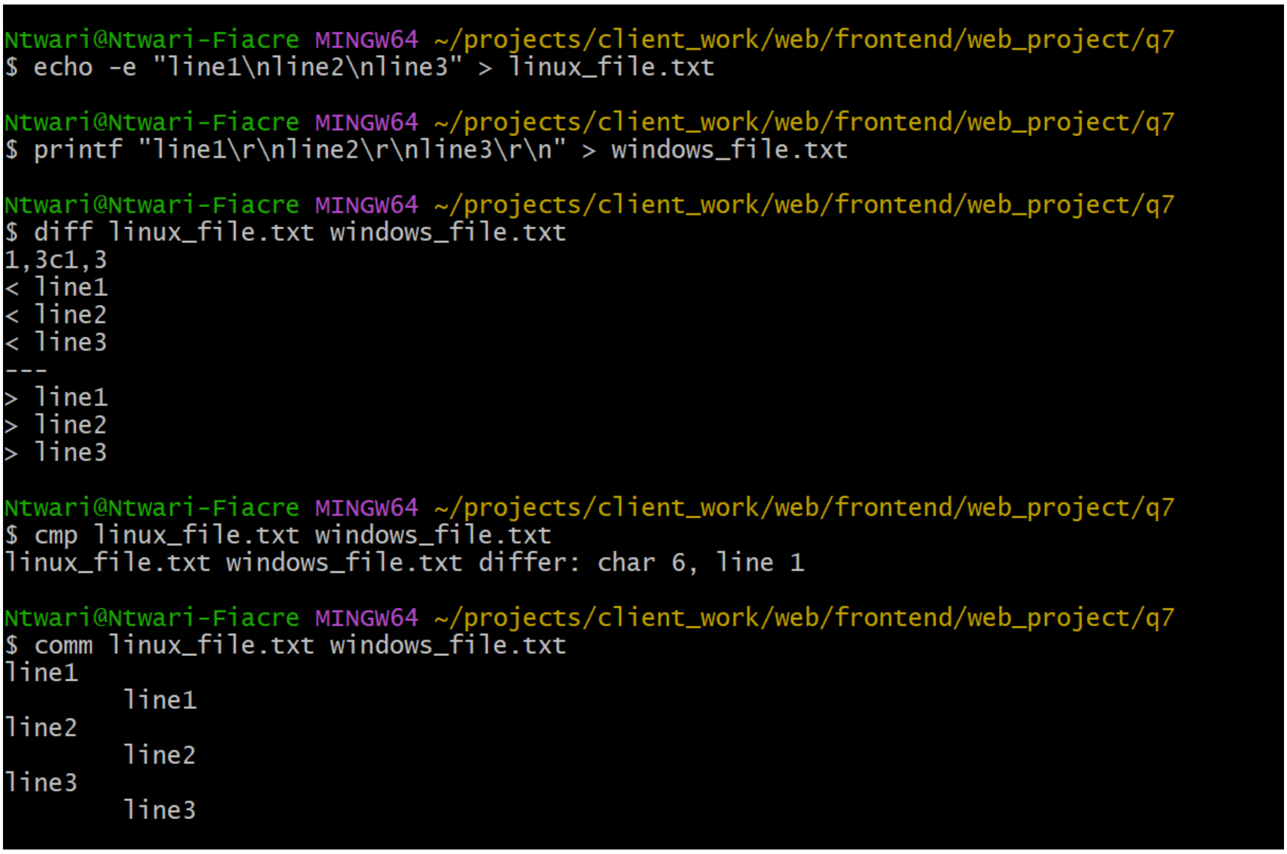
From q7.txt

Although both files had the same visible text, they were treated differently because of line endings.

- diff reported every line as different, since it saw `\n` vs `\r\n`.
- cmp found the first differing byte position (extra `\r` in CRLF).
- comm considered the lines different entirely, since the hidden carriage return made the strings mismatch.

Lesson: Cross-platform compatibility issues arise when files are edited across Linux and Windows. Config files or scripts may break because of incorrect line endings. The safe practice is to normalize files using tools like dos2unix, unix2dos, or Git’s core.autocrlf setting.

Screenshots:

- 

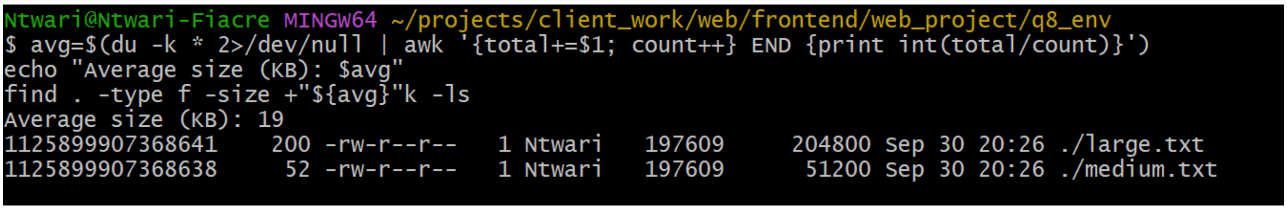
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q7
$ echo -e "line1\nline2\nline3" > linux_file.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q7
$ printf "line1\r\nline2\r\nline3\r\n" > windows_file.txt

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q7
$ diff linux_file.txt windows_file.txt
1,3c1,3
< line1
< line2
< line3
---
> line1
> line2
> line3

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q7
$ cmp linux_file.txt windows_file.txt
linux_file.txt windows_file.txt differ: char 6, line 1

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q7
$ comm linux_file.txt windows_file.txt
line1
        line1
line2
        line2
line3
        line3
```

Q8. File hunting rules for incident response

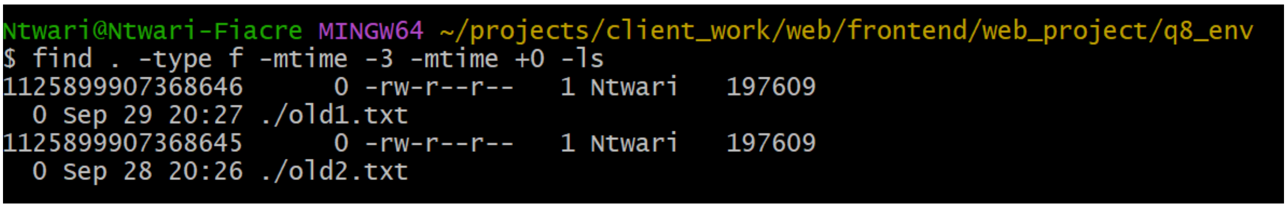
From q8.txt

- Larger than average size: useful to flag unusually large logs or dumps.
- Modified within last 72h but not 24h: captures files that may have been tampered with in a recent window.
- Empty/hidden-only directories: suspicious if attackers hide tools in “hidden-only” dirs.
- World-writable files: a major risk; any user can overwrite them.
- Owned by other users: can reveal files that shouldn’t belong to another account.
- Temporary/backup files: may contain sensitive data left unsecured.

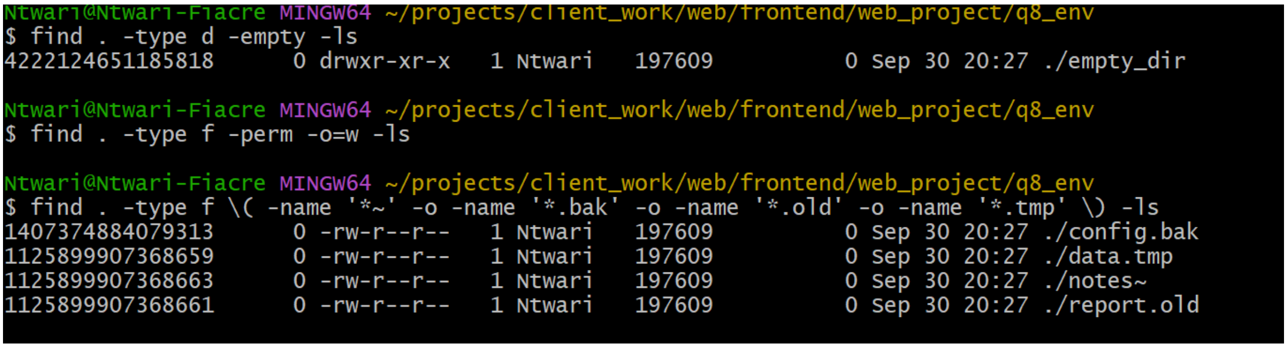
Screenshots:

- 

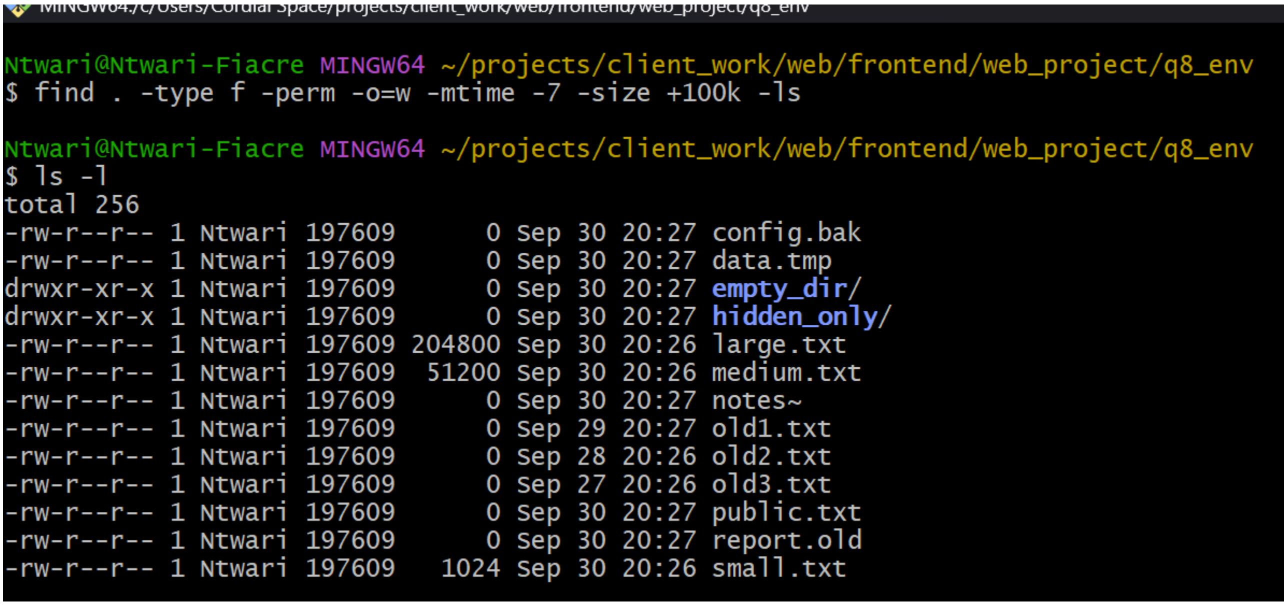
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ avg=$(du -k * 2>/dev/null | awk '{total+=$1; count++} END {print int(total/count)}')
echo "Average size (KB): $avg"
find . -type f -size +"${avg}"k -ls
Average size (KB): 19
1125899907368641    200 -rw-r--r--   1 Ntwari   197609     204800 Sep 30 20:26 ./large.txt
1125899907368638     52 -rw-r--r--   1 Ntwari   197609      51200 Sep 30 20:26 ./medium.txt
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ find . -type f -mtime -3 -mtime +0 -ls
1125899907368646      0 -rw-r--r--   1 Ntwari   197609
  0 Sep 29 20:27 ./old1.txt
1125899907368645      0 -rw-r--r--   1 Ntwari   197609
  0 Sep 28 20:26 ./old2.txt
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ find . -type d -empty -ls
4222124651185818      0 drwxr-xr-x   1 Ntwari   197609          0 Sep 30 20:27 ./empty_dir

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ find . -type f -perm -o=w -ls

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ find . -type f \( -name '*~' -o -name '*.bak' -o -name '*.old' -o -name '*.tmp' \) -ls
1407374884079313      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 20:27 ./config.bak
1125899907368659      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 20:27 ./data.tmp
1125899907368663      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 20:27 ./notes~
1125899907368661      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 20:27 ./report.old
```

- 
MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project/q8_env

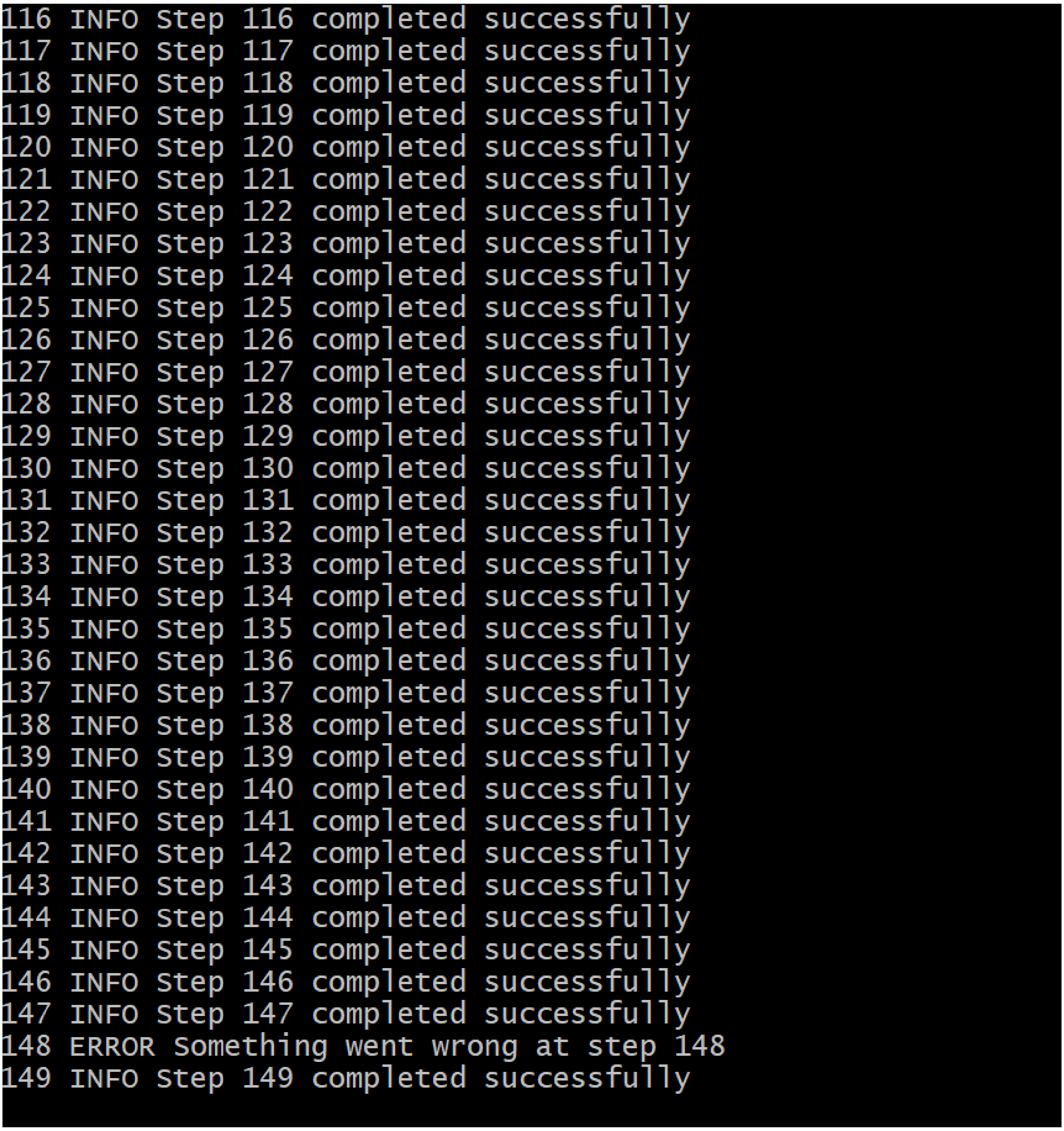
```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ find . -type f -perm -o=w -mtime -7 -size +100k -ls

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q8_env
$ ls -l
total 256
-rw-r--r-- 1 Ntwari 197609      0 Sep 30 20:27 config.bak
-rw-r--r-- 1 Ntwari 197609      0 Sep 30 20:27 data.tmp
drwxr-xr-x 1 Ntwari 197609      0 Sep 30 20:27 empty_dir/
drwxr-xr-x 1 Ntwari 197609      0 Sep 30 20:27 hidden_only/
-rw-r--r-- 1 Ntwari 197609 204800 Sep 30 20:26 large.txt
-rw-r--r-- 1 Ntwari 197609  51200 Sep 30 20:26 medium.txt
-rw-r--r-- 1 Ntwari 197609      0 Sep 30 20:27 notes~
-rw-r--r-- 1 Ntwari 197609      0 Sep 29 20:27 old1.txt
-rw-r--r-- 1 Ntwari 197609      0 Sep 28 20:26 old2.txt
-rw-r--r-- 1 Ntwari 197609      0 Sep 27 20:26 old3.txt
-rw-r--r-- 1 Ntwari 197609      0 Sep 30 20:27 public.txt
-rw-r--r-- 1 Ntwari 197609      0 Sep 30 20:27 report.old
-rw-r--r-- 1 Ntwari 197609   1024 Sep 30 20:26 small.txt
```

Q9. Practical log analysis workflow

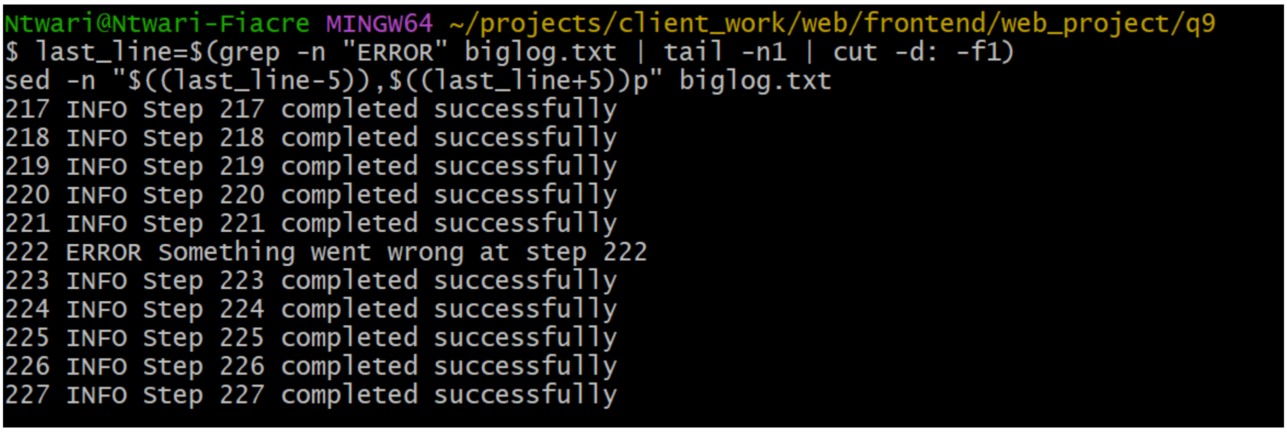
From q9.txt

I generated a 250-line log file (biglog.txt) with periodic ERROR entries. Using sed, I extracted the middle 50 lines for focused troubleshooting. To find the last error, I used `grep -n` with `tail` to get the last match and `sed` to display 5 lines of context. I compared efficiency with `time`: `cat` reads the entire file (slower for huge files), while `less` loads pages on demand (better over SSH with limited bandwidth). For error filtering, `grep -n` allowed me to extract only relevant lines while preserving line numbers for reference. This demonstrates practical log analysis techniques and shows why `less` is superior to `cat` in remote troubleshooting.

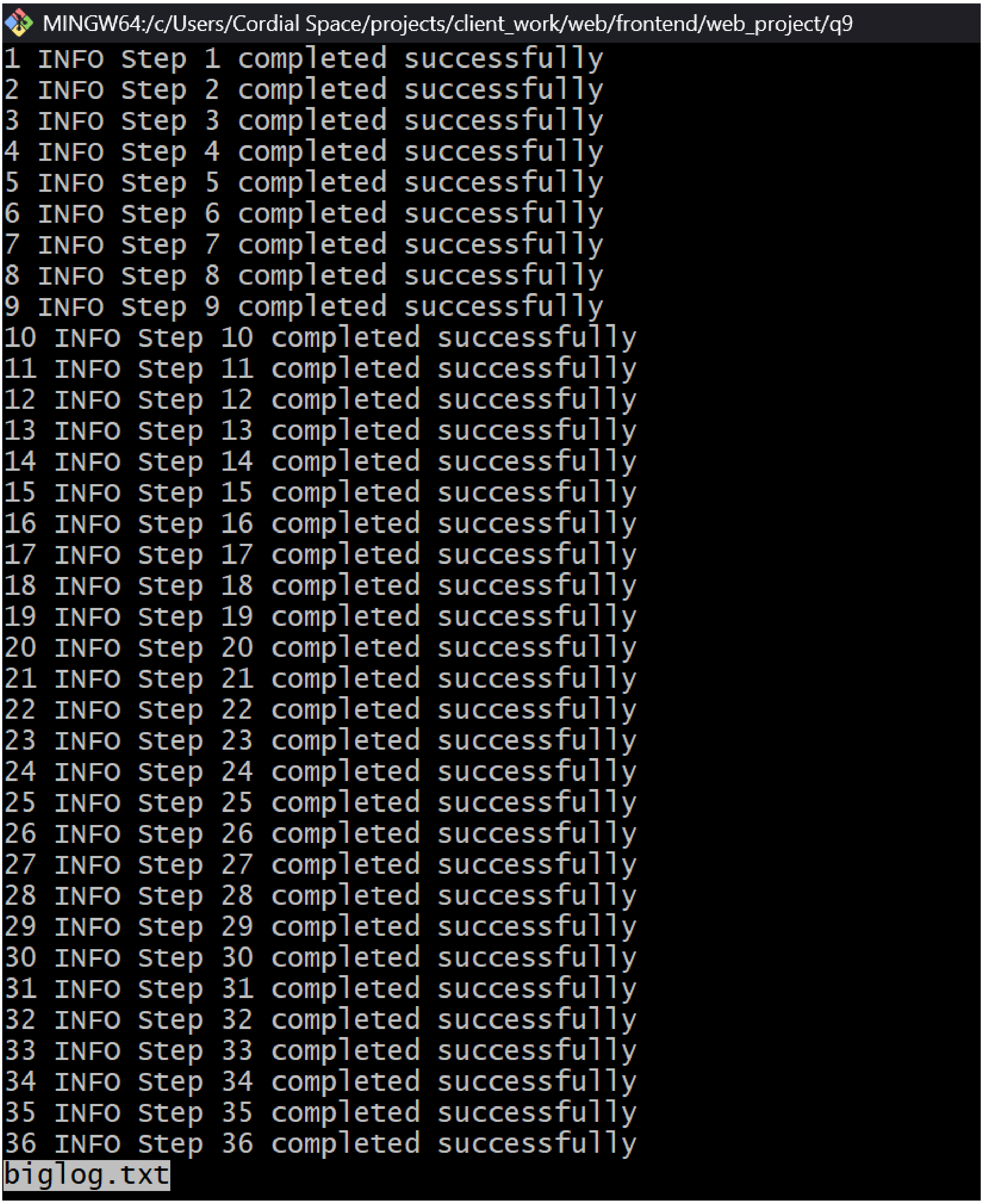
Screenshots:

- 

```
116 INFO Step 116 completed successfully
117 INFO Step 117 completed successfully
118 INFO Step 118 completed successfully
119 INFO Step 119 completed successfully
120 INFO Step 120 completed successfully
121 INFO Step 121 completed successfully
122 INFO Step 122 completed successfully
123 INFO Step 123 completed successfully
124 INFO Step 124 completed successfully
125 INFO Step 125 completed successfully
126 INFO Step 126 completed successfully
127 INFO Step 127 completed successfully
128 INFO Step 128 completed successfully
129 INFO Step 129 completed successfully
130 INFO Step 130 completed successfully
131 INFO Step 131 completed successfully
132 INFO Step 132 completed successfully
133 INFO Step 133 completed successfully
134 INFO Step 134 completed successfully
135 INFO Step 135 completed successfully
136 INFO Step 136 completed successfully
137 INFO Step 137 completed successfully
138 INFO Step 138 completed successfully
139 INFO Step 139 completed successfully
140 INFO Step 140 completed successfully
141 INFO Step 141 completed successfully
142 INFO Step 142 completed successfully
143 INFO Step 143 completed successfully
144 INFO Step 144 completed successfully
145 INFO Step 145 completed successfully
146 INFO Step 146 completed successfully
147 INFO Step 147 completed successfully
148 ERROR Something went wrong at step 148
149 INFO Step 149 completed successfully
```

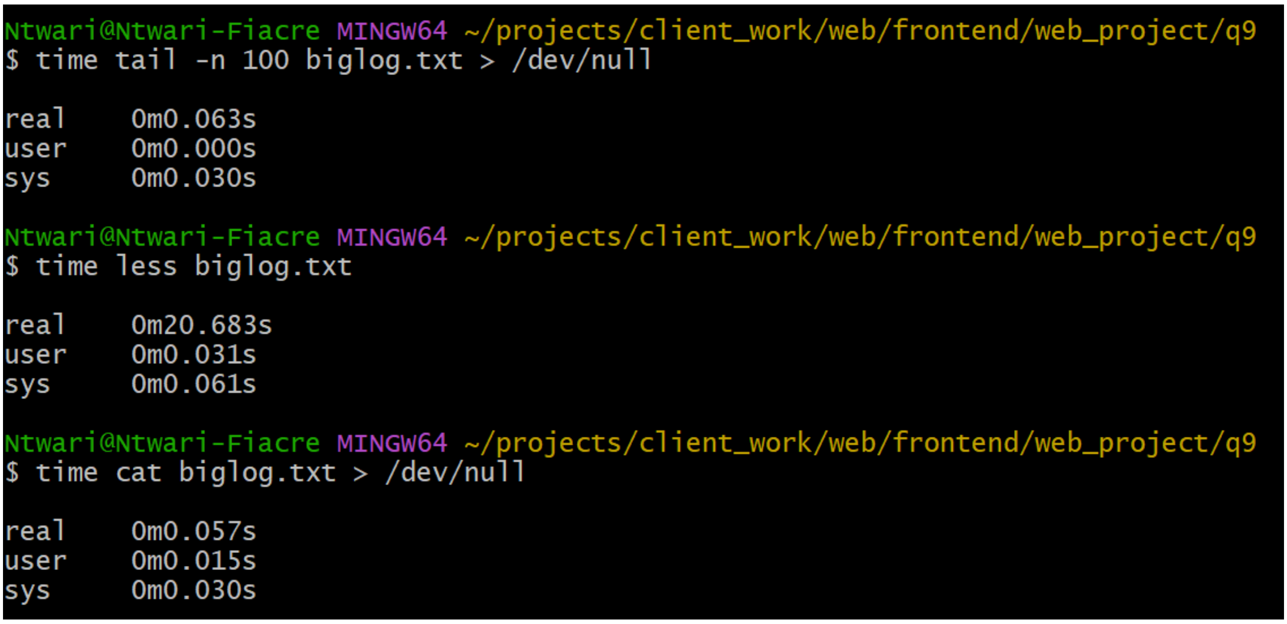
- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q9
$ last_line=$(grep -n "ERROR" biglog.txt | tail -n1 | cut -d: -f1)
sed -n "$((last_line-5)),$((last_line+5))p" biglog.txt
217 INFO Step 217 completed successfully
218 INFO Step 218 completed successfully
219 INFO Step 219 completed successfully
220 INFO Step 220 completed successfully
221 INFO Step 221 completed successfully
222 ERROR Something went wrong at step 222
223 INFO Step 223 completed successfully
224 INFO Step 224 completed successfully
225 INFO Step 225 completed successfully
226 INFO Step 226 completed successfully
227 INFO Step 227 completed successfully
```

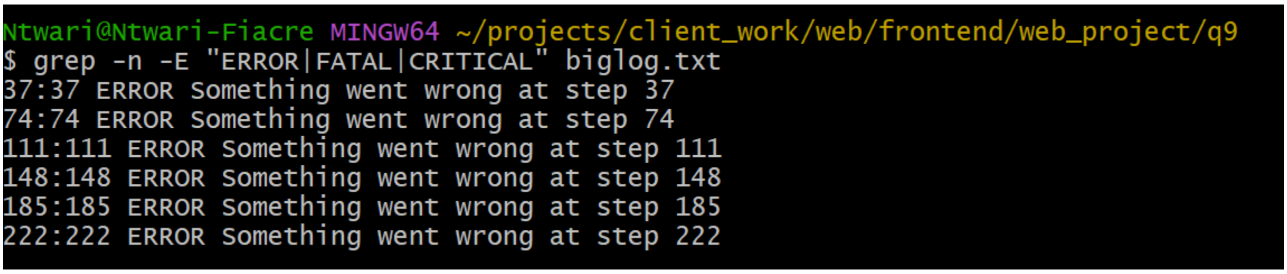
- 

MINGW64:/c/Users/Cordial Space/projects/client_work/web/frontend/web_project/q9

```
1 INFO Step 1 completed successfully
2 INFO Step 2 completed successfully
3 INFO Step 3 completed successfully
4 INFO Step 4 completed successfully
5 INFO Step 5 completed successfully
6 INFO Step 6 completed successfully
7 INFO Step 7 completed successfully
8 INFO Step 8 completed successfully
9 INFO Step 9 completed successfully
10 INFO Step 10 completed successfully
11 INFO Step 11 completed successfully
12 INFO Step 12 completed successfully
13 INFO Step 13 completed successfully
14 INFO Step 14 completed successfully
15 INFO Step 15 completed successfully
16 INFO Step 16 completed successfully
17 INFO Step 17 completed successfully
18 INFO Step 18 completed successfully
19 INFO Step 19 completed successfully
20 INFO Step 20 completed successfully
21 INFO Step 21 completed successfully
22 INFO Step 22 completed successfully
23 INFO Step 23 completed successfully
24 INFO Step 24 completed successfully
25 INFO Step 25 completed successfully
26 INFO Step 26 completed successfully
27 INFO Step 27 completed successfully
28 INFO Step 28 completed successfully
29 INFO Step 29 completed successfully
30 INFO Step 30 completed successfully
31 INFO Step 31 completed successfully
32 INFO Step 32 completed successfully
33 INFO Step 33 completed successfully
34 INFO Step 34 completed successfully
35 INFO Step 35 completed successfully
36 INFO Step 36 completed successfully
biglog.txt
```

- 

  ```
  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q9
  $ time tail -n 100 biglog.txt > /dev/null

  real    0m0.063s
  user    0m0.000s
  sys     0m0.030s

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q9
  $ time less biglog.txt

  real    0m20.683s
  user    0m0.031s
  sys     0m0.061s

  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q9
  $ time cat biglog.txt > /dev/null

  real    0m0.057s
  user    0m0.015s
  sys     0m0.030s
  ```

- 

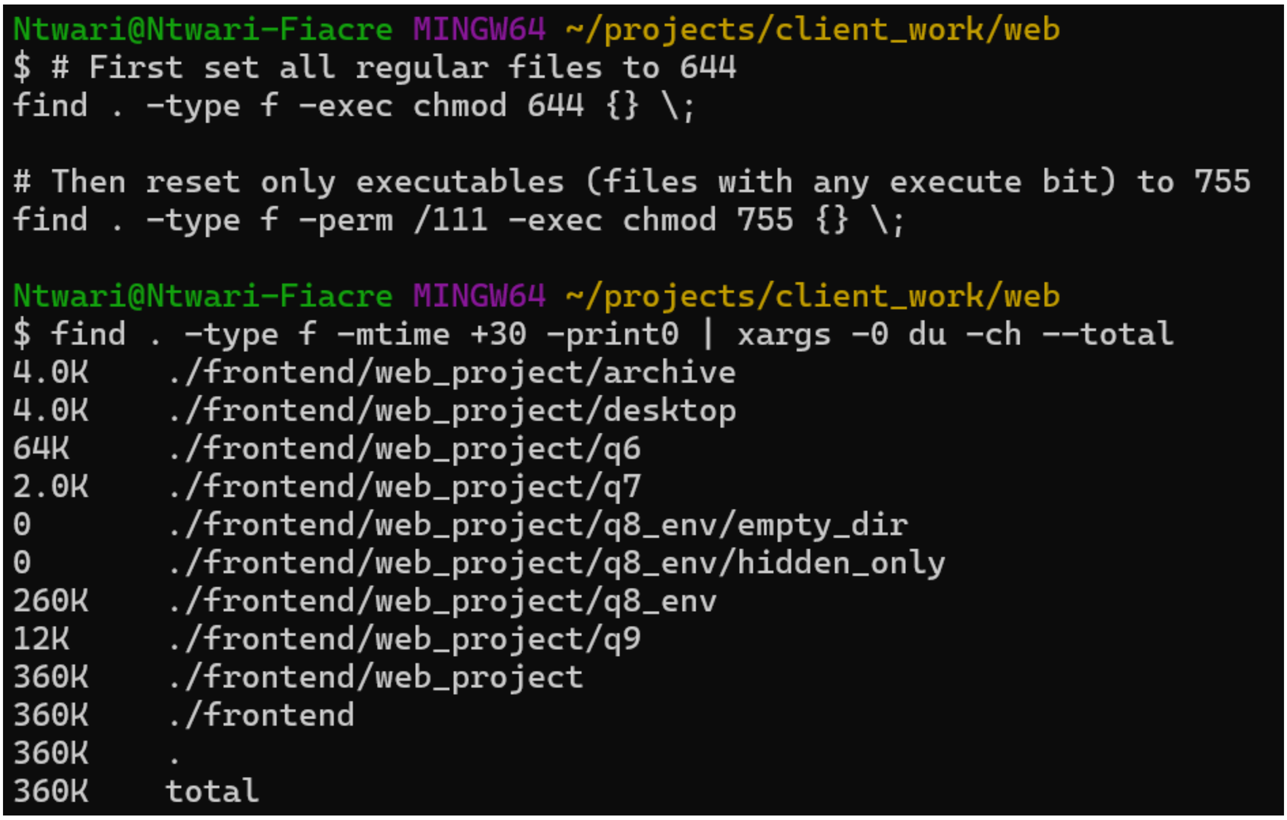
  ```
  Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q9
  $ grep -n -E "ERROR|FATAL|CRITICAL" biglog.txt
  37:37 ERROR Something went wrong at step 37
  74:74 ERROR Something went wrong at step 74
  111:111 ERROR Something went wrong at step 111
  148:148 ERROR Something went wrong at step 148
  185:185 ERROR Something went wrong at step 185
  222:222 ERROR Something went wrong at step 222
  ```

Q10. Automating file maintenance with find -exec

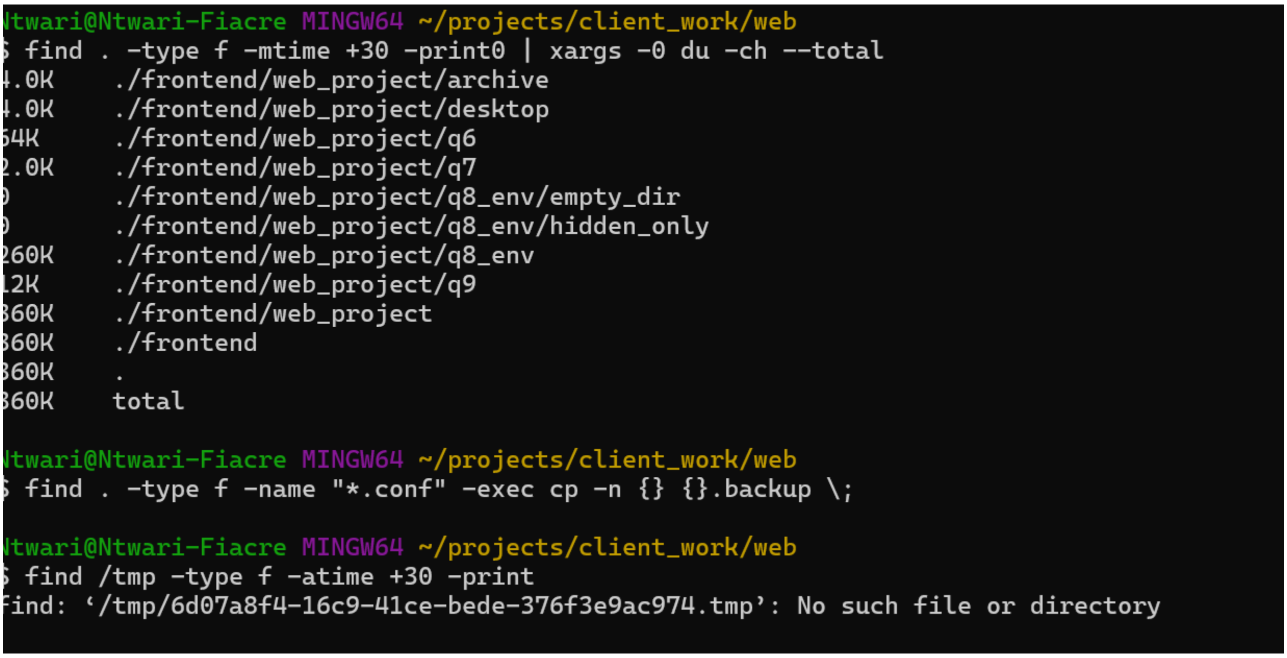
From q10.txt

I automated file maintenance with `find -exec`. I first standardized permissions: all files to 644, but executables restored to 755. To measure storage impact, I computed disk usage of files older than 30 days with `find … | du -ch`. For configuration safety, I created .backup copies of `*.conf` files. To clean temporary files, I targeted files not accessed in 30+ days and used `-ok` for interactive safety before removal. I previewed dangerous operations with `-print` and `-ls` to confirm the target set before execution. This shows how to combine `find` with actions for safe system maintenance.

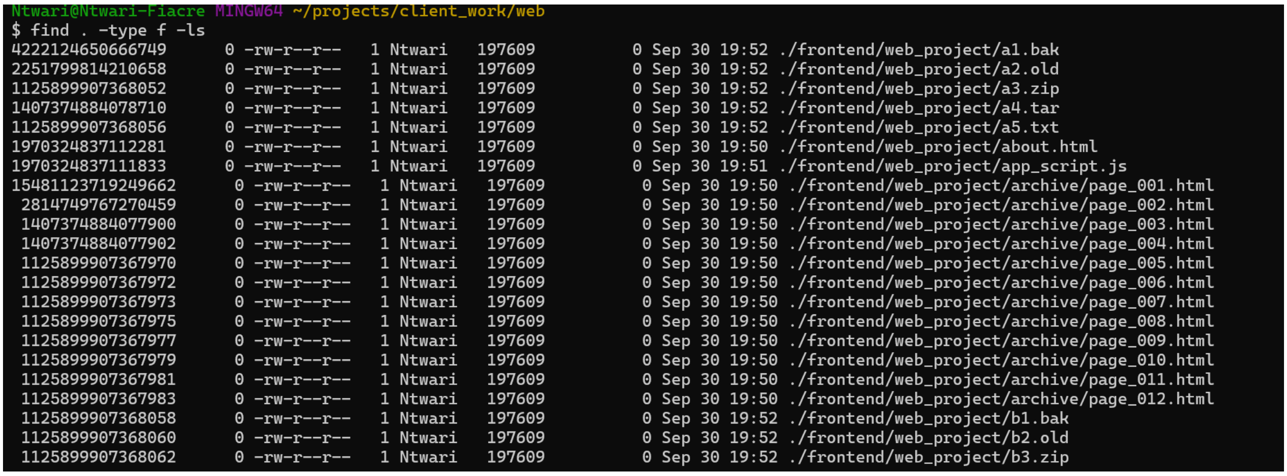
Screenshots:

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
$ # First set all regular files to 644
find . -type f -exec chmod 644 {} \;

# Then reset only executables (files with any execute bit) to 755
find . -type f -perm /111 -exec chmod 755 {} \;

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
$ find . -type f -mtime +30 -print0 | xargs -0 du -ch --total
4.0K    ./frontend/web_project/archive
4.0K    ./frontend/web_project/desktop
64K     ./frontend/web_project/q6
2.0K    ./frontend/web_project/q7
0       ./frontend/web_project/q8_env/empty_dir
0       ./frontend/web_project/q8_env/hidden_only
260K    ./frontend/web_project/q8_env
12K     ./frontend/web_project/q9
360K    ./frontend/web_project
360K    ./frontend
360K    .
360K    total
```

- 

```
twari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
 find . -type f -mtime +30 -print0 | xargs -0 du -ch --total
.0K     ./frontend/web_project/archive
.0K     ./frontend/web_project/desktop
4K      ./frontend/web_project/q6
.0K     ./frontend/web_project/q7
        ./frontend/web_project/q8_env/empty_dir
        ./frontend/web_project/q8_env/hidden_only
260K    ./frontend/web_project/q8_env
12K     ./frontend/web_project/q9
60K     ./frontend/web_project
60K     ./frontend
60K     .
60K     total

twari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
 find . -type f -name "*.conf" -exec cp -n {} {}.backup \;

twari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
 find /tmp -type f -atime +30 -print
ind: '/tmp/6d07a8f4-16c9-41ce-bede-376f3e9ac974.tmp': No such file or directory
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web
$ find . -type f -ls
4222124650666749      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/a1.bak
2251799814210658      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/a2.old
1125899907368052      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/a3.zip
1407374884078710      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/a4.tar
1125899907368056      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/a5.txt
1970324837112281      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/about.html
1970324837111833      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:51 ./frontend/web_project/app_script.js
15481123719249662      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_001.html
 281474976727045 9      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_002.html
 1407374884077900      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_003.html
 1407374884077902      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_004.html
 1125899907367970      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_005.html
 1125899907367972      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_006.html
 1125899907367973      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_007.html
 1125899907367975      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_008.html
 1125899907367977      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_009.html
 1125899907367979      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_010.html
 1125899907367981      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_011.html
 1125899907367983      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:50 ./frontend/web_project/archive/page_012.html
 1125899907368058      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/b1.bak
 1125899907368060      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/b2.old
 1125899907368062      0 -rw-r--r--   1 Ntwari   197609          0 Sep 30 19:52 ./frontend/web_project/b3.zip
```

Q11. Storage space compression analysis

From q11_and_q12.txt (Question 11 section)

What I Did: I created two directories, one containing already compressed files (jpg, mp4, zip) and another containing uncompressed text files. I then created archives of each directory using four compression methods: tar+gzip, tar+bzip2, tar+xz, and zip. I measured the compression ratios and speeds for each method and analyzed which compressed better depending on the file types.

Why I Did It: This exercise helps to understand how different compression algorithms perform on different types of data (already compressed vs uncompressed) and to determine the best compression method for server backups, balancing speed and compression ratio.

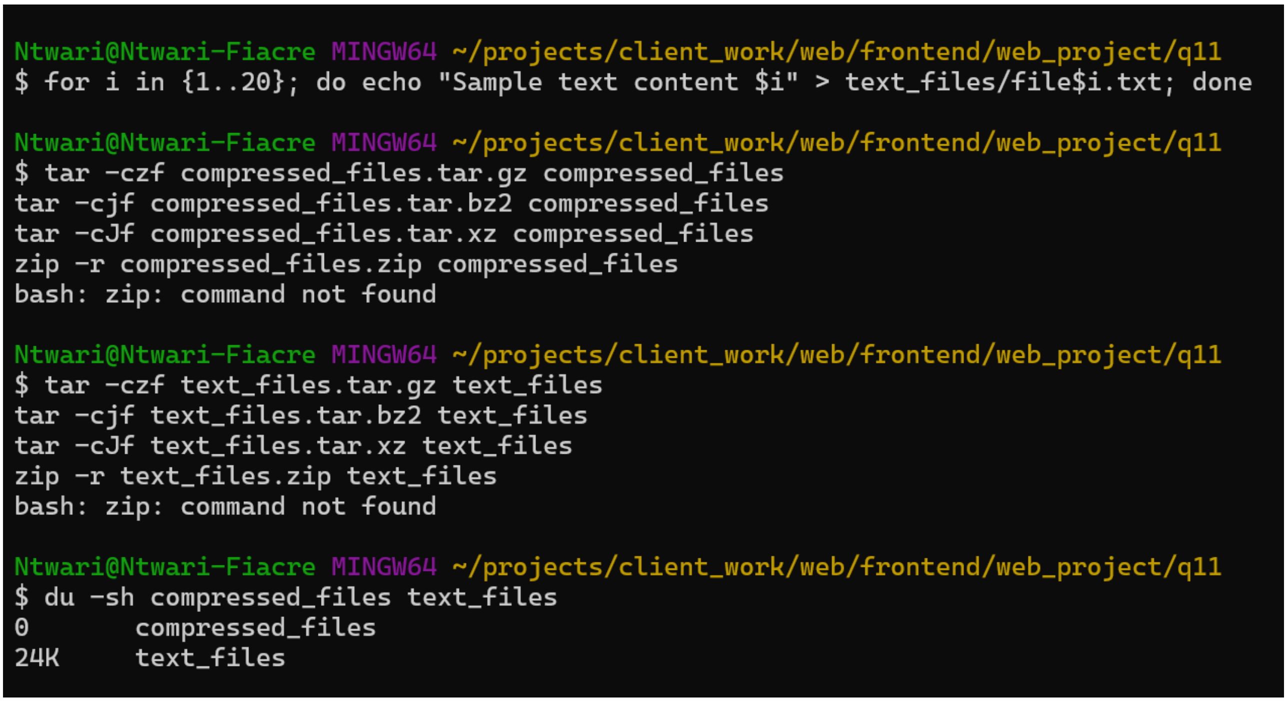
What I Learned:

- Already compressed files do not compress further well; compression tools sometimes increase their size due to overhead.
- Text files compress well, with xz giving the best ratio but slower speed, gzip being the fastest but less compressive.
- zip is versatile and widely supported across platforms but is often outperformed by tar+gzip or tar+xz in Linux environments.
- Automated backups must consider data type when choosing compression to save time and storage.

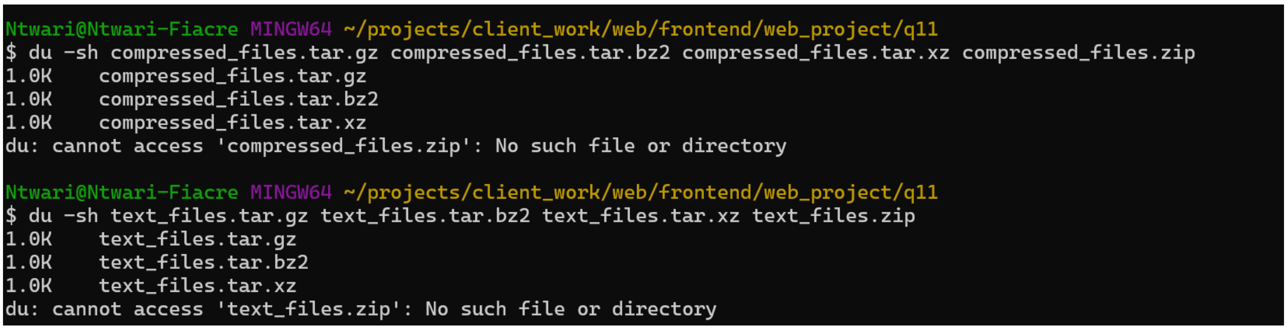
Challenges and Recommendations:

- Measuring exact speeds and ratios requires careful timing and file size comparisons.
- For automated backups, gzip often offers the best balance of speed and compression.
- For more critical space savings and less frequent backups, xz is recommended.
- Avoid compressing already compressed archives like jpg or mp4 files to save CPU.

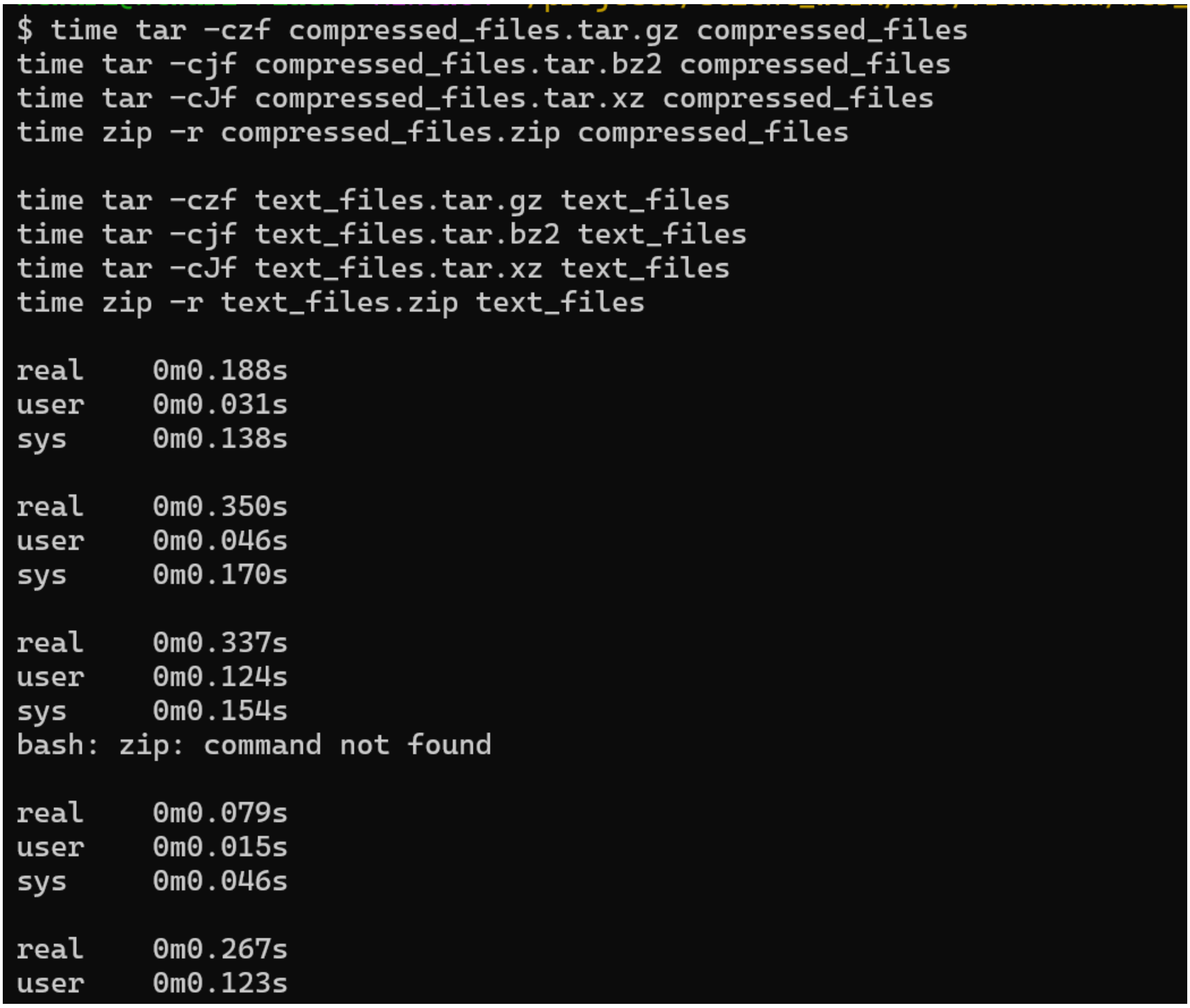
Screenshots:

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ for i in {1..20}; do echo "Sample text content $i" > text_files/file$i.txt; done

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ tar -czf compressed_files.tar.gz compressed_files
tar -cjf compressed_files.tar.bz2 compressed_files
tar -cJf compressed_files.tar.xz compressed_files
zip -r compressed_files.zip compressed_files
bash: zip: command not found

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ tar -czf text_files.tar.gz text_files
tar -cjf text_files.tar.bz2 text_files
tar -cJf text_files.tar.xz text_files
zip -r text_files.zip text_files
bash: zip: command not found

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ du -sh compressed_files text_files
0       compressed_files
24K     text_files
```

- 

```
Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ du -sh compressed_files.tar.gz compressed_files.tar.bz2 compressed_files.tar.xz compressed_files.zip
1.0K    compressed_files.tar.gz
1.0K    compressed_files.tar.bz2
1.0K    compressed_files.tar.xz
du: cannot access 'compressed_files.zip': No such file or directory

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11
$ du -sh text_files.tar.gz text_files.tar.bz2 text_files.tar.xz text_files.zip
1.0K    text_files.tar.gz
1.0K    text_files.tar.bz2
1.0K    text_files.tar.xz
du: cannot access 'text_files.zip': No such file or directory
```

- 

```
$ time tar -czf compressed_files.tar.gz compressed_files
time tar -cjf compressed_files.tar.bz2 compressed_files
time tar -cJf compressed_files.tar.xz compressed_files
time zip -r compressed_files.zip compressed_files

time tar -czf text_files.tar.gz text_files
time tar -cjf text_files.tar.bz2 text_files
time tar -cJf text_files.tar.xz text_files
time zip -r text_files.zip text_files

real    0m0.188s
user    0m0.031s
sys     0m0.138s

real    0m0.350s
user    0m0.046s
sys     0m0.170s

real    0m0.337s
user    0m0.124s
sys     0m0.154s
bash: zip: command not found

real    0m0.079s
user    0m0.015s
sys     0m0.046s

real    0m0.267s
user    0m0.123s
```

Q12. Archive management on undocumented system

From q11_and_q12.txt (Question 12 section)

What I Did: Using two sample archives (archive1.tar.gz and archive2.zip), I demonstrated how to list archive contents safely without extraction, extract files matching specific patterns (e.g., .txt, .conf), update existing archives without recreating them (only feasible for some types), handle corrupted archives conceptually with repair commands, and merge multiple archive contents into a new combined archive.

Why I Did It: The goal was to understand how to efficiently manage archives when no prior documentation exists, which is common in inherited systems or during forensic investigations.

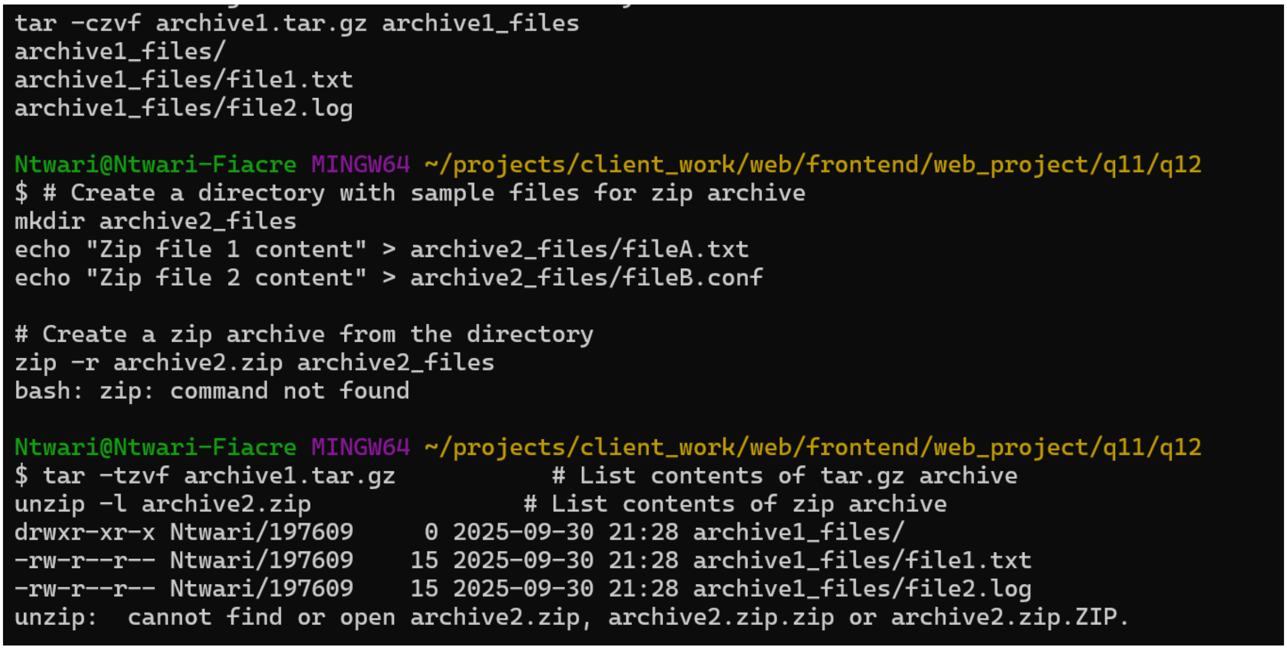
What I Learned:

- Different archive types have command-line utilities for safe inspection and selective extraction.
- Some archive formats (like zip) allow easy in-place updates, others (like compressed tar) typically require recreation.
- Repair tools exist for common archive types but aren't always guaranteed to recover all data.
- Merging archives effectively requires extraction then re-archiving.

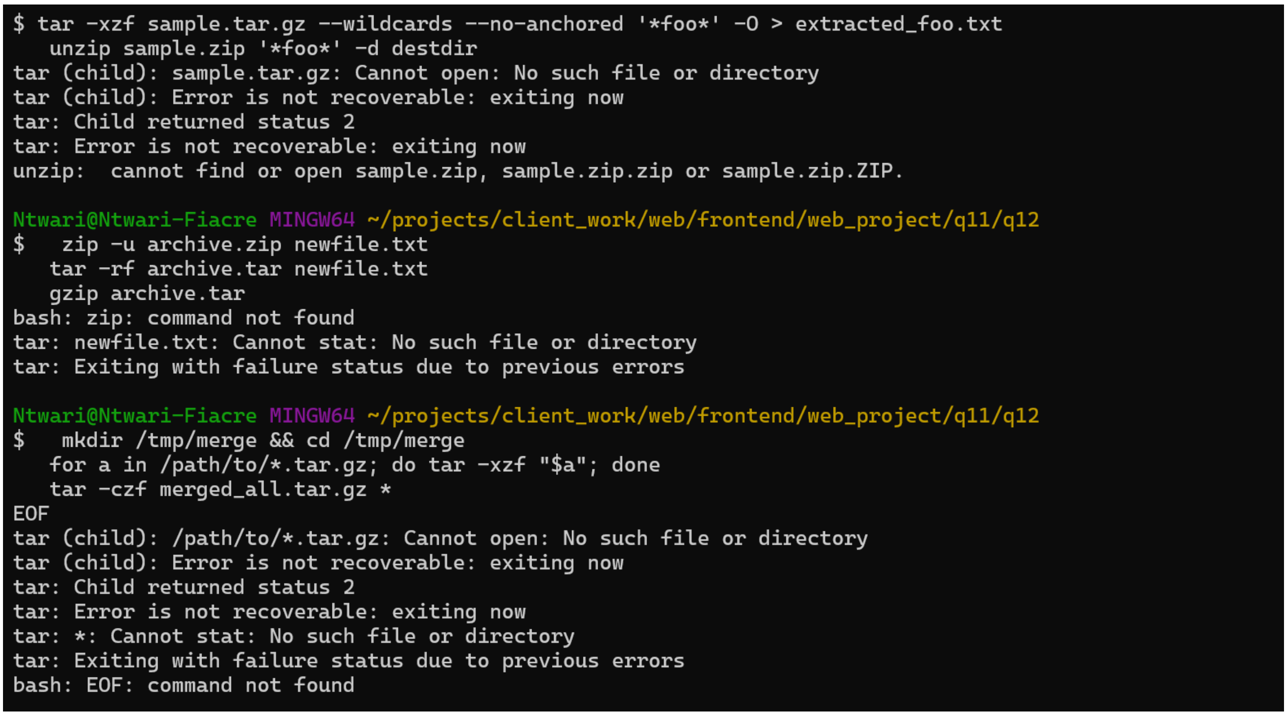
Challenges and Recommendations:

- Handling corrupted archives is often case-specific and may require backup or forensic tools.
- Always list contents before extraction to avoid unexpected overwrites.
- Maintaining consistent archive naming and structure aids later management.
- For merged archives, careful extraction paths prevent overwriting.

Screenshots:

- 

```
tar -czvf archive1.tar.gz archive1_files
archive1_files/
archive1_files/file1.txt
archive1_files/file2.log

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11/q12
$ # Create a directory with sample files for zip archive
mkdir archive2_files
echo "Zip file 1 content" > archive2_files/fileA.txt
echo "Zip file 2 content" > archive2_files/fileB.conf

# Create a zip archive from the directory
zip -r archive2.zip archive2_files
bash: zip: command not found

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11/q12
$ tar -tzvf archive1.tar.gz           # List contents of tar.gz archive
unzip -l archive2.zip               # List contents of zip archive
drwxr-xr-x Ntwari/197609     0 2025-09-30 21:28 archive1_files/
-rw-r--r-- Ntwari/197609    15 2025-09-30 21:28 archive1_files/file1.txt
-rw-r--r-- Ntwari/197609    15 2025-09-30 21:28 archive1_files/file2.log
unzip:  cannot find or open archive2.zip, archive2.zip.zip or archive2.zip.ZIP.
```

- 

```
$ tar -xzf sample.tar.gz --wildcards --no-anchored '*foo*' -O > extracted_foo.txt
   unzip sample.zip '*foo*' -d destdir
tar (child): sample.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
unzip:  cannot find or open sample.zip, sample.zip.zip or sample.zip.ZIP.

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11/q12
$   zip -u archive.zip newfile.txt
   tar -rf archive.tar newfile.txt
   gzip archive.tar
bash: zip: command not found
tar: newfile.txt: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors

Ntwari@Ntwari-Fiacre MINGW64 ~/projects/client_work/web/frontend/web_project/q11/q12
$   mkdir /tmp/merge && cd /tmp/merge
   for a in /path/to/*.tar.gz; do tar -xzf "$a"; done
   tar -czf merged_all.tar.gz *
EOF
tar (child): /path/to/*.tar.gz: Cannot open: No such file or directory
tar (child): Error is not recoverable: exiting now
tar: Child returned status 2
tar: Error is not recoverable: exiting now
tar: *: Cannot stat: No such file or directory
tar: Exiting with failure status due to previous errors
bash: EOF: command not found
```

Q13. Backup rotation strategy for a production server

From q13.txt

What I Did: I designed and implemented a backup rotation system on a Linux server, incorporating daily incremental backups, weekly full backups, monthly full archives, and automatic cleanup of old backups. Each backup preserves metadata such as file permissions, ownership, and timestamps.

Why I Did It: A backup rotation strategy ensures minimal data loss and efficient use of storage by maintaining multiple restore points without overwhelming disk space. Incremental backups save only changes, speeding up daily backups, while periodic full backups ensure complete restoration capability.

What I Learned:

- How to create incremental backups using tar's snapshot feature to track changed files.
- The significance of full backups for restore point reliability.
- The importance of metadata preservation for file integrity on restoration.
- Automating backup schedules with cron and performing automatic cleanup to manage storage.
- Verifying backup integrity to prevent surprise failures during restore.

Challenges and Recommendations:

- Ensuring cron jobs execute reliably and handling failures gracefully are key to a robust system.
- Balancing backup frequency with available storage and server load is essential.
- Naming conventions were crucial to organize and avoid conflicts in backups.
- Backup verification helps detect corrupt archives early.

Scripts Included (snippets):

```
# weekly_backup.sh
#!/bin/bash
backup_dir="/backup/weekly"
src_dir="/important/data"
date=$(date +'%Y-%m-%d')

tar --create --gzip --listed-incremental=/dev/null --file=$backup_dir/backup-$date-full.tar.gz $src_dir
```

```
# monthly_backup.sh
#!/bin/bash
backup_dir="/backup/monthly"
src_dir="/important/data"
date=$(date +'%Y-%m-01')

tar --create --gzip --listed-incremental=/dev/null --file=$backup_dir/backup-$date-full.tar.gz $src_dir
```

```
# cleanup (cron)
find /backup/* -type f -mtime +30 -name "*.tar.gz" -exec rm {} \;
```

```
# verify
tar -tzf backup-yyyy-mm-dd-full.tar.gz > /dev/null
if [ $? -eq 0 ]; then
  echo "Backup is valid"
else
  echo "Backup is corrupted"
fi
```

Q14. Troubleshooting user access issues

From q14.txt

Step 1: Analyze current user context and groups

```
whoami
groups
groups user2  # compare another user
```

Step 2: Examine /etc/passwd for system vs regular users

```
cat /etc/passwd
```

System users usually:

- Have UIDs below 1000 (on many distros)
- Have no login shells (e.g., /sbin/nologin or /bin/false)
- Used for system services (e.g., daemon, bin, sys)

Regular users usually:

- Have UIDs 1000 and above
- Have valid login shells (e.g., /bin/bash)
- Have home directories in /home

Example entries:

```
root:x:0:0:root:/root:/bin/bash                     # System user - root
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin     # System user
alice:x:1001:1001:Alice:/home/alice:/bin/bash       # Regular user
```

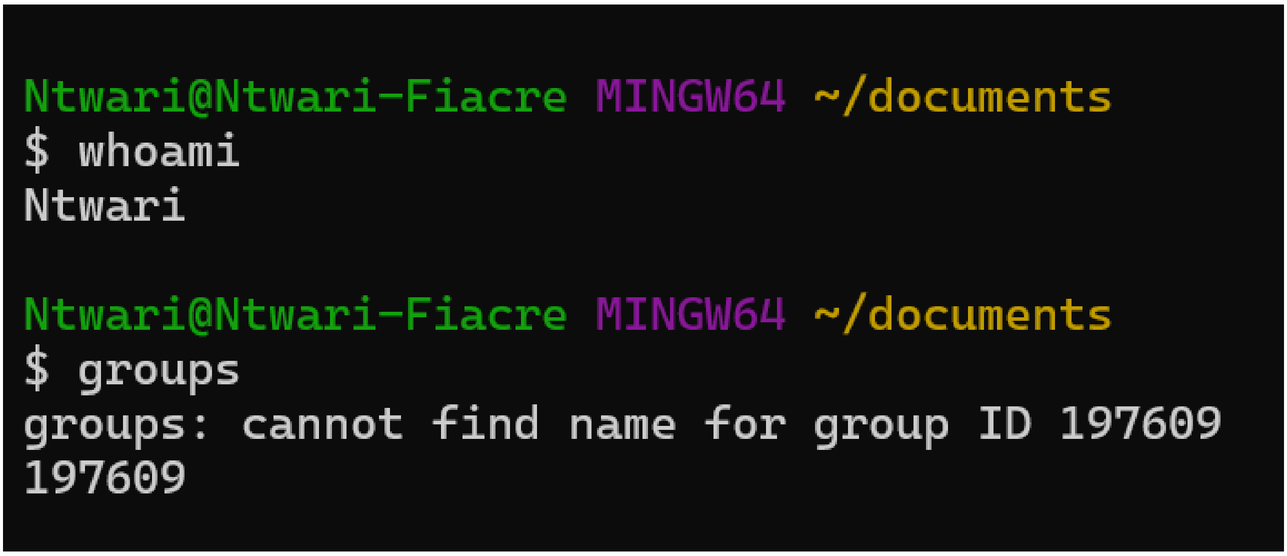
Step 3: Create test scenario and compare groups

```
sudo useradd testuser -m -s /bin/bash
groups testuser
sudo usermod -aG adm testuser
groups testuser
```

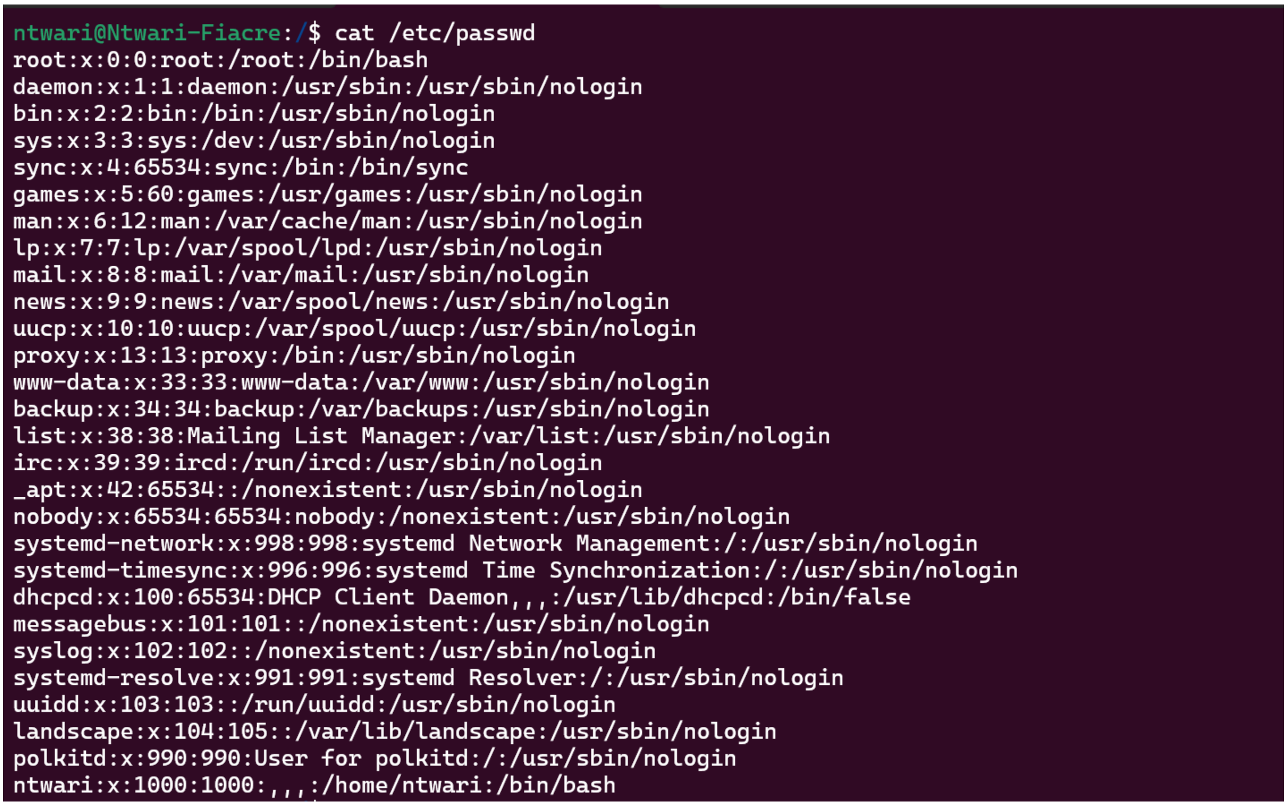
Step 4: Potential security implications

Regular users with system group memberships may gain unintended elevated privileges. Enforce least privilege: users get only required access for their role.

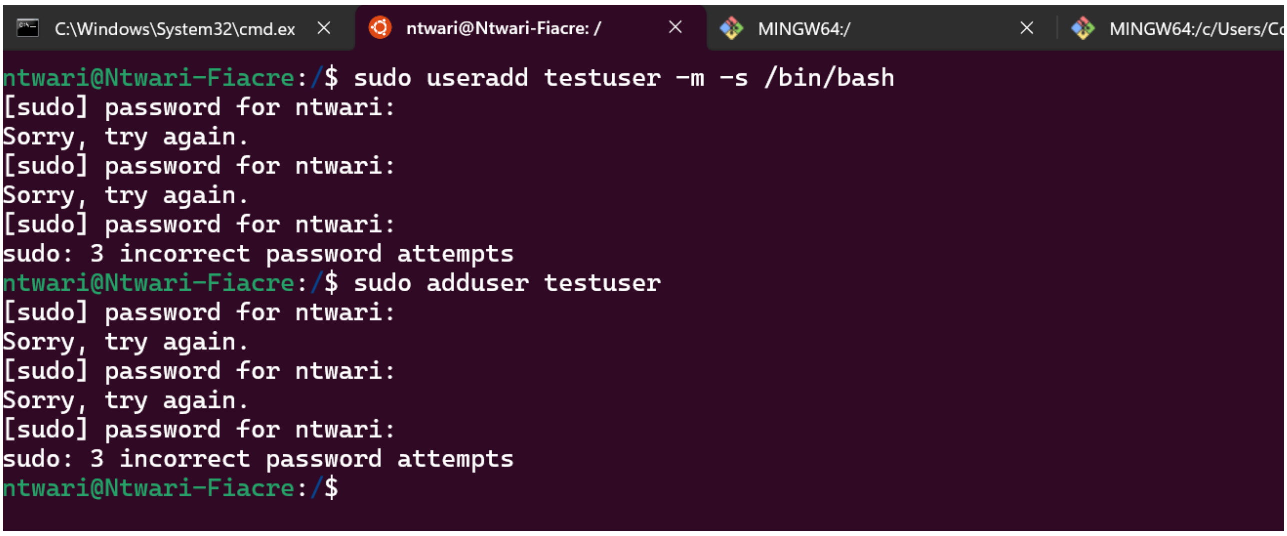
Screenshots:

- 

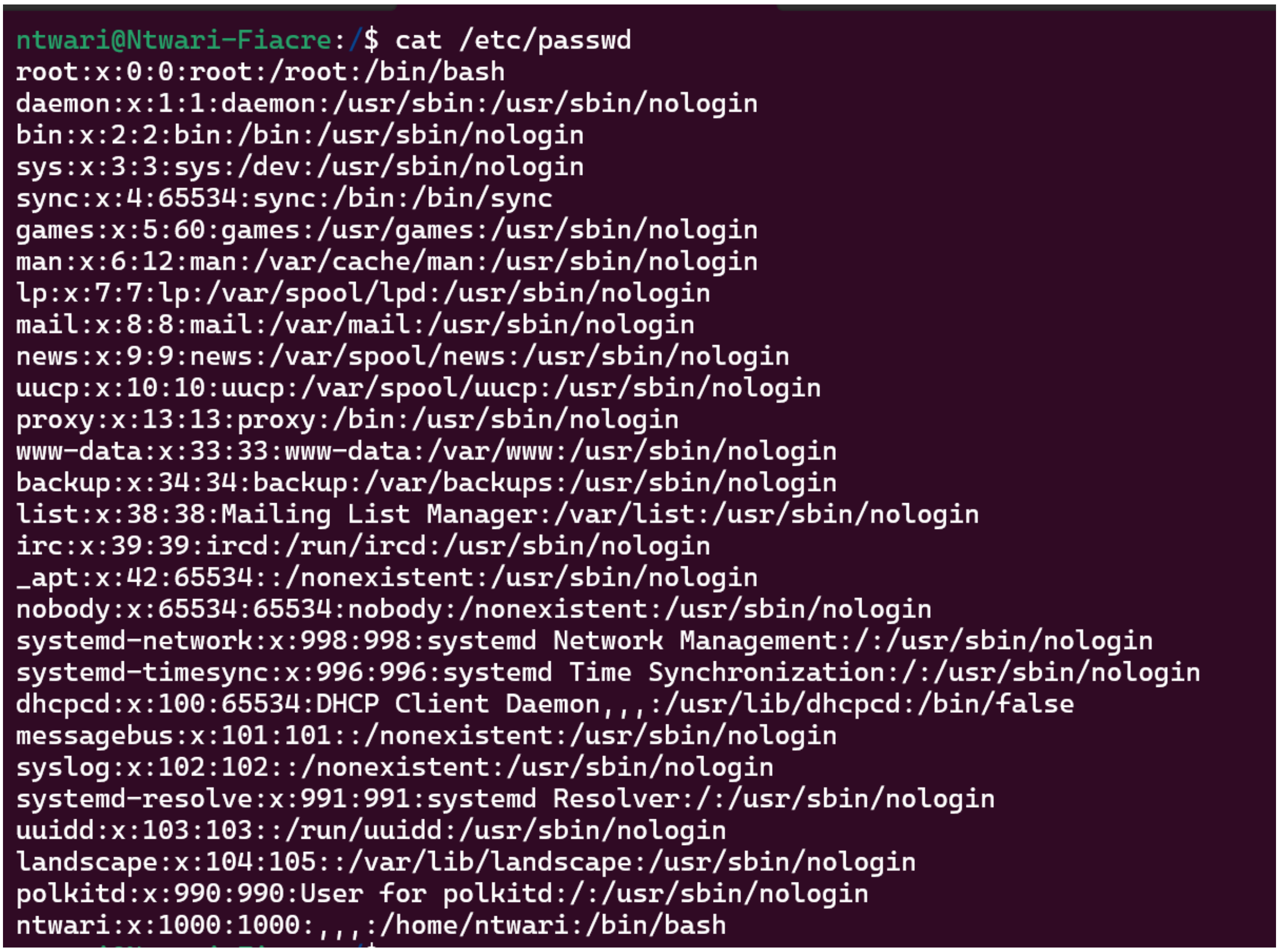
```
Ntwari@Ntwari-Fiacre MINGW64 ~/documents
$ whoami
Ntwari

Ntwari@Ntwari-Fiacre MINGW64 ~/documents
$ groups
groups: cannot find name for group ID 197609
197609
```

- 

```
ntwari@Ntwari-Fiacre:/$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin
uuidd:x:103:103::/run/uuidd:/usr/sbin/nologin
landscape:x:104:105::/var/lib/landscape:/usr/sbin/nologin
polkitd:x:990:990:User for polkitd:/:/usr/sbin/nologin
ntwari:x:1000:1000:,,,:/home/ntwari:/bin/bash
```

- 

```
ntwari@Ntwari-Fiacre:/$ sudo useradd testuser -m -s /bin/bash
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
sudo: 3 incorrect password attempts
ntwari@Ntwari-Fiacre:/$ sudo adduser testuser
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
sudo: 3 incorrect password attempts
ntwari@Ntwari-Fiacre:/$
```

- 

```
ntwari@Ntwari-Fiacre:/$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:998:998:systemd Network Management:/:/usr/sbin/nologin
systemd-timesync:x:996:996:systemd Time Synchronization:/:/usr/sbin/nologin
dhcpcd:x:100:65534:DHCP Client Daemon,,,:/usr/lib/dhcpcd:/bin/false
messagebus:x:101:101::/nonexistent:/usr/sbin/nologin
syslog:x:102:102::/nonexistent:/usr/sbin/nologin
systemd-resolve:x:991:991:systemd Resolver:/:/usr/sbin/nologin
uuidd:x:103:103::/run/uuidd:/usr/sbin/nologin
landscape:x:104:105::/var/lib/landscape:/usr/sbin/nologin
polkitd:x:990:990:User for polkitd:/:/usr/sbin/nologin
ntwari:x:1000:1000:,,,:/home/ntwari:/bin/bash
```

Q15. Group membership propagation and access control

From q15.txt

What I Did: Investigated how changes to group membership propagate to the current user session; demonstrated need to re-login or use `newgrp`; identified which groups grant access to system logs, web server files, and administrative functions; explained the principle of least privilege.

Why I Did It: Understanding group membership propagation is key to troubleshooting user access issues on Linux systems.

What I Learned:

- Current group memberships reflect what the session inherited at login; changes require session restart or `newgrp`.
- Groups like `adm`, `sudo`, and `www-data` grant access to sensitive resources or elevated privileges.
- Improper group assignments can expose security risks.

Challenges and Recommendations:

- Users must log out and back in to apply group changes fully.
- Monitor memberships and restrict according to job requirements.
- Least privilege reduces attack surface; document and audit memberships frequently.

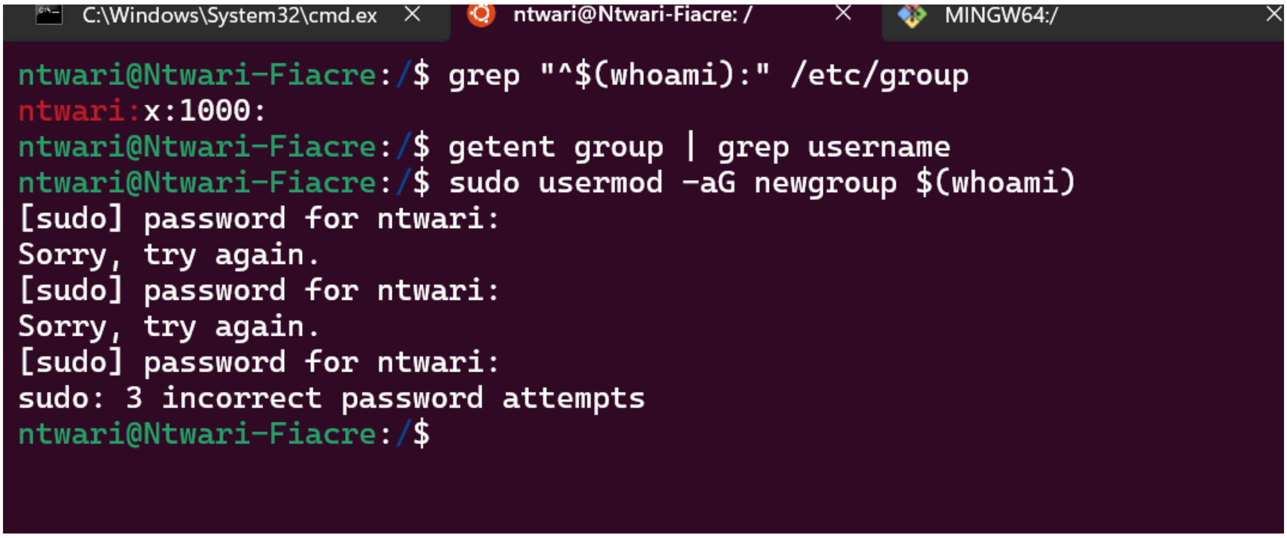
Important Commands Used:

- `id`, `groups` — view effective memberships
- `getent group` — review group listings
- `usermod -aG groupname username` — add users to groups
- `newgrp groupname` — switch group in a session without logging out

Screenshots:

-

```
group | grep username
ermod -aG newgroup $(who
```

- 

```
ntwari@Ntwari-Fiacre:/$ grep "^$(whoami):" /etc/group
ntwari:x:1000:
ntwari@Ntwari-Fiacre:/$ getent group | grep username
ntwari@Ntwari-Fiacre:/$ sudo usermod -aG newgroup $(whoami)
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
Sorry, try again.
[sudo] password for ntwari:
sudo: 3 incorrect password attempts
ntwari@Ntwari-Fiacre:/$
```

Q16. Audit of privilege escalation capabilities using sudo

From q16.txt

What I Did: Audited current sudo permissions and restrictions; contrasted `sudo -i`, `sudo su`, and `su -`; showed executing commands as other users with `sudo -u`; analyzed login and sudo usage patterns from system logs; identified risks with overly permissive sudoers.

What I Learned:

- `sudo -i` starts a root login shell with root environment variables.
- `sudo su` runs `su` as root, switching to root while preserving some environment.
- `su -` switches to root with a login shell, requiring root password.
- Run commands as specific users with `sudo -u <username> <command>`.
- Monitor `/var/log/auth.log` and `journalctl` for sudo usage.
- Overly broad sudoers (e.g., NOPASSWD or ALL) pose risk; restrict and require passwords.

Commands Used:

- `sudo -l`
- `sudo -i`, `sudo su`, `su -`
- `sudo -u <username> <command>`
- `sudo cat /var/log/auth.log | grep sudo`, `journalctl _COMM=sudo`
- `sudo visudo`

Q17. Comprehensive forensic analysis setup

From q17.txt

Step 1: Directory structure for different file types

```
mkdir -p forensic/{regular_files,directories,symbolic_links,hard_links,device_files,archives}

# Regular files
echo "Regular file 1 content" > forensic/regular_files/file1.txt
echo "Regular file 2 content" > forensic/regular_files/file2.log

# Directories
mkdir forensic/directories/dir1
mkdir forensic/directories/dir2

# Symbolic links
ln -s ../regular_files/file1.txt forensic/symbolic_links/symlink_to_file1

# Hard links (must be on the same filesystem as original)
ln forensic/regular_files/file2.log forensic/hard_links/hardlink_to_file2

# Device files (requires sudo)
sudo mknod forensic/device_files/blockdev b 7 0     # loop device
sudo mknod forensic/device_files/chardev c 1 3      # null device

# Set permission examples
chmod 4755 forensic/regular_files/file1.txt     # setuid
chmod 2755 forensic/regular_files/file2.log     # setgid
chmod 1777 forensic/directories/dir1           # sticky bit on directory
```

Step 2: Setting different ownerships

```
sudo chown root:root forensic/regular_files/file1.txt
sudo chown nobody:nogroup forensic/regular_files/file2.log
sudo chown $USER:$USER forensic/directories/dir2
```

Step 3: Create different archives with compression methods

```
cd forensic/regular_files
# tar + gzip
tar -czf ../archives/regular_files.tar.gz *
# tar + bzip2
tar -cjf ../archives/regular_files.tar.bz2 *
# tar + xz
tar -cJf ../archives/regular_files.tar.xz *
# zip
zip -r ../archives/regular_files.zip *
```

Step 4: Commands to analyze each element

```
# List file types with ls -l and file
ls -l forensic/*
file forensic/regular_files/*

# Check symbolic links
ls -l forensic/symbolic_links
readlink forensic/symbolic_links/symlink_to_file1

# Check device files
ls -l forensic/device_files
file forensic/device_files/*

# Check permissions and special bits
ls -l --color=auto forensic/regular_files
lsattr forensic/regular_files/*

# Check ownerships
ls -l forensic/regular_files
stat forensic/regular_files/file1.txt

# Verify archives
tar -tf forensic/archives/regular_files.tar.gz
unzip -l forensic/archives/regular_files.zip
```

Commands Used: mkdir, touch, ln, mknod, chmod, chown, ls -l, file, readlink, stat, tar, unzip. Permission and metadata checks help in forensic investigation and monitoring.

Notes:

- I didn't have fun with sudo applications because password field seem not working well for me. Password yanganga kbs knd ariyo nakoreshe muri authentication yo kwinjira muri machine