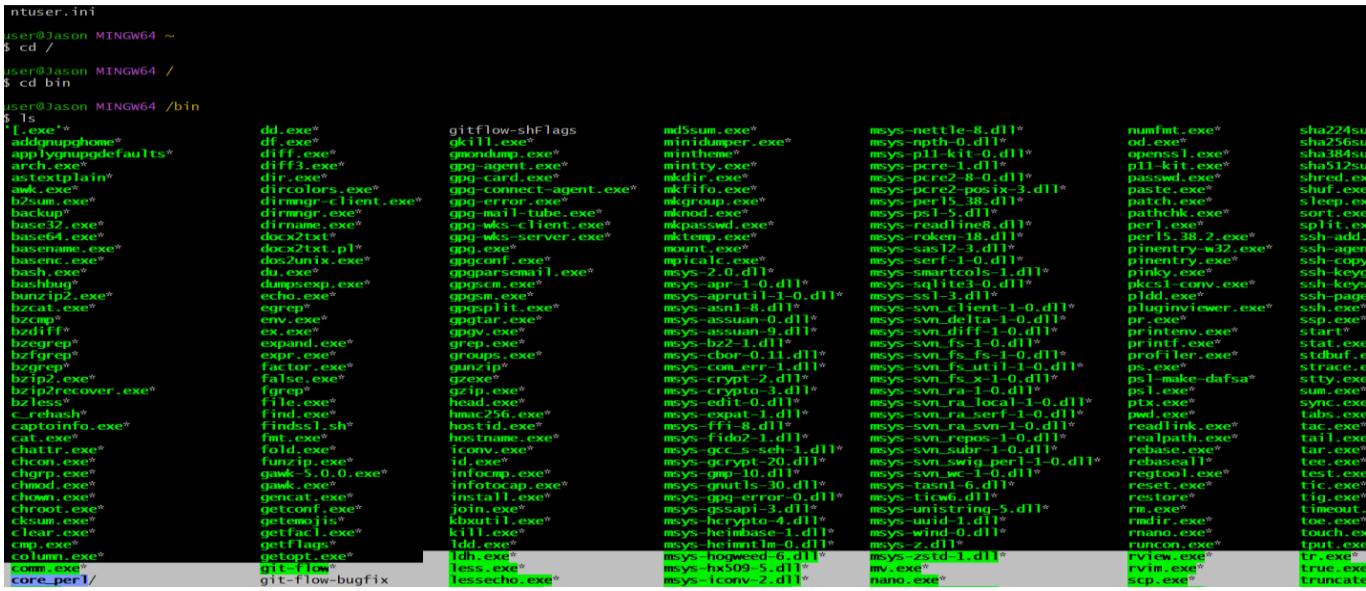
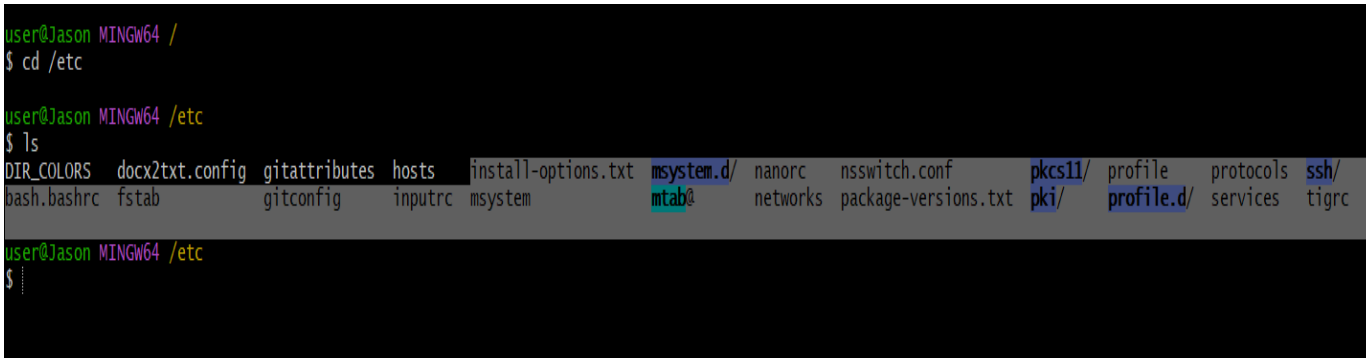
Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Linux_Assignment2

Qn1:_ **/bin → Essential binaries**. Core commands (e.g., ls, cp) live here and could be replaced with malicious versions.

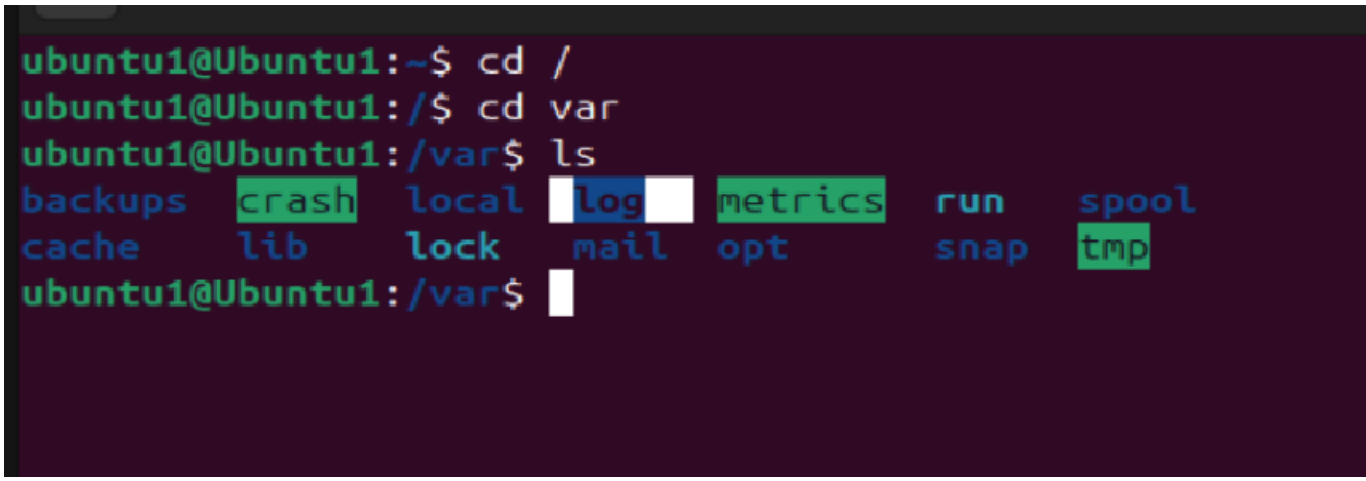


_ **/etc → System configuration files**. Attackers may modify settings here (like /etc/passwd, network configs).

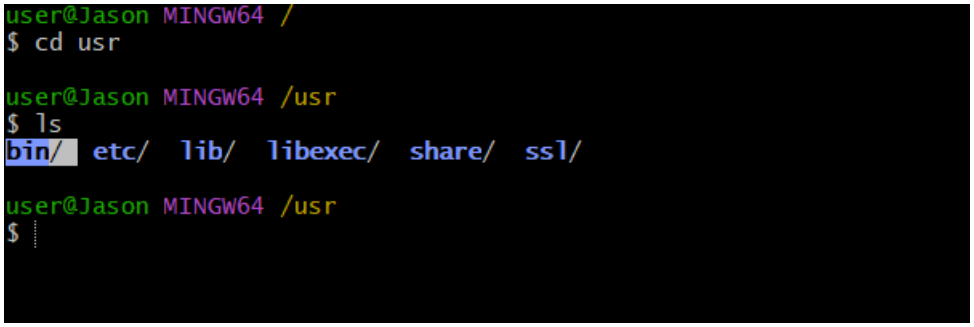


_ **/var → Log files**. Logs are usually in /var/log, showing evidence of intrusions. (using Ubuntu because in gitbush it wasn't found)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

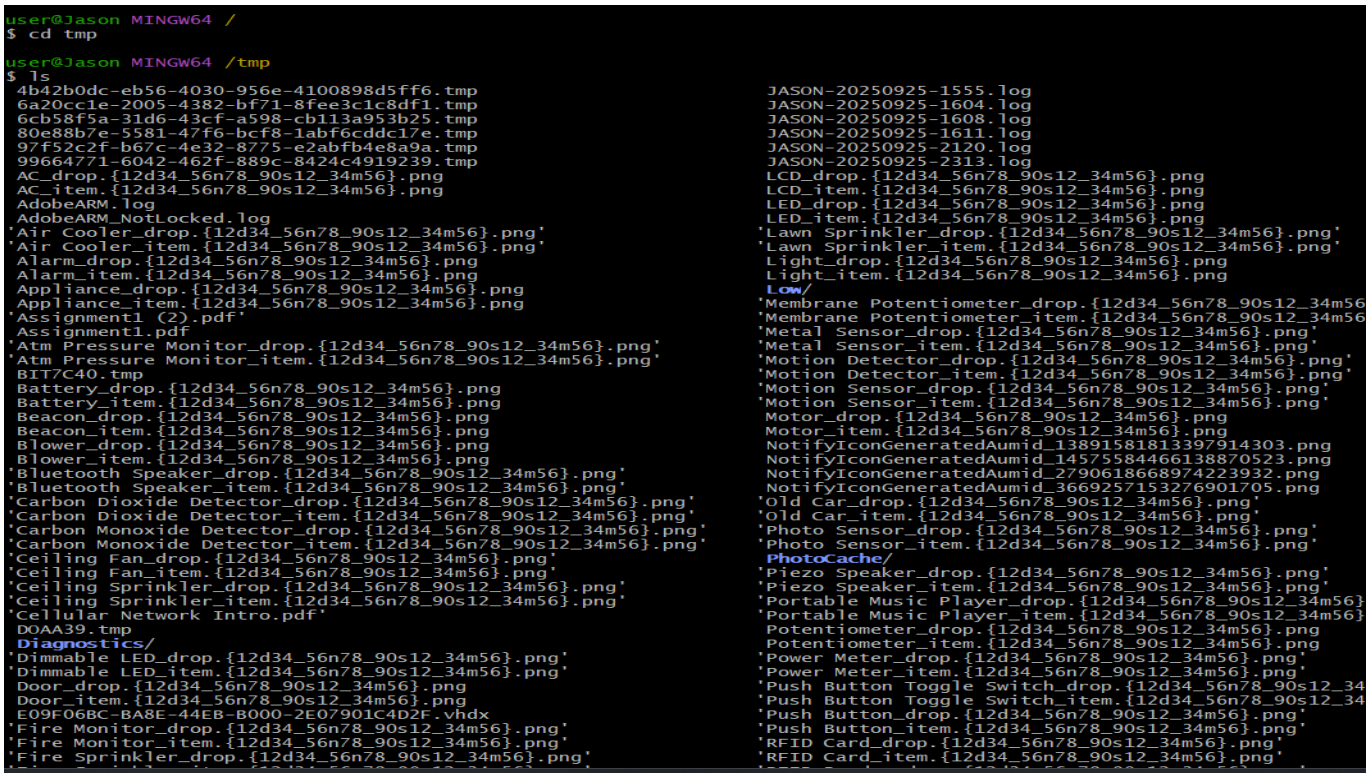


__ **/usr** → Additional binaries and software; less critical than /bin but can hold apps an attacker may alter.

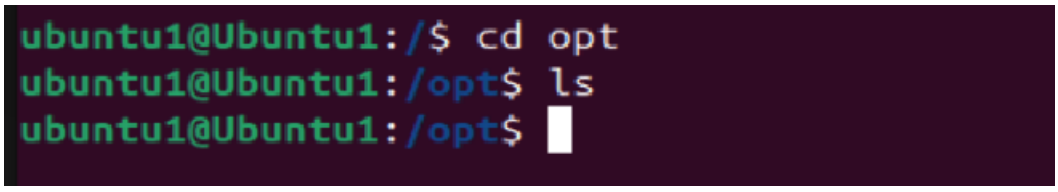


__/**tmp** → Temporary files; often used by attackers to store scripts or payloads.

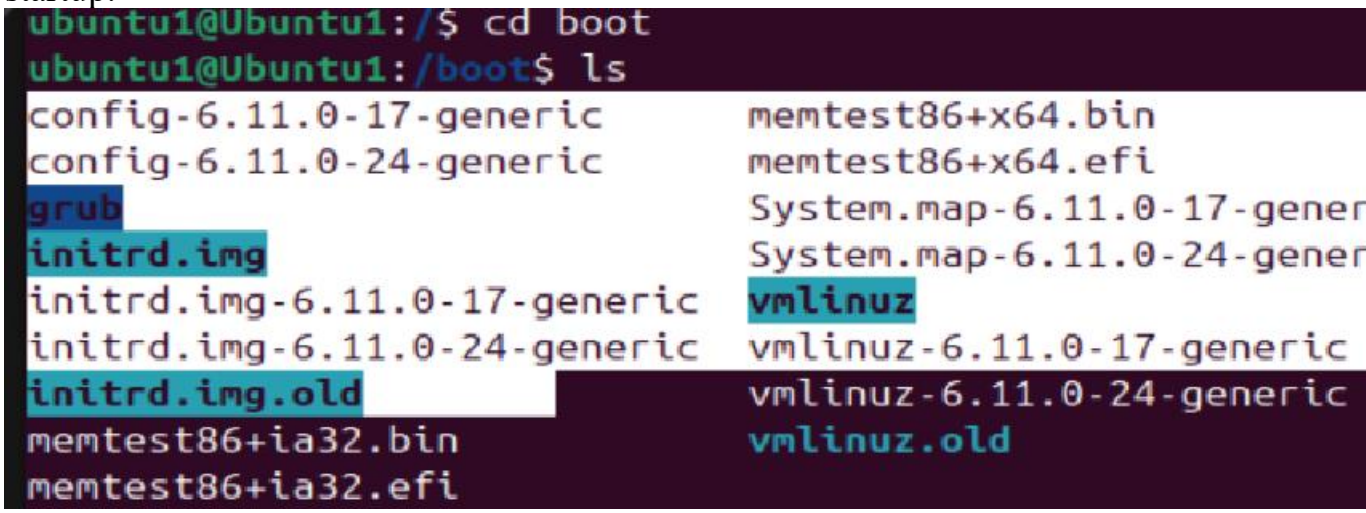
Name: Irumva Jason ID:27241 Date:Fri/26th/2025



__ **/opt** → Optional software; could hold installed malware if attackers drop custom tools



__ **/boot** → Bootloader and kernel; tampering here can compromise system startup.



__ **/home** → User files; attackers might modify personal scripts or plant backdoors.

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Qn2:Creation of Directory with Minimum Code
Using: mkdir -p
~/projects/{client_work/{web/{frontend,backend,database},mobile/{ios,android}},
personal/{experiments,archive},shared/{templates,resources}}
To verify the structure: ls -R ~/projects

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

tree ~/projects

```
user@Jason MINGW64 ~
$ mkdir -p ~/projects/{client_work/{web/{frontend,backend,database},mobile/{ios,android}},perso

user@Jason MINGW64 ~
$ tree ~/projects
bash: tree: command not found

user@Jason MINGW64 ~
$ ls -R ~/projects
tree ~/projects
/c/Users/user/projects:
client_work/  personal/  shared/

/c/Users/user/projects/client_work:
mobile/  web/

/c/Users/user/projects/client_work/mobile:
android/  ios/

/c/Users/user/projects/client_work/mobile/android:

/c/Users/user/projects/client_work/mobile/ios:

/c/Users/user/projects/client_work/web:
backend/  database/  frontend/

/c/Users/user/projects/client_work/web/backend:

/c/Users/user/projects/client_work/web/database:

/c/Users/user/projects/client_work/web/frontend:

/c/Users/user/projects/personal:
archive/  experiments/

/c/Users/user/projects/personal/archive:

/c/Users/user/projects/personal/experiments:

/c/Users/user/projects/shared:
resources/  templates/

/c/Users/user/projects/shared/resources:

/c/Users/user/projects/shared/templates:
```

Qn3: Navigating to the Directory: ~/projects/client_work/web/frontend

```
user@Jason MINGW64 ~
$ cd projects/client_work/web/frontend

user@Jason MINGW64 ~/projects/client_work/web/frontend
$
```

___From ~/projects/client_work/web/frontend → ~/projects/personal/experiments

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
user@Jason MINGW64 ~/projects/client_work/web/frontend
$ cd ../../../personal/experiments

user@Jason MINGW64 ~/projects/personal/experiments
$ pwd
/c/Users/user/projects/personal/experiments
```

___From ~/projects/personal/experiments → ~/projects/shared/templates

```
user@Jason MINGW64 ~/projects/personal/experiments
$ cd ../../shared/templates

user@Jason MINGW64 ~/projects/shared/templates
$ pwd
/c/Users/user/projects/shared/templates
```

___From ~/projects/shared/templates → back to ~/projects/client_work/web/frontend(Initially where I was).

```
user@Jason MINGW64 ~/projects/shared/templates
$ cd ../../client_work/web/frontend

user@Jason MINGW64 ~/projects/client_work/web/frontend
$ pwd
/c/Users/user/projects/client_work/web/frontend
```

Qn4: Using the “mkdir -p” I created and navigated to my Parents directory web_project and

created there bas project folders css,js and backups

```
user@Jason MINGW64 ~
$ cd web_Project

user@Jason MINGW64 ~/web_Project
$ ls
backups/  css/  js/

user@Jason MINGW64 ~/web_Project
```

__I created 15 HTML files in the parents directory web_Project

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
user@Jason MINGW64 ~/web_Project
$ touch index.html about.html contact.html \
      page_{001..012}.html

user@Jason MINGW64 ~/web_Project
$ ls
about.html  contact.html  index.html  page_001.html  page_003.html  page_005.html  page_007.ht
backups/    css/          js/          page_002.html  page_004.html  page_006.html  page_008.ht

user@Jason MINGW64 ~/web_Project
$
```

__8 css files as named in the question are also created in there respective directory css

```
user@Jason MINGW64 ~/web_Project
$ cd css

user@Jason MINGW64 ~/web_Project/css
$ touch main.css reset.css theme_light.css theme_dark.css \
      mobile.css tablet.css desktop.css print.css

user@Jason MINGW64 ~/web_Project/css
$ ls
desktop.css  main.css  mobile.css  print.css  reset.css  tablet.css  theme_dark.css  theme_lig

user@Jason MINGW64 ~/web_Project/css
$
```

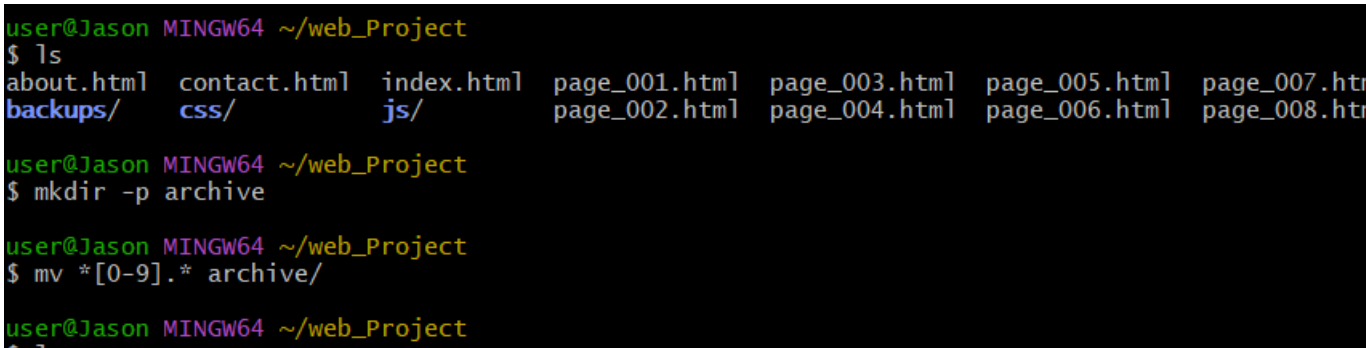
__I also navigated back to the parent directory and switched to the js directory for javascript files and created the 6 js files as required with respective naming

```
user@Jason MINGW64 ~/web_Project
$ cd js

user@Jason MINGW64 ~/web_Project/js
$ touch script.js script_util.js script_config.js \
      util.js util_helper.js config.js

user@Jason MINGW64 ~/web_Project/js
$ ls
config.js  script.js  script_config.js  script_util.js  util.js  util_helper.js

user@Jason MINGW64 ~/web_Project/js
```

__Back to the parent dierectory using “cd ..” I navigated to backups directory and also created 20 required files as required

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Qn5: Moving all files ending in numbers (like page_001.html) to archive/



__Copying all CSS files **except those containing mobile or tablet** → desktop/

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
user@Jason MINGW64 ~/web_Project/css
$    mkdir -p desktop

user@Jason MINGW64 ~/web_Project/css
$ ls
desktop/  desktop.css  main.css  mobile.css  print.css  reset.css  tablet.cs

user@Jason MINGW64 ~/web_Project/css
$ for file in *.css; do
  if [[ "$file" != *mobile* && "$file" != *tablet* ]]; then
    cp "$file" desktop/
  fi
done

user@Jason MINGW64 ~/web_Project/css
$ cd desktop

user@Jason MINGW64 ~/web_Project/css/desktop
$ ls
desktop.css  main.css  print.css  reset.css  theme_dark.css  theme_light.css

user@Jason MINGW64 ~/web_Project/css/desktop
$
```

___Listing only files with **exactly 3 characters before the dot**

```
user@Jason MINGW64 ~/web_Project
$ ls ???.*
gym.txt  man.js  tim.html

user@Jason MINGW64 ~/web_Project
$
```

___Finding files starting with any **consonant** (not a vowel)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
user@Jason MINGW64 ~/web_Project
$ ls [bcdfghjklmnpqrstvwxyzBCDFGHJKLMNPQRSTVWXYZ]*
contact.html  gym.txt  man.js  tim.html

backups:
a1.bak  a2.bak  a3.bak  a4.bak  a5.bak  b1.tmp  b2.tmp  b3.tmp  b4.tmp  b5.tmp  c1.old  c2.old

css:
desktop/  desktop.css  main.css  mobile.css  print.css  reset.css  tablet.css  theme_dark.css

js:
config.js  script.js  script_config.js  script_util.js  util.js  util_helper.js

user@Jason MINGW64 ~/web_Project
$
```

___Identifying files where the extension has exactly 2 characters

```
user@Jason MINGW64 ~/web_Project
$ ls *.[a-zA-Z][a-zA-Z]
man.js

user@Jason MINGW64 ~/web_Project
$ ls js *.[a-zA-Z][a-zA-Z]
man.js

js:
config.js  script.js  script_config.js  script_util.js  util.js  util_helper.js

user@Jason MINGW64 ~/web_Project
$
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Qn6: Log files for every day of 1st Qaurter 2024

```
user@Jason MINGW64 ~
$ mkdir -p logs

user@Jason MINGW64 ~
$ cd logs

user@Jason MINGW64 ~/logs
$ for month in {01..03}; do
  for day in {01..31}; do
    if [[ $month -eq 02 && $day -gt 29 ]]; then continue; fi
    if [[ $month =~ 04|06|09|11 && $day -gt 30 ]]; then continue; fi
    touch log_2024-$month-$day.txt
  done
done
bash: [[: 08: value too great for base (error token is "08")
bash: [[: 09: value too great for base (error token is "09")

user@Jason MINGW64 ~/logs
$ ls
log_2024-01-01.txt  log_2024-01-11.txt  log_2024-01-21.txt  log_2024-01-31.txt  log_2024-02-10.txt  log_2024-02-20.txt  log_2024
log_2024-01-02.txt  log_2024-01-12.txt  log_2024-01-22.txt  log_2024-02-01.txt  log_2024-02-11.txt  log_2024-02-21.txt  log_2024
log_2024-01-03.txt  log_2024-01-13.txt  log_2024-01-23.txt  log_2024-02-02.txt  log_2024-02-12.txt  log_2024-02-22.txt  log_2024
log_2024-01-04.txt  log_2024-01-14.txt  log_2024-01-24.txt  log_2024-02-03.txt  log_2024-02-13.txt  log_2024-02-23.txt  log_2024
log_2024-01-05.txt  log_2024-01-15.txt  log_2024-01-25.txt  log_2024-02-04.txt  log_2024-02-14.txt  log_2024-02-24.txt  log_2024
log_2024-01-06.txt  log_2024-01-16.txt  log_2024-01-26.txt  log_2024-02-05.txt  log_2024-02-15.txt  log_2024-02-25.txt  log_2024
log_2024-01-07.txt  log_2024-01-17.txt  log_2024-01-27.txt  log_2024-02-06.txt  log_2024-02-16.txt  log_2024-02-26.txt  log_2024
log_2024-01-08.txt  log_2024-01-18.txt  log_2024-01-28.txt  log_2024-02-07.txt  log_2024-02-17.txt  log_2024-02-27.txt  log_2024
log_2024-01-09.txt  log_2024-01-19.txt  log_2024-01-29.txt  log_2024-02-08.txt  log_2024-02-18.txt  log_2024-02-28.txt  log_2024
log_2024-01-10.txt  log_2024-01-20.txt  log_2024-01-30.txt  log_2024-02-09.txt  log_2024-02-19.txt  log_2024-02-29.txt  log_2024

user@Jason MINGW64 ~/logs
$
```

__Configuration files for dev, staging, production across 3 services

```
user@Jason MINGW64 ~
$ mkdir -p configs

user@Jason MINGW64 ~
$ touch configs/{web,api,db}_{dev,staging,prod}.conf

user@Jason MINGW64 ~
$ ls configs
api_dev.conf   api_staging.conf  db_prod.conf     web_dev.conf   web_staging.conf
api_prod.conf  db_dev.conf       db_staging.conf  web_prod.conf
```

__Test files combining letters A-C with numbers 10-12 plus suffixes input and output

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
user@Jason MINGW64 ~
$ mkdir -p tests

user@Jason MINGW64 ~
$ touch tests/{A,B,C}_{10..12}_{input,output}.txt

user@Jason MINGW64 ~
$ ls tests
A_10_input.txt   A_11_output.txt  B_10_input.txt   B_11_output.txt  C_10_input.txt   C_11_outpu
A_10_output.txt  A_12_input.txt   B_10_output.txt  B_12_input.txt   C_10_output.txt  C_12_inpu
A_11_input.txt   A_12_output.txt  B_11_input.txt   B_12_output.txt  C_11_input.txt   C_12_outpu
```

QN7: Creating Two similar files with the same content

```
user@Jason MINGW64 ~/prac
$ #Linux line ending #lf

user@Jason MINGW64 ~/prac
$ echo -e "server=localhost\nport=8080\nmode=dev" > config_linux.txt

user@Jason MINGW64 ~/prac
$ #Windows line endings

user@Jason MINGW64 ~/prac
$ echo -e "server=localhost\r\nport=8080\r\nmode=dev" > config_windows.txt
```

```
user@Jason MINGW64 ~/prac
$ cat config_windows.txt  config_linux.txt


server=localhost
port=8080
mode=dev
server=localhost
port=8080
mode=dev

user@Jason MINGW64 ~/prac
$
```

__Comparison:

Diff: Shows the lines as different even though visually identical.

Reason: it detects line ending differences (\n vs \r\n).

```
user@Jason MINGW64 ~/prac
$ diff config_linux.txt config_windows.txt
1,2c1,2
< server=localhost
< port=8080
---
> server=localhost
> port=8080
```

___cmp: Reports the first differing byte position.

More “raw,” byte-level comparison, doesn’t care about lines.

```
user@Jason MINGW64 ~/prac
$ cmp config_linux.txt config_windows.txt
config_linux.txt config_windows.txt differ: char 17, line 1

user@Jason MINGW64 ~/prac
```

comm:

Expects sorted input line by line.

Treats the whole line as different because of the extra \r.

```
user@Jason MINGW64 ~/prac
$ comm config_linux.txt config_windows.txt
server=localhost
comm: file 1 is not in sorted order
port=8080
mode=dev
        server=localhost
comm: file 2 is not in sorted order
        port=8080
        mode=dev
comm: input is not in sorted order
```

Lesson on cross-platform compatibility

Windows uses **CRLF (\r\n)** line endings.

Linux/Unix uses **LF (\n)**.

Tools see these as differences, even if text looks identical.

This is why configs, scripts, or code may break when moved between systems.

Qn8: Creating Test Environment

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ mkdir -p audit_env/{docs,logs,tmp,hidden,emp
ubuntu1@Ubuntu1:~$ cd audit_env
ubuntu1@Ubuntu1:~/audit_env$ echo "Report" > docs/report.txt
ubuntu1@Ubuntu1:~/audit_env$ head -c 1K </dev/urandom > docs/la
1.bin
ubuntu1@Ubuntu1:~/audit_env$ head -c 2K </dev/urandom > docs/la
2.bin
ubuntu1@Ubuntu1:~/audit_env$ touch -t 202409010101 logs/old.log
   # very old
ubuntu1@Ubuntu1:~/audit_env$ touch -t 202509250101 logs/recent.
   # ~72h ago
ubuntu1@Ubuntu1:~/audit_env$ touch -t 202509270101 logs/today.l
   # within 24h
```

```
ubuntu1@Ubuntu1:~/audit_env$ touch hidden/.secret hidden/.confi
ubuntu1@Ubuntu1:~/audit_env$ mkdir empty/dir
ubuntu1@Ubuntu1:~/audit_env$ touch tmp/file.tmp tmp/data~ tmp/n
s.bak
ubuntu1@Ubuntu1:~/audit_env$ touch docs/open.txt
ubuntu1@Ubuntu1:~/audit_env$ chmod 666 docs/open.txt
ubuntu1@Ubuntu1:~/audit_env$
```

__Using the Find command to find files that are larger than the average size

```
ubuntu1@Ubuntu1:~/audit_env$ avg=$(find . -type f -pri
 awk '{sum+=$1; n++} END{if(n>0) print int(sum/n)}')
ubuntu1@Ubuntu1:~/audit_env$ find . -type f -size +"${
./docs/large1.bin
./docs/large2.bin
```

__Files that were modified within last 72h but not last 24h

```
ubuntu1@Ubuntu1:~/audit_env$ find . -type f -mtime -3 !
./logs/today.log
ubuntu1@Ubuntu1:~/audit_env$
```

__Empty Directories or Containing only hidden files

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~/audit_env$ find . -type d -empty
find . -type d ! -empty -exec sh -c '
  for d; do
    files=$(ls -A "$d")
    [[ -n "$files" && "$files" == .* ]] && echo "$d"
  done
' sh {} +
./empty/dir
sh: 4: [[: not found
sh: 4: [[: not found
sh: 4: [[: not found
sh: 4: [[: not found
sh: 4: [[: not found
```

__World-writable files

```
ubuntu1@Ubuntu1:~$ find . -type f -perm -002
./Desktop/prac/games.txt
./doc.txt
./audit_env/docs/open.txt
ubuntu1@Ubuntu1:~$
```

__Owned by users rather than me or root

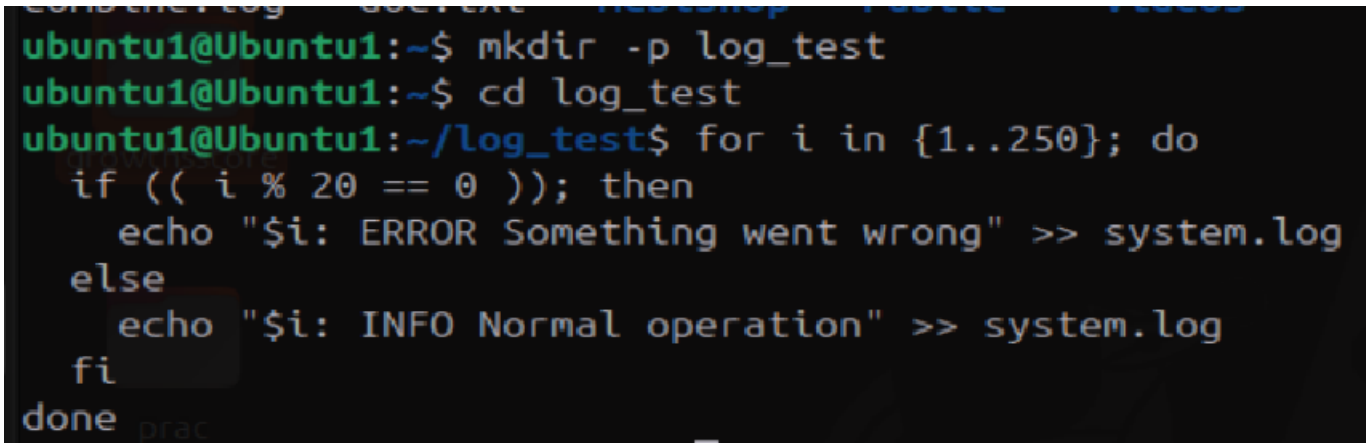
```
ubuntu1@Ubuntu1:~$ find . -type f ! -user $USER ! -user
./audit_env/docs/otheruser_testfile.txt
ubuntu1@Ubuntu1:~$
```

__Temporaly/backup files (*.tmp, *~, *.bak)

```
ubuntu1@Ubuntu1:~$ find . -type f \( -name '*.tmp' -o -n
o -name '*.bak' \)
./audit_env/tmp/file.tmp
./audit_env/tmp/notes.bak
./audit_env/tmp/data~
```

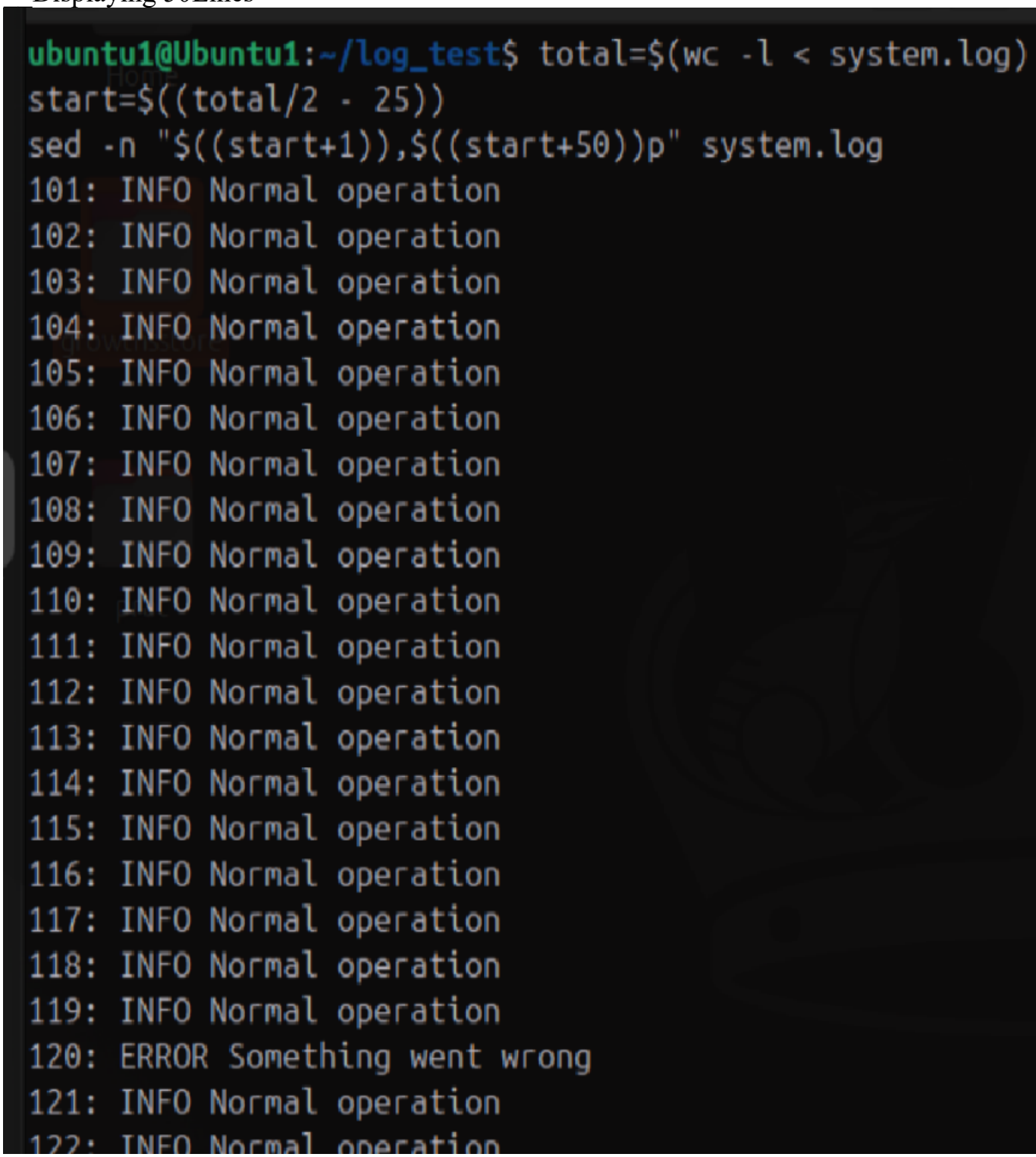
Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Qn9: Creating a large log file



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Displaying 50Lines



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
130: INFO Normal operation
131: INFO Normal operation
132: INFO Normal operation
133: INFO Normal operation
134: INFO Normal operation
135: INFO Normal operation
136: INFO Normal operation
137: INFO Normal operation
138: INFO Normal operation
139: INFO Normal operation
140: ERROR Something went wrong
141: INFO Normal operation
142: INFO Normal operation
143: INFO Normal operation
144: INFO Normal operation
```

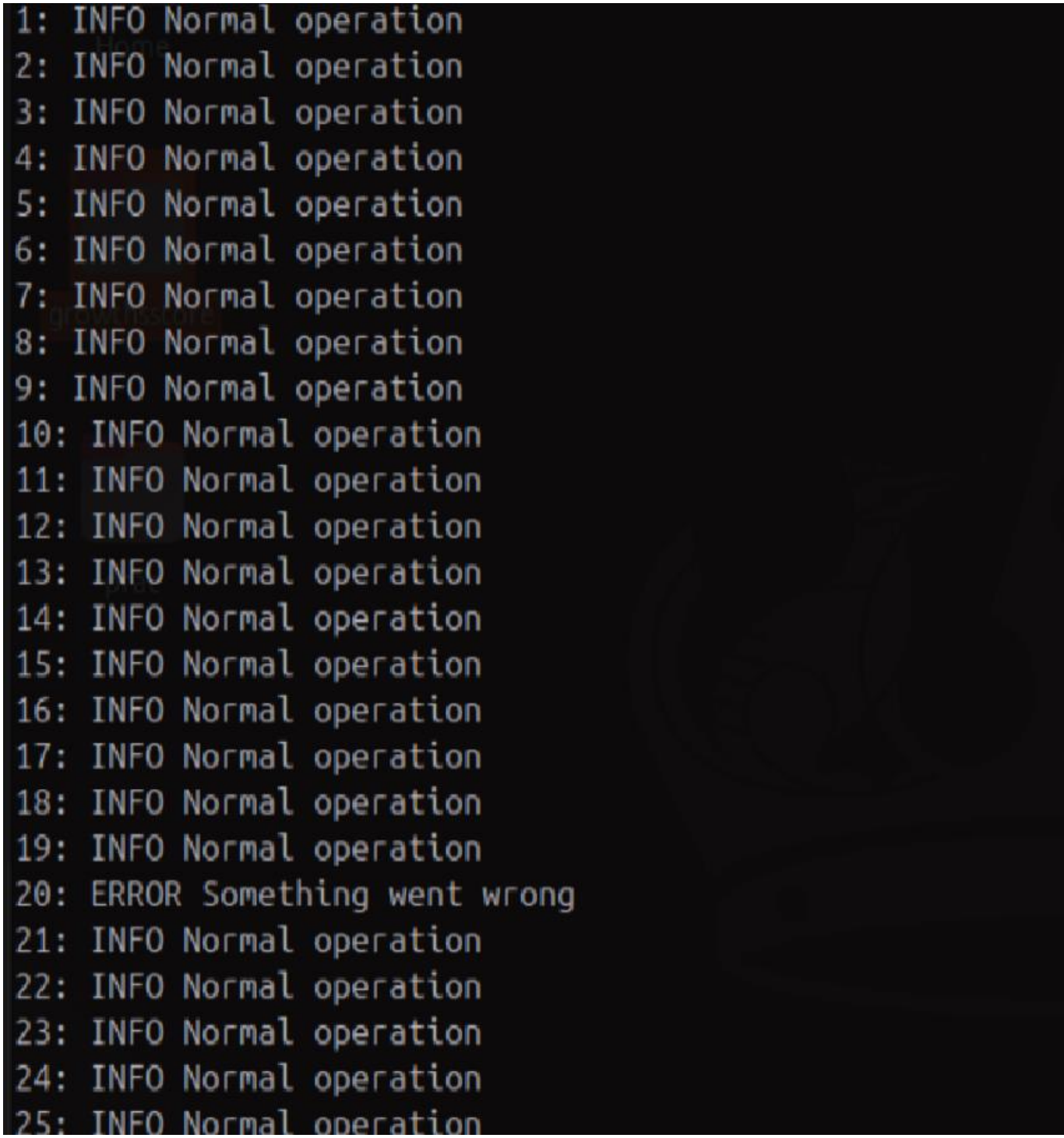
__Finding the **last occurrence** of a word (ERROR) with 5 lines of context

```
ubuntu1@Ubuntu1:~/log_test$ grep -n -B5 -A5 "ERROR" syst
ail -n 11
235-235: INFO Normal operation
236-236: INFO Normal operation
237-237: INFO Normal operation
238-238: INFO Normal operation
239-239: INFO Normal operation
240:240: ERROR Something went wrong
241-241: INFO Normal operation
242-242: INFO Normal operation
243-243: INFO Normal operation
244-244: INFO Normal operation
245-245: INFO Normal operation
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

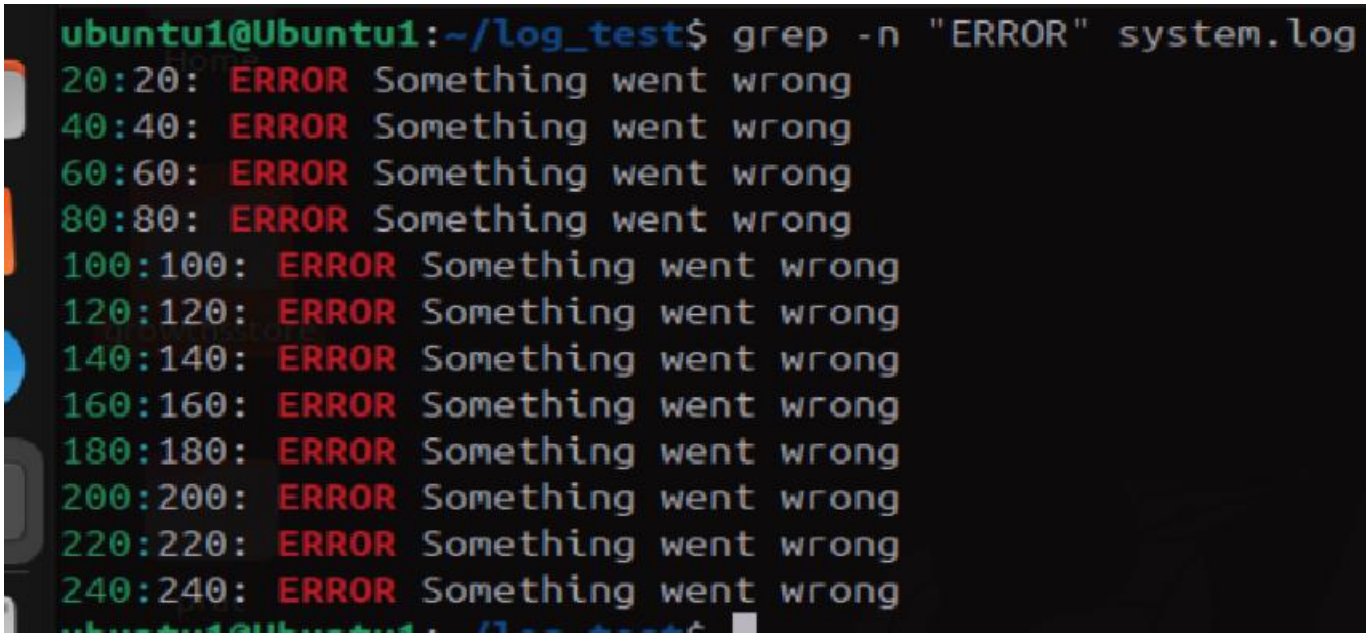
__Comparing the efficiency of viewing with different tools, “less” and “cat”

less is more efficient over SSH → it **fetches and displays pages**, unlike cat which sends the full file at once**. cat** streams the entire file immediately → can be slow on large files, consumes bandwidth**. less** displays **one screen at a time**, fetches data as needed → saves bandwidth, easier to scroll/search.

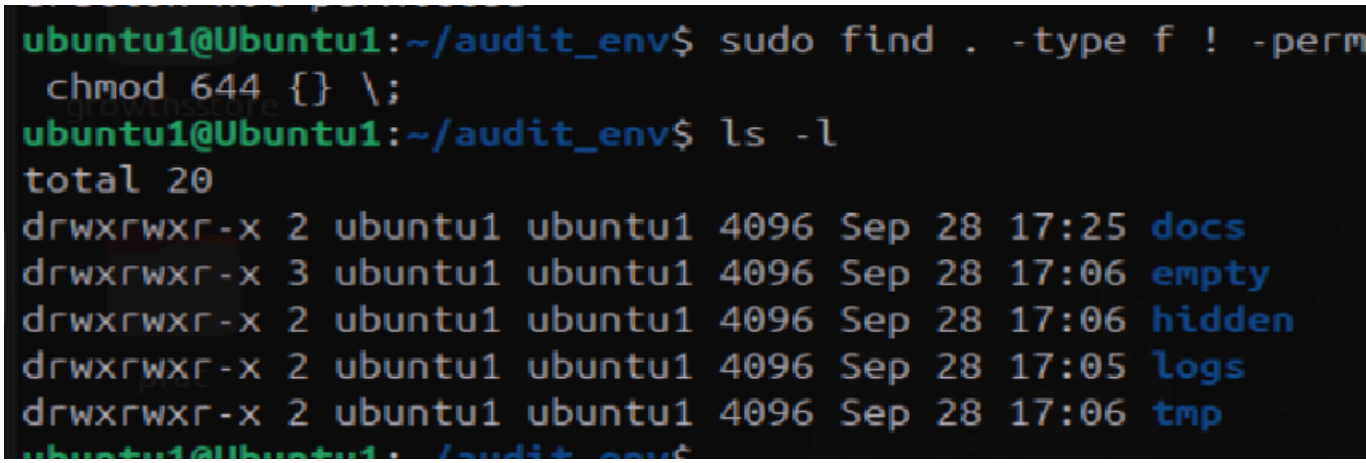


Name: Irumva Jason ID:27241 Date:Fri/26th/2025

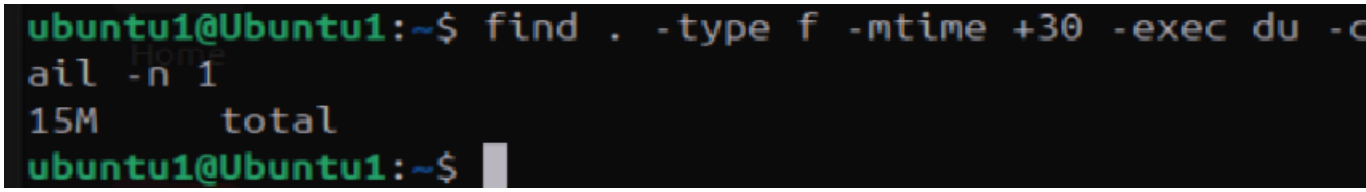
__Extracting **only lines with error patterns**, preserving line numbers



Qn10: Changing Permisdions

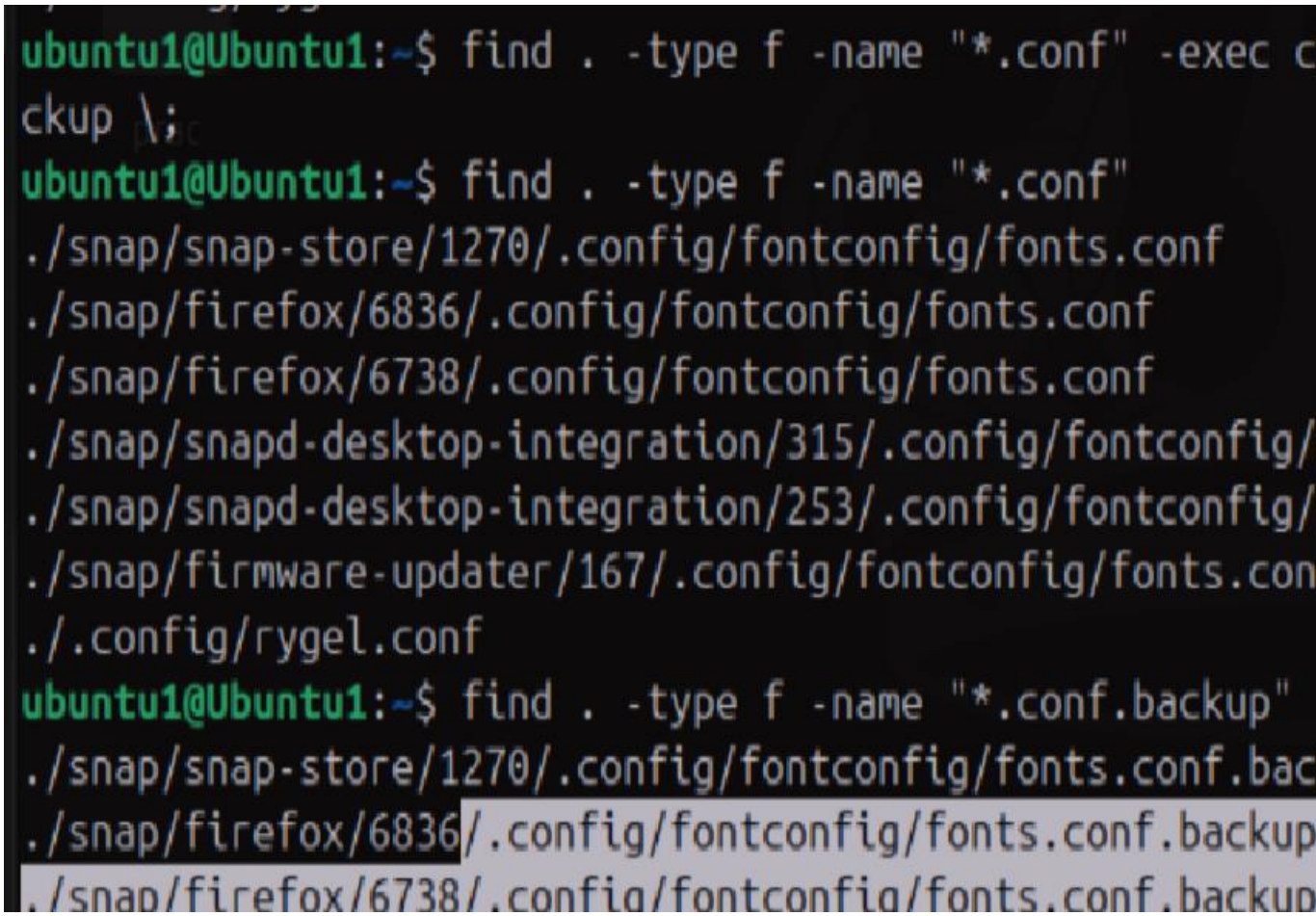


__Calculate total disk space used by files older than 30 days

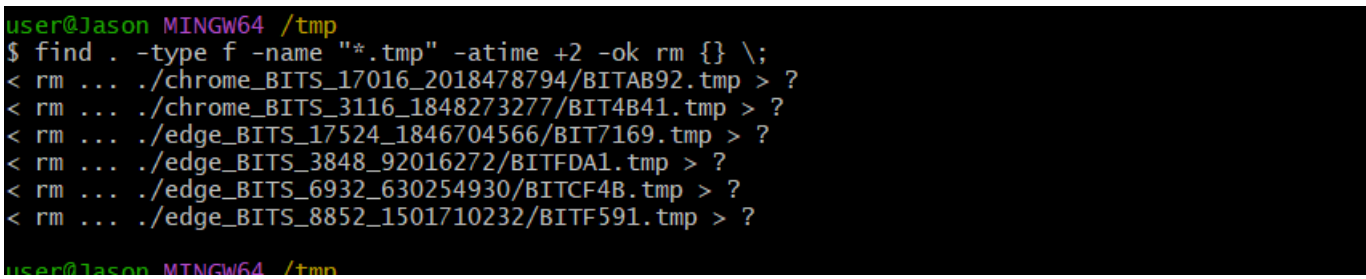


__Backingup all configuration files (*.conf)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



___Removing temporary files **safely** (not accessed recently)



__ Showing how to preview dangerous operations before executing them.

Always **preview with find first** before -exec.

Use -ok instead of -exec for destructive commands.

You can replace dangerous commands with echo first to verify:

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:/$ sudo find . -type f -name "*.tmp" -at
ec echo rm {} \;
find: ‘./run/user/1000/gvfs’: Permission denied
find: ‘./run/user/1000/doc’: Permission denied
```

Qn11: Creating test directories and files

```
ubuntu1@Ubuntu1:~$ mkdir -p ~/compression_test/{media,t
ubuntu1@Ubuntu1:~$ cp /usr/share/backgrounds/*.jpg ~/co
est/media/ 2>/dev/null
ubuntu1@Ubuntu1:~$ cp /usr/share/sounds/* ~/compression
/ 2>/dev/null
ubuntu1@Ubuntu1:~$ echo "This is a sample log file" > ~
n_test/text/log.txt
ubuntu1@Ubuntu1:~$ yes "Some repeated text line" | head
 ~/compression_test/text/biglog.txt
ubuntu1@Ubuntu1:~$ ls
audit_env          data.csv   Downloads  Music      Templ
combined.log       Desktop    log_test   Pictures   textd
combine.log        doc.txt    mcbishop   Public     Video
compression_test   Documents  media      snap
ubuntu1@Ubuntu1:~$ ls compression_test
media  text
```

__Creating archives with different algorithms

```
ubuntu1@Ubuntu1:~$ tar -cvf media_bz2.tar ~/compression
 && bzip2 media_bz2.tar
tar: Removing leading `/' from member names
/home/ubuntu1/compression_test/media/
/home/ubuntu1/compression_test/media/Province_of_the_so
ce_by_orbitelambda.jpg
/home/ubuntu1/compression_test/media/Clouds_by_Tibor_Mo
/home/ubuntu1/compression_test/media/Monument_valley_by
da.jpg
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

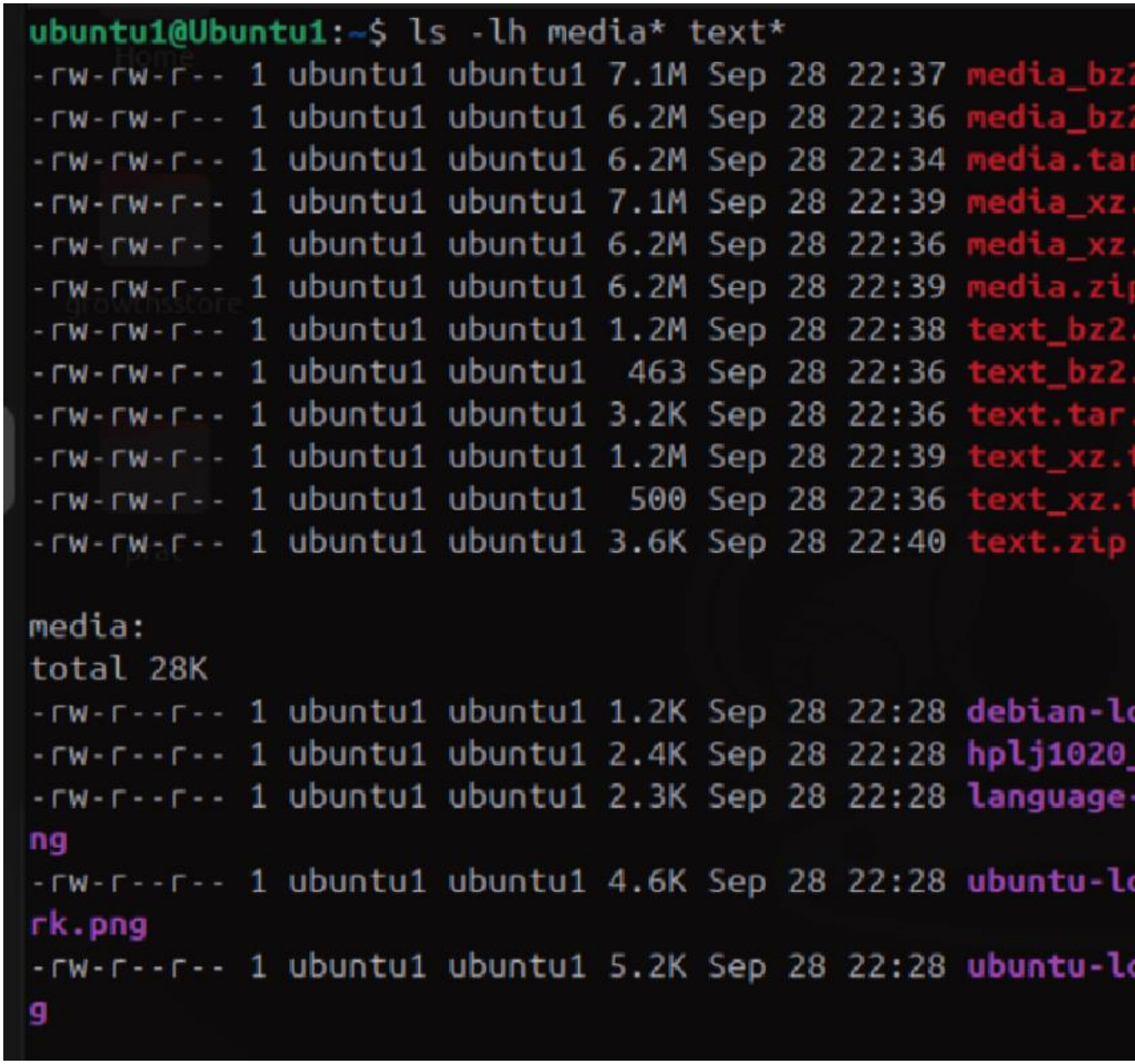
```
ubuntu1@Ubuntu1:~$ tar -cvf text_bz2.tar ~/compression_te
& bzip2 text_bz2.tar
tar: Removing leading `/' from member names
/home/ubuntu1/compression_test/text/
/home/ubuntu1/compression_test/text/log.txt
/home/ubuntu1/compression_test/text/biglog.txt
```

```
ubuntu1@Ubuntu1:~$ # xz
tar -cvf media_xz.tar ~/compression_test/media && xz med
tar: Removing leading `/' from member names
/home/ubuntu1/compression_test/media/
/home/ubuntu1/compression_test/media/Province_of_the_sou
ce_by_orbitelambda.jpg
/home/ubuntu1/compression_test/media/Clouds_by_Tibor_Mok
/home/ubuntu1/compression_test/media/Monument_valley_by_
```

```
ubuntu1@Ubuntu1:~$ zip -r media.zip ~/compression_test/m
updating: home/ubuntu1/compression_test/media/ (stored 0
updating: home/ubuntu1/compression_test/media/Province_o
h_of_france_by_orbitelambda.jpg (deflated 53%)
updating: home/ubuntu1/compression_test/media/Clouds_by_
nszki.jpg (deflated 2%)
updating: home/ubuntu1/compression_test/media/Monument_v
ubuntu1@Ubuntu1:~$ ls
audit_env          media_bz2.tar       test_text.tar.gz
combined.log       media_bz2.tar.bz2   test_text.tar.xz
combine.log        media.tar.gz        test_text.zip
compression_test   media_xz.tar        text_bz2.tar
data.csv           media_xz.tar.xz     text_bz2.tar.bz2
Desktop            media.zip           textdata
doc.txt            Music               text.tar.gz
Documents          Pictures            text_xz.tar
Downloads          Public              text_xz.tar.xz
log_test           snap                text.zip
```

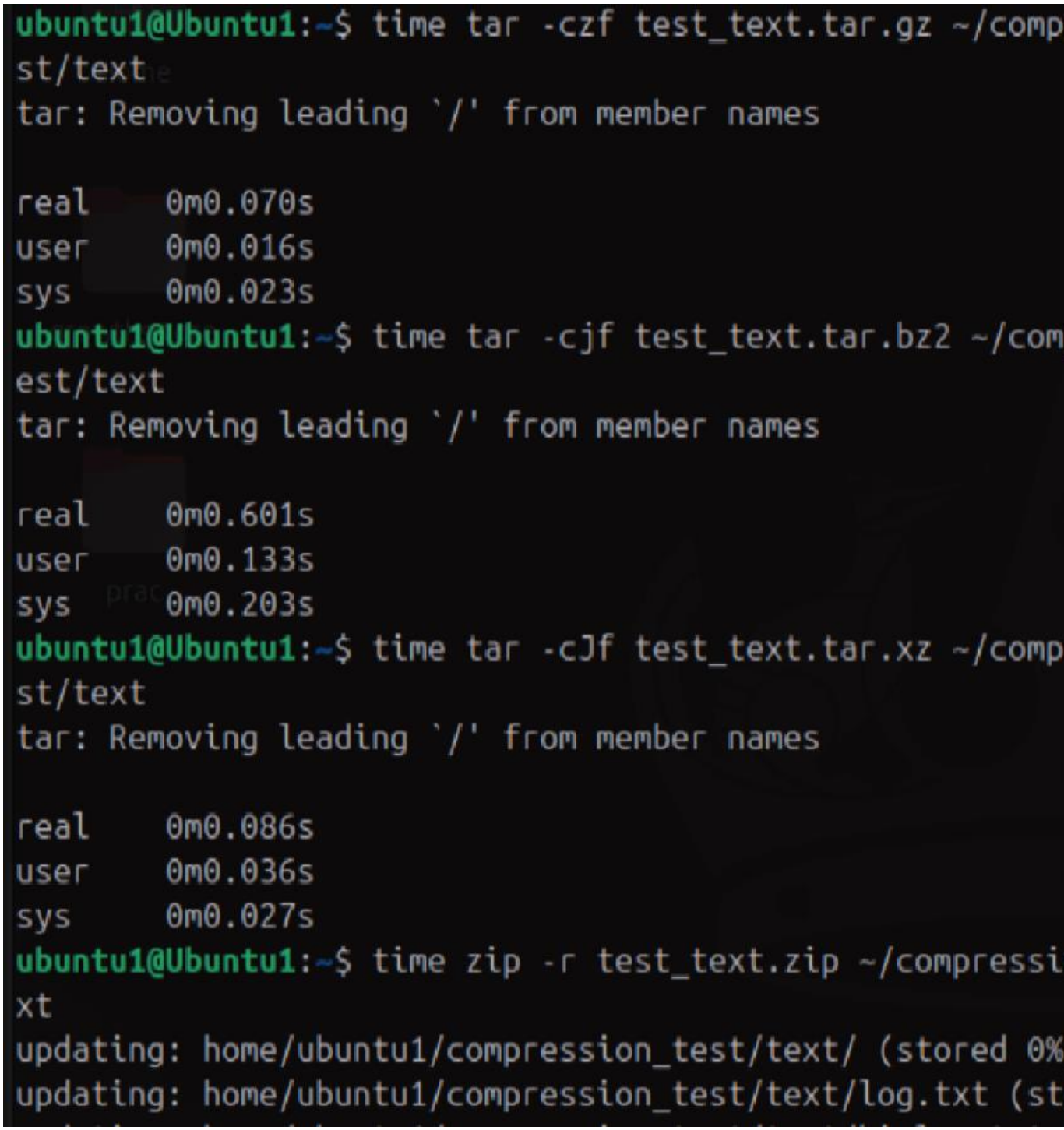
Name: Irumva Jason ID:27241 Date:Fri/26th/2025

__Measuring sizes



___Measuring speed (compression time)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Expected results (analysis)

- **Media (.jpg, .mp4, .zip):** Already compressed → all methods show little/no size reduction, only wasted CPU time. Best to just archive (tar without compression).
- **Text files (logs, configs):**

  gzip: **Fast**, good compression.

bzip2: **Better compression** but slower.

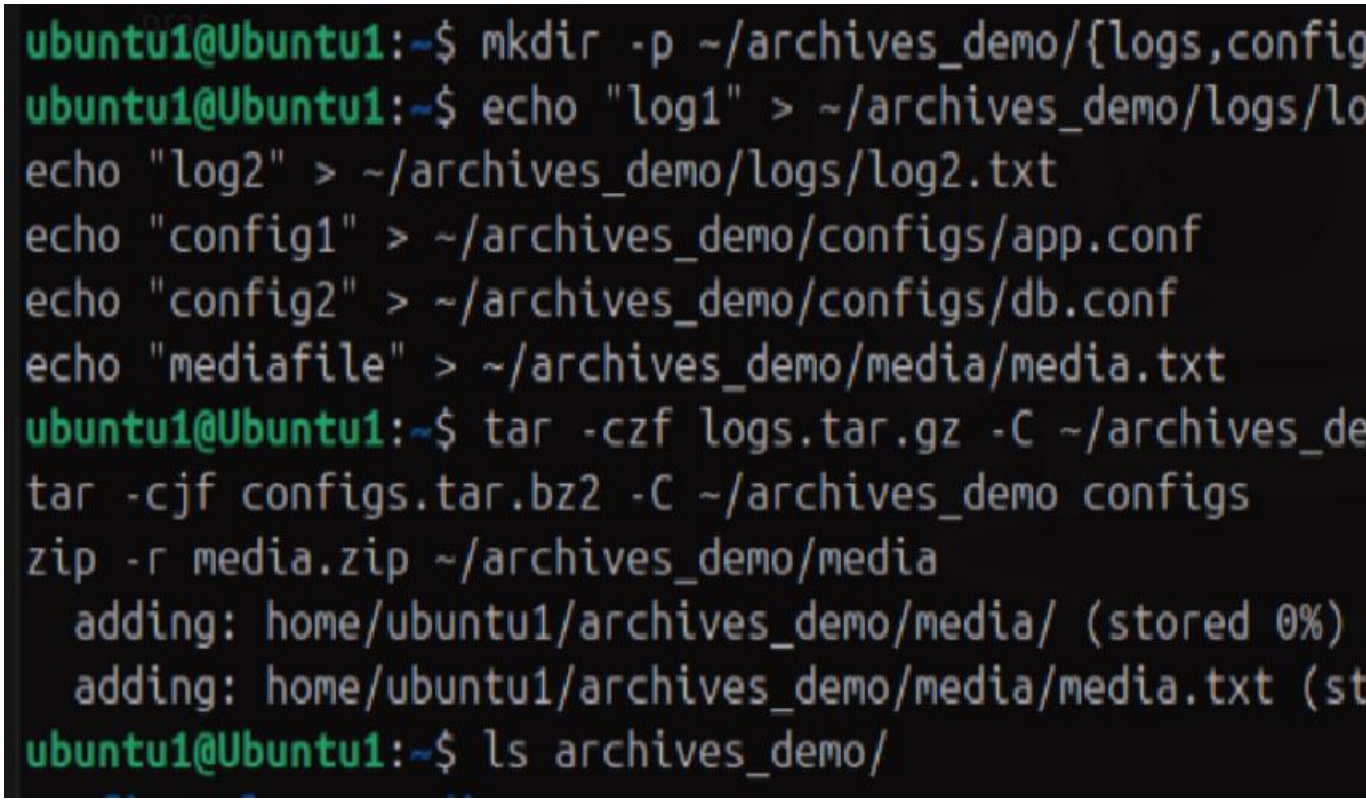
xz: **Best compression ratio** but **slowest**.

zip: Cross-platform, moderate compression.

Recommendation for automated backups

- Use **tar + gzip (.tar.gz)**: good balance of speed + compression, widely supported.
- For **long-term archival storage** (rare restores, max space-saving) → tar + xz.
- For **media files** → just tar (no compression) to save time/CPU.

Qn12: Creating sample archives



__Safely examining archive contents (no extraction)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ tar -tzf logs.tar.gz
logs/
logs/log1.txt
logs/log2.txt
ubuntu1@Ubuntu1:~$ tar -tjf configs.tar.bz2
configs/
configs/db.conf
configs/app.conf
ubuntu1@Ubuntu1:~$ unzip -l media.zip
Archive:  media.zip
  Length      Date    Time    Name
---------  ---------- -----   ----
        0  2025-09-28 22:32   home/ubuntu1/compression_test/media/
  1408323  2025-09-28 22:32   home/ubuntu1/compression_test/media/
Province_of_the_south_of_france_by_orbitelambda.jpg
  4032506  2025-09-28 22:32   home/ubuntu1/compression_test/media/
Clouds_by_Tibor_Mokanszki.jpg
  1957897  2025-09-28 22:32   home/ubuntu1/compression_test/media/
Monument_valley_by_orbitelambda.jpg
        0  2025-09-28 23:15   home/ubuntu1/archives_demo/media/
       10  2025-09-28 23:15   home/ubuntu1/archives_demo/media/med
ia.txt
---------                     -------
  7398736                     6 files
```

__Extracting only matching files

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ tar -xjf configs.tar.bz2 --wildcards
ubuntu1@Ubuntu1:~$ ls configs/*.conf
configs/app.conf  configs/db.conf
ubuntu1@Ubuntu1:~$ unzip media.zip '*/media.txt'
Archive:  media.zip
 extracting: home/ubuntu1/archives_demo/media/media.txt
ubuntu1@Ubuntu1:~$ cat extracted/*/media.txt
cat: 'extracted/*/media.txt': No such file or directory
ubuntu1@Ubuntu1:~$ cat archives_demo/media/media.txt
mediafile
ubuntu1@Ubuntu1:~$ tar -xzf logs.tar.gz logs/log1.txt
ubuntu1@Ubuntu1:~$ cat logs/log1.txt
log1
```

__Updating existing archives without full recreation

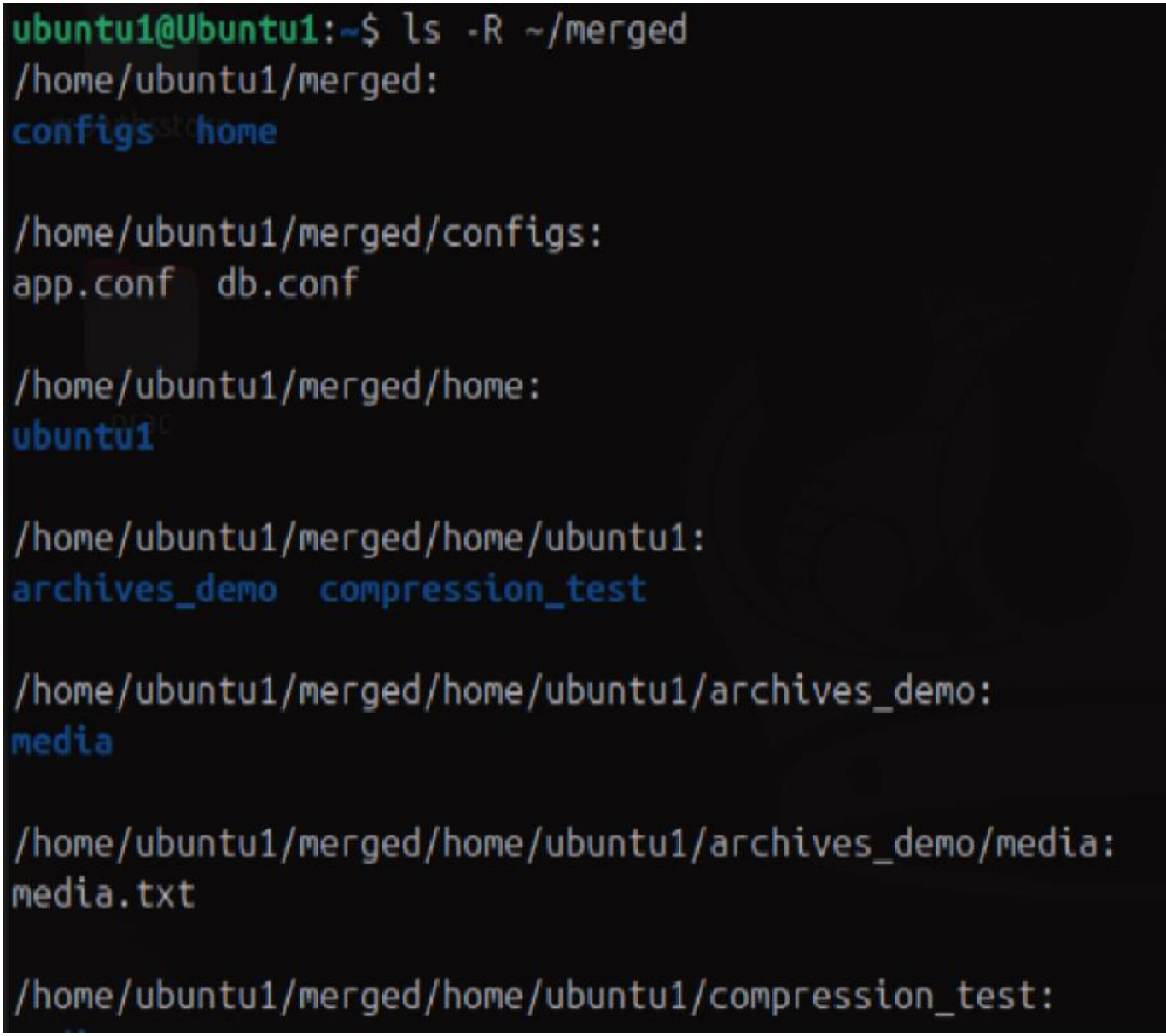
```
ubuntu1@Ubuntu1:~$ tar -rf logs.tar logs/log3.txt    # a
gzip logs.tar
tar: logs/log3.txt: Cannot stat: No such file or directo
tar: Exiting with failure status due to previous errors
gzip: logs.tar.gz already exists; do you wish to overwri
)? y
ubuntu1@Ubuntu1:~$ tar -tzf logs.tar.gz
```

___Handling corrupted archives

```
ubuntu1@Ubuntu1:~$ cp logs.tar.gz broken.tar.gz
ubuntu1@Ubuntu1:~$ truncate -s -10 broken.tar.gz   # cho
 bytes
ubuntu1@Ubuntu1:~$ tar -tzf broken.tar.gz

gzip: stdin: unexpected end of file
tar: Child returned status 1
tar: Error is not recoverable: exiting now
ubuntu1@Ubuntu1:~$ gzip -tv broken.tar.gz
broken.tar.gz:
gzip: broken.tar.gz: unexpected end of file
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

__Merging contents into a single new archive



Qn13:Setting up Backup directories and sample data

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ mkdir -p ~/prod_data
ubuntu1@Ubuntu1:~$ echo "file1" > ~/prod_data/app.log
ubuntu1@Ubuntu1:~$ echo "file2" > ~/prod_data/config.cfg
ubuntu1@Ubuntu1:~$ ls -l ~/prod_data
total 8
-rw-rw-r-- 1 ubuntu1 ubuntu1 6 Sep 29 14:12 app.log
-rw-rw-r-- 1 ubuntu1 ubuntu1 6 Sep 29 14:12 config.cfg
```

```
ubuntu1@Ubuntu1:~$ mkdir -p ~/backups/{daily,weekly,mont
ubuntu1@Ubuntu1:~$ ls -R ~/backups
/home/ubuntu1/backups:
daily  monthly  weekly

/home/ubuntu1/backups/daily:

/home/ubuntu1/backups/monthly:

/home/ubuntu1/backups/weekly:
```

```
ubuntu1@Ubuntu1:~$ mkdir -p ~/data
ubuntu1@Ubuntu1:~$ echo "file1" > ~/data/file1.txt
ubuntu1@Ubuntu1:~$ echo "file2" > ~/data/file2.txt
ubuntu1@Ubuntu1:~$ ls -l ~/data
total 8
-rw-rw-r-- 1 ubuntu1 ubuntu1 6 Sep 29 14:32 file1.txt
-rw-rw-r-- 1 ubuntu1 ubuntu1 6 Sep 29 14:33 file2.txt
```

____Daily incremental backup (using --listed-incremental)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

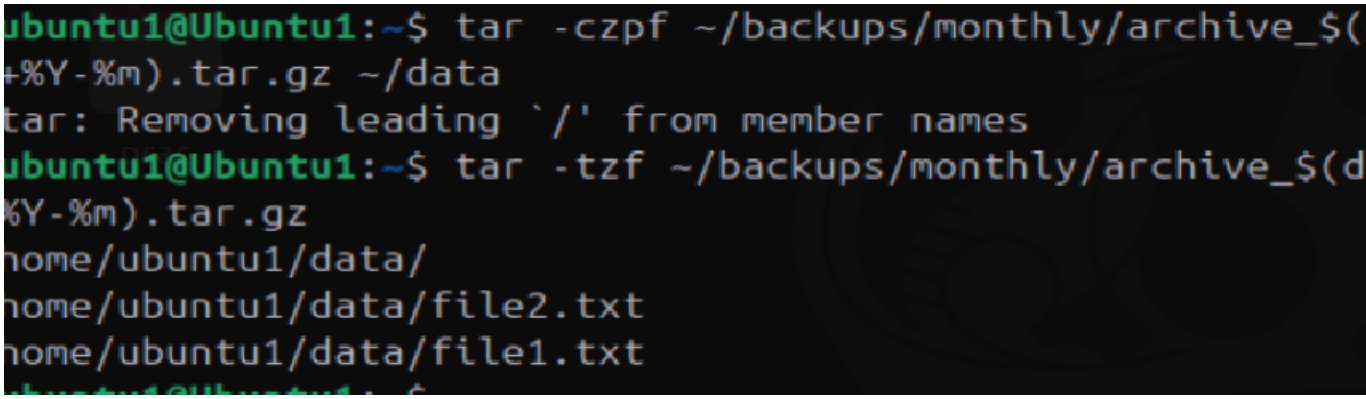
```
ubuntu1@Ubuntu1:~$ # 1. Make sure full structure exists
mkdir -p ~/backups/daily

# 2. Create the snapshot file
touch ~/backups/daily.snar

# 3. Run incremental backup again
tar --listed-incremental=~/backups/daily.snar -czpf ~/ba
daily/backup_$(date +%F).tar.gz ~/data

# 4. Verify contents
tar -tzf ~/backups/daily/backup_$(date +%F).tar.gz
tar: ~/backups/daily.snar: Cannot open: No such file or
ory
tar: Removing leading `/' from member names
tar: Exiting with failure status due to previous errors
home/ubuntu1/data/
home/ubuntu1/data/file1.txt
home/ubuntu1/data/file2.txt
```

__Weekly full backup

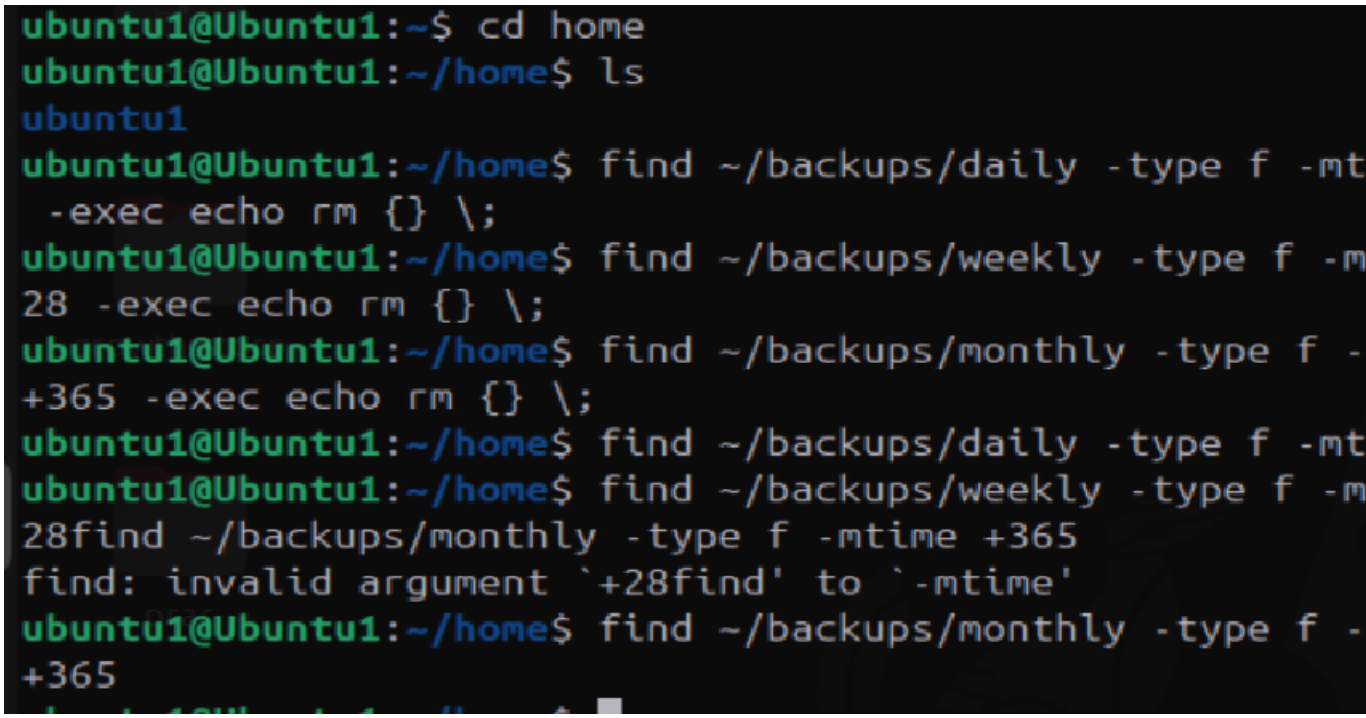
```
ubuntu1@Ubuntu1:~$ tar --listed-incremental=/dev/null -c
backups/weekly/full_$(date +%F).tar.gz ~/data
tar: Removing leading `/' from member names
tar: /dev/null: Cannot truncate: Invalid argument
tar: Exiting with failure status due to previous errors
ubuntu1@Ubuntu1:~$ tar -tzf ~/backups/weekly/full_$(date
tar.gz
home/ubuntu1/data/
home/ubuntu1/data/file1.txt
home/ubuntu1/data/file2.txt
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

__MonthlyArchive



___Automatic cleanup (keep last 7 daily, 4 weekly, 12 monthly)



__Verifying backup integrity

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ tar -tzf ~/backups/daily/backup_$(date
.tar.gz > /dev/null && echo "OK"
OK
ubuntu1@Ubuntu1:~$ tar -tzf ~/backups/daily/backup_$(date
.tar.gz > /dev/null && echo "OK"
OK
```

Qn14: Checking my current user context(ubuntu1)

```
ubuntu1@Ubuntu1:~$ whoami
ubuntu1
ubuntu1@Ubuntu1:~$ id
uid=1000(ubuntu1) gid=1000(ubuntu1) groups=1000(ubuntu1)
dm),24(cdrom),27(sudo),30(dip),46(plugdev),100(users),11
admin)
ubuntu1@Ubuntu1:~$ groups
ubuntu1 adm cdrom sudo dip plugdev users lpadmin
ubuntu1@Ubuntu1:~$
```

__Creating a test user to compare groups:

```
ubuntu1@Ubuntu1:~$ sudo useradd -m testuser
[sudo] password for ubuntu1:
ubuntu1@Ubuntu1:~$ id testuser
uid=1002(testuser) gid=1002(testuser) groups=1002(testuse
ubuntu1@Ubuntu1:~$ groups testuser
testuser : testuser
```

Observation: system accounts usually have limited groups, regular users often in sudo, users, etc.

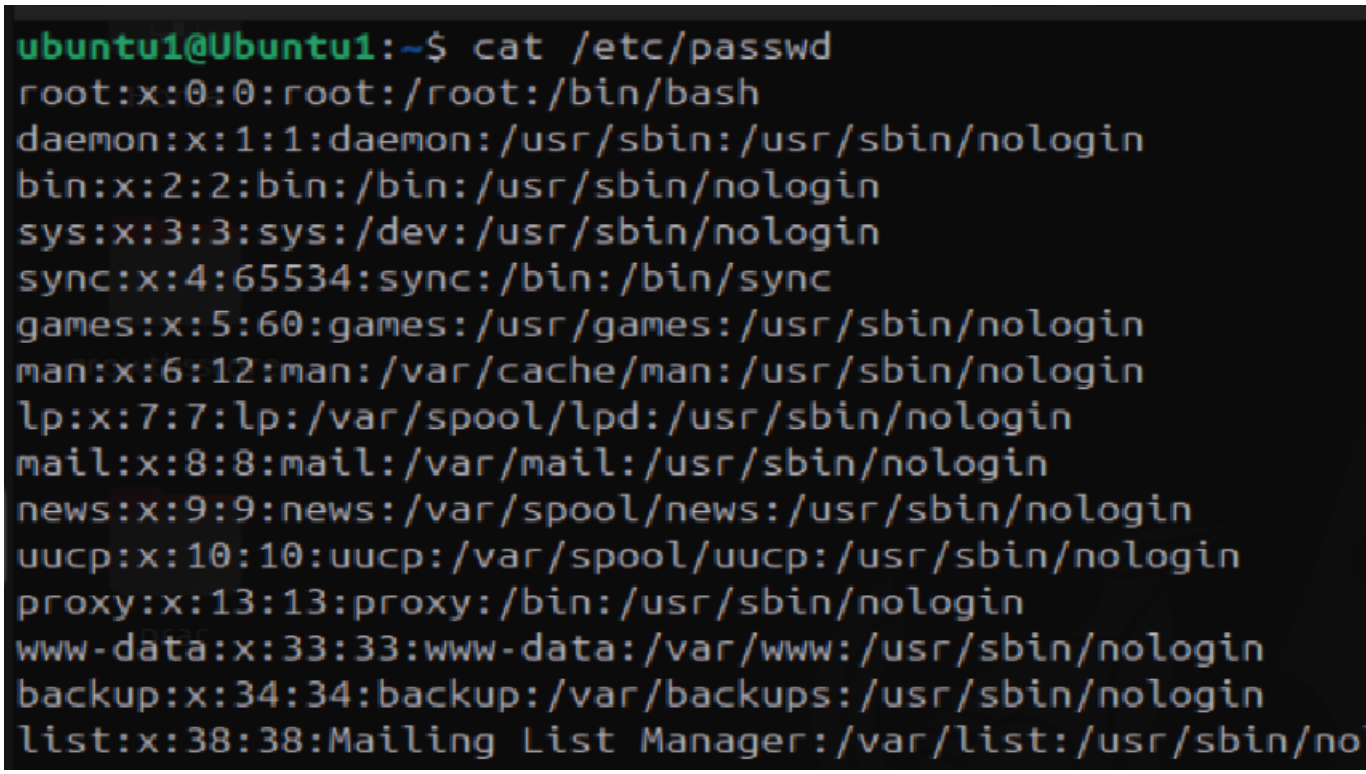
```
ubuntu1@Ubuntu1:~$ groups
ubuntu1 adm cdrom sudo dip plugdev users lpadmin
ubuntu1@Ubuntu1:~$ groups testuser
testuser : testuser
```

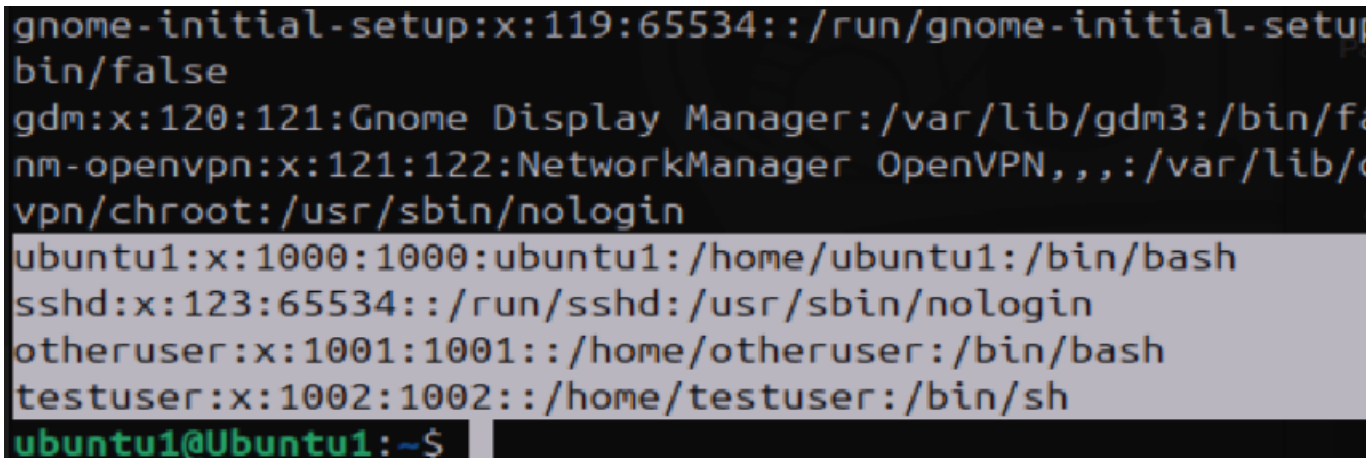
Name: Irumva Jason ID:27241 Date:Fri/26th/2025

__Examining /etc/passwd entries

System users: UID < 1000, login shell usually /usr/sbin/nologin or /bin/false, no home directory for normal login.

Regular users: UID ≥ 1000, valid shell (e.g., /bin/bash), home directory /home/username.





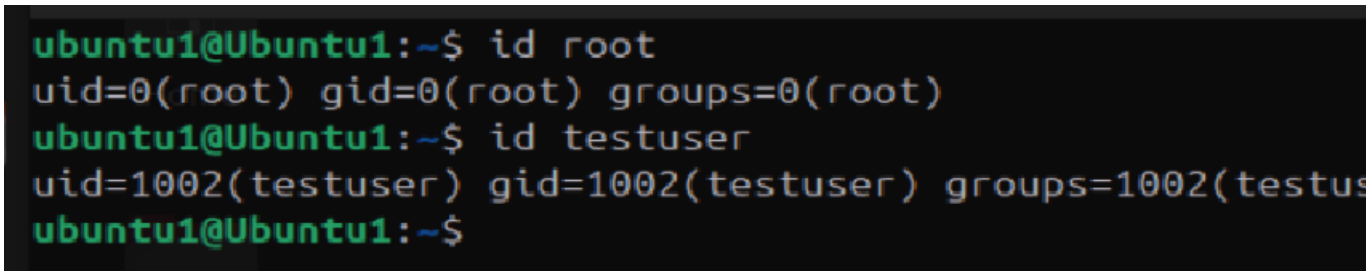
- ___Comparing groups with a system user
  Risk: If a regular user is added to groups like root or adm, they could:

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

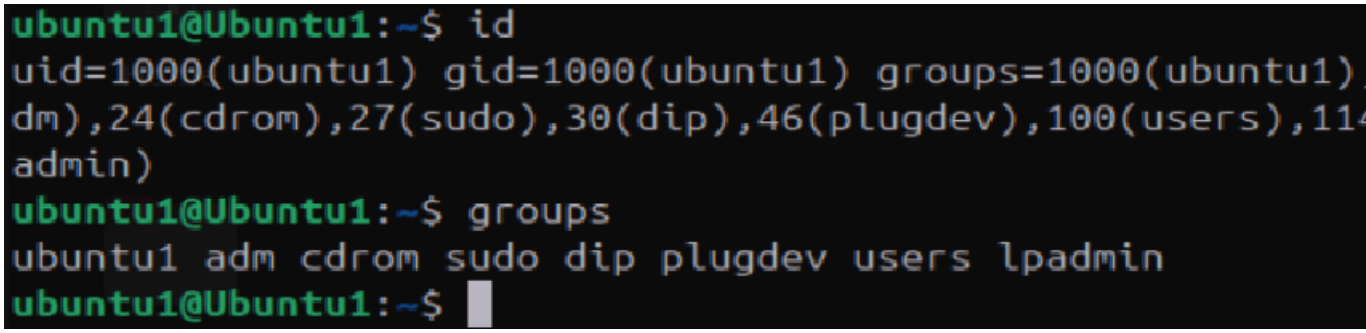
Read/write sensitive files

Change system configurations

Escalate privileges



Qn15: Checking current effective vs configured groups



__Demonstrating re-login requirement

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ sudo useradd -m colleague
ubuntu1@Ubuntu1:~$ sudo passwd ubuntu
passwd: user 'ubuntu' does not exist
ubuntu1@Ubuntu1:~$ sudo passwd colleague
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: password updated successfully
ubuntu1@Ubuntu1:~$ sudo usermod -aG adm colleague
ubuntu1@Ubuntu1:~$ id colleague
uid=1003(colleague) gid=1003(colleague) groups=1003(coll
e),4(adm)
ubuntu1@Ubuntu1:~$ groups colleague
colleague : colleague adm
ubuntu1@Ubuntu1:~$ groups
ubuntu1 adm cdrom sudo dip plugdev users lpadmin
```

After relog-in:

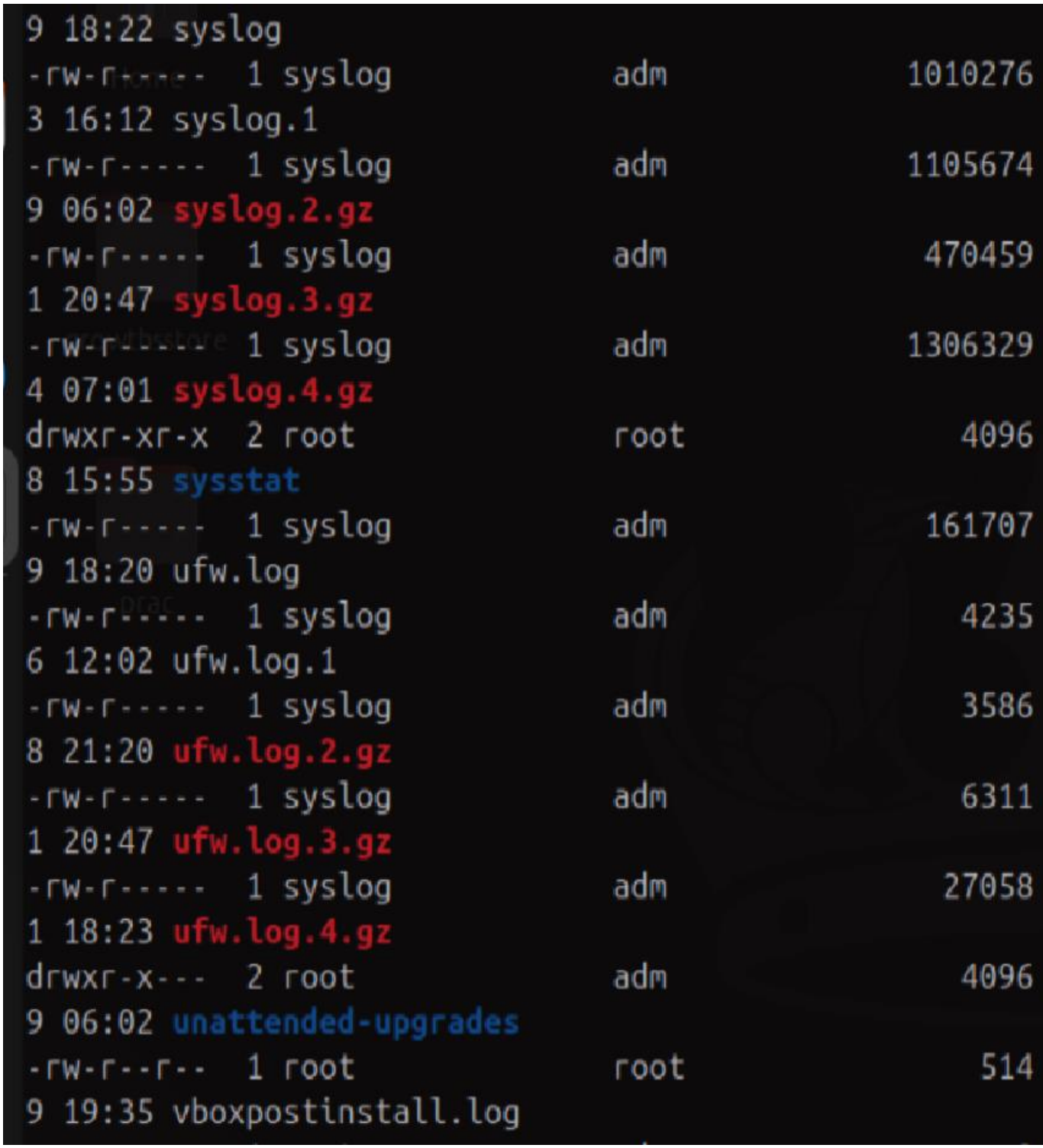
```
ubuntu1@Ubuntu1:~$ id colleague
uid=1003(colleague) gid=1003(colleague) groups=1003(coll
),4(adm)
ubuntu1@Ubuntu1:~$ groups colleague
colleague : colleague adm
ubuntu1@Ubuntu1:~$
```

__Identifying groups for common resources(Also checking group ownership and web files)

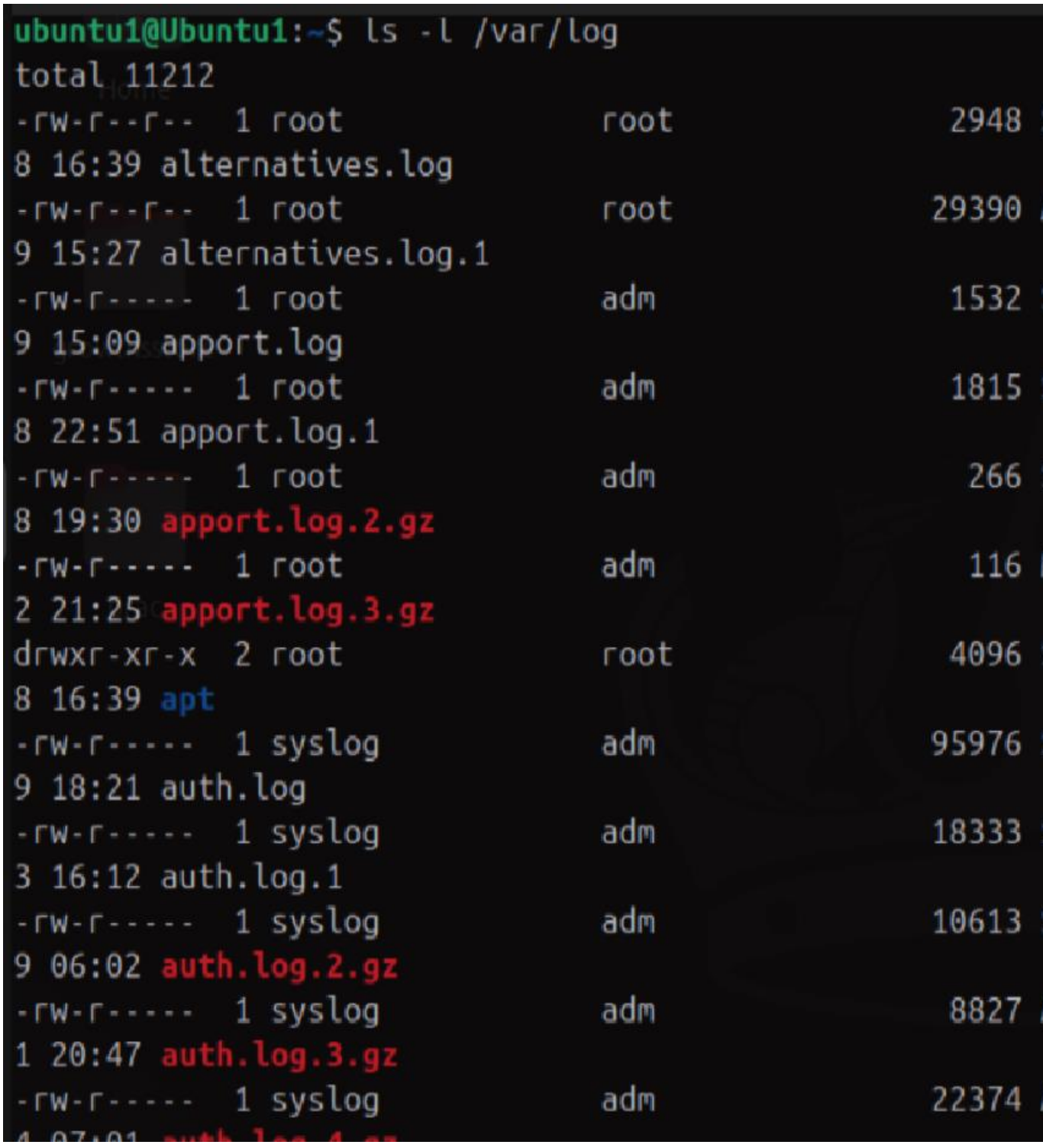
Resource	Typical Group
System logs (`/var/log`)	`adm`, `syslog`
Web server files	`www-data`
Administrative functions	`sudo`, `wheel`

Group ownership:

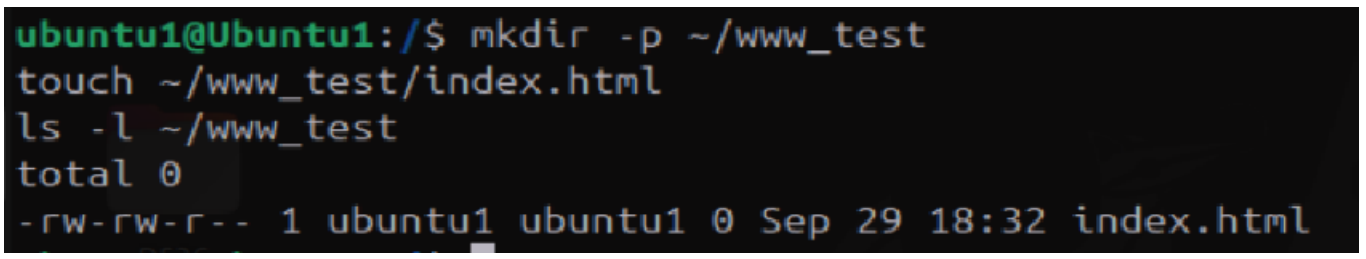
Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Web files:



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

Principle of Least Privilege

Assign users **only to the groups they need** for their job.

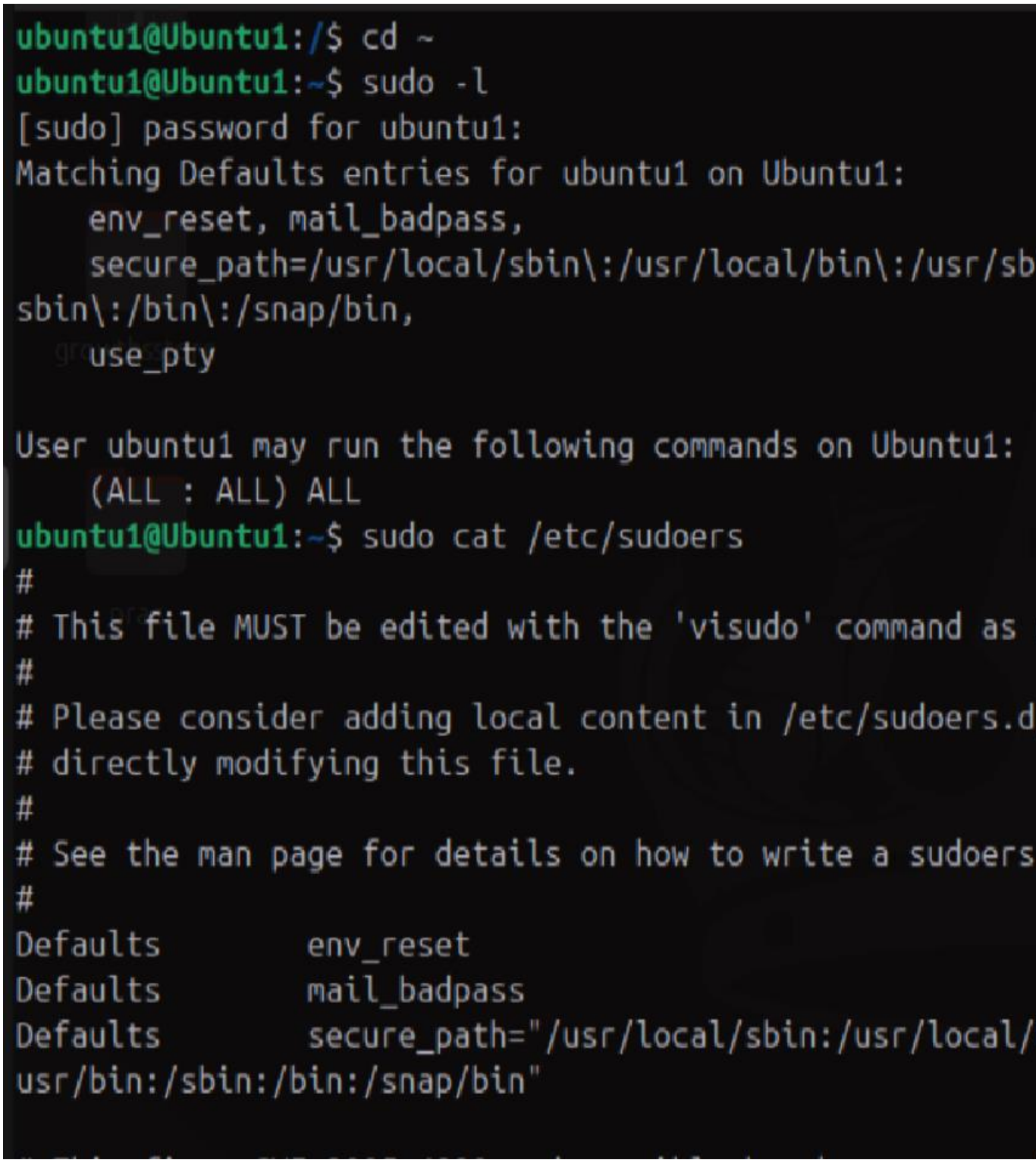
Avoid giving access to root, adm, or sudo unless necessary.

Helps **limit potential damage** from mistakes or compromised accounts.

Qn16: **Documenting my sudo permissions & restrictions**

Showing what my user is allowed to do and inspect sudoers safely.

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
# While you shouldn't normally run git as root, you need
per
#Defaults:%sudo env_keep += "GIT_AUTHOR_* GIT_COMMITTER_*

# Per-user preferences; root won't have sensible values f
#Defaults:%sudo env_keep += "EMAIL DEBEMAIL DEBFULLNAME"

# "sudo scp" or "sudo rsync" should be able to use your S
#Defaults:%sudo env_keep += "SSH_AGENT_PID SSH_AUTH_SOCK"

# Ditto for GPG agent
#Defaults:%sudo env_keep += "GPG_AGENT_INFO"

# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL:ALL) ALL

# Members of the admin group may gain root privileges
%admin ALL=(ALL) ALL
```

```
ubuntu1@Ubuntu1:~$ sudo ls -l /etc/sudoers.d
total 4
-r--r----- 1 root root 1068 Jan 29  2024 README
ubuntu1@Ubuntu1:~$ sudo sed -n '1,200p' /etc/sudoers.d/*
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ sudo sed -n '1,200p' /etc/sudoers.d/*
 true
#
# The default /etc/sudoers file created on installation
# sudo  package now includes the directive:
#
#       @includedir /etc/sudoers.d
#
# This will cause sudo to read and parse any files in the
d
# directory that do not end in '~' or contain a '.' chara
# exists. It is not an error if the directory does not ex
#
# Note also, that because sudoers contents can vary widel
is
# made to add this directive to existing sudoers files or
l free
# to add the above directive to the end of your /etc/sudo
able
# this functionality for existing installations if you wi
# Sudo versions older than 1.9.1 will only support the ol
# #includedir. That means that the sudo versions in Debia
)
# and later will happily accept both @includedir and #inc
#
```

```
ubuntu1@Ubuntu1:~$ sudo visudo -c
/etc/sudoers: parsed OK
/etc/sudoers.d/README: parsed OK
ubuntu1@Ubuntu1:~$
```

___ Demonstrating difference: sudo -i vs sudo su vs su

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

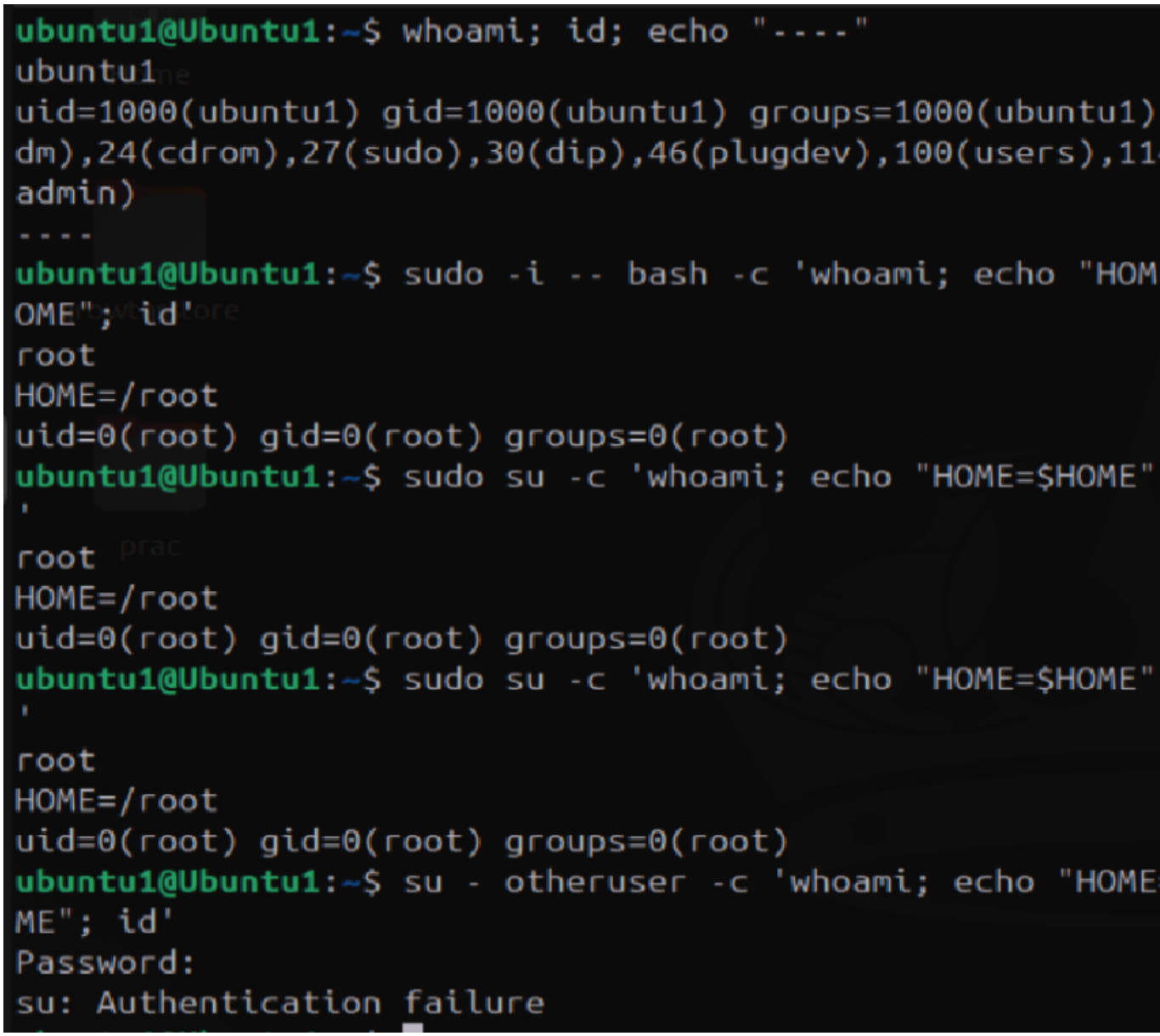
Each command below prints who you are, your HOME, and effective groups so you can inspect differences.

sudo -i : start a login shell as root (reads root's shell startup files)

sudo su : run su as root; typically gives a root shell (behavior depends on su flags)

su : switch user (requires target user's password unless run as root)

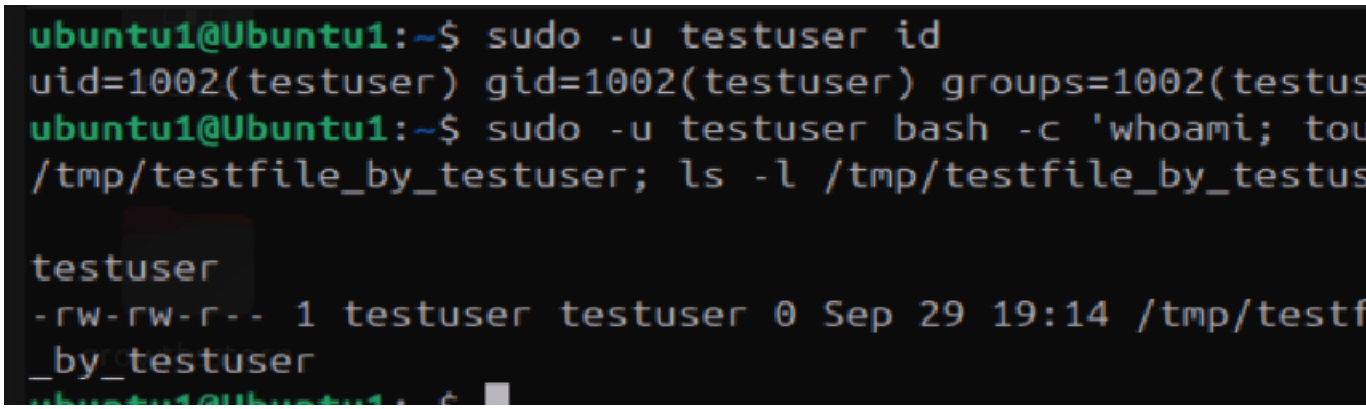
Example: run a command as another user "otheruser" (may prompt for password)



__Run commands as specific users (not root)

Examples to execute a command as testuser (created recently)
Run a single command as testuser (using sudo)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



__Analyzing login / sudo patterns using logs

Commands to extract recent relevant entries

For systemd systems: SSH/session journal (last 7 days)

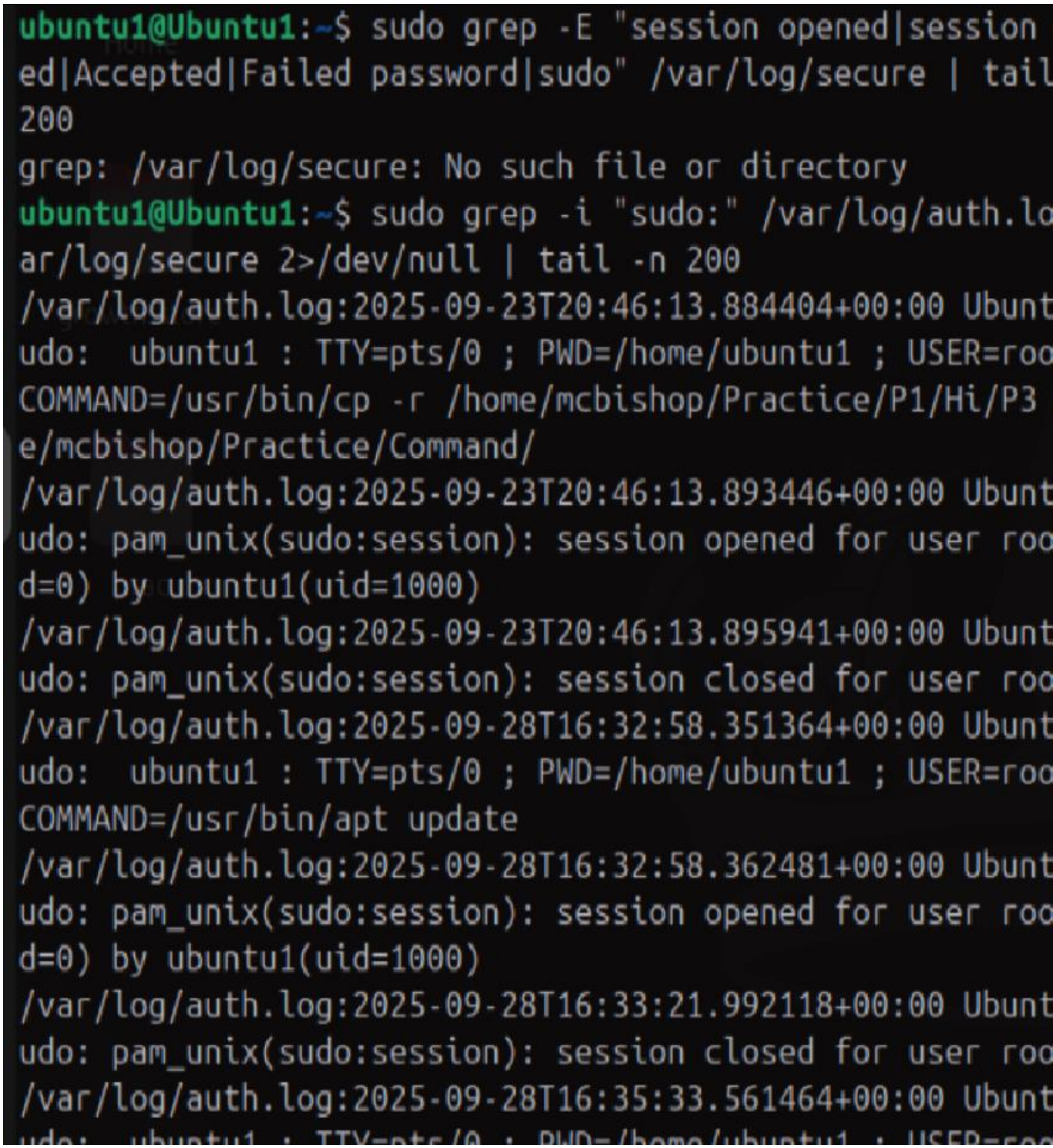
Show only sudo usage entries

List recent successful interactive logins

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ sudo journalctl --since "7 days ago"
TEMD_UNIT=sshd.service | tail -n 200
-- No entries --
ubuntu1@Ubuntu1:~$ sudo grep -E "session opened|session
ed|Accepted|Failed password|sudo" /var/log/auth.log | ta
n 200
2025-09-28T21:15:01.214382+00:00 Ubuntu1 CRON[6836]: pam
x(cron:session): session closed for user root
2025-09-28T21:17:01.234813+00:00 Ubuntu1 CRON[6848]: pam
x(cron:session): session opened for user root(uid=0) by
(uid=0)
2025-09-28T21:17:01.360929+00:00 Ubuntu1 CRON[6848]: pam
x(cron:session): session closed for user root
2025-09-28T21:25:01.422947+00:00 Ubuntu1 CRON[6899]: pam
x(cron:session): session opened for user root(uid=0) by
(uid=0)
2025-09-28T21:25:01.469652+00:00 Ubuntu1 CRON[6899]: pam
x(cron:session): session closed for user root
2025-09-28T21:30:01.581945+00:00 Ubuntu1 CRON[6940]: pam
x(cron:session): session opened for user root(uid=0) by
(uid=0)
2025-09-28T21:30:01.596258+00:00 Ubuntu1 CRON[6940]: pam
x(cron:session): session closed for user root
2025-09-28T21:35:01.678221+00:00 Ubuntu1 CRON[6968]: pam
x(cron:session): session opened for user root(uid=0) by
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ sudo last -a | head -n 20
ubuntu1  tty2         Mon Sep 29 18:19   still logged in
tty2
ubuntu1  seat0        Mon Sep 29 18:19   still logged in
login screen
reboot   system boot  Mon Sep 29 18:16   still running
6.11.0-24-generic
colleagu tty3         Mon Sep 29 16:56 - crash  (01:19)
tty3
colleagu seat0        Mon Sep 29 16:56 - crash  (01:19)
login screen
ubuntu1  tty2         Mon Sep 29 16:54 - crash  (01:21)
tty2
ubuntu1  seat0        Mon Sep 29 16:54 - 16:56  (00:02)
login screen
reboot   system boot  Mon Sep 29 16:54   still running
6.11.0-24-generic
ubuntu1  tty2         Mon Sep 29 15:56 - crash  (00:58)
tty2
ubuntu1  seat0        Mon Sep 29 15:56 - crash  (00:58)
login screen
ubuntu1  tty2         Mon Sep 29 14:09 - 15:56  (01:46)
tty2
ubuntu1  seat0        Mon Sep 29 14:09 - 15:56  (01:46)
login screen
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

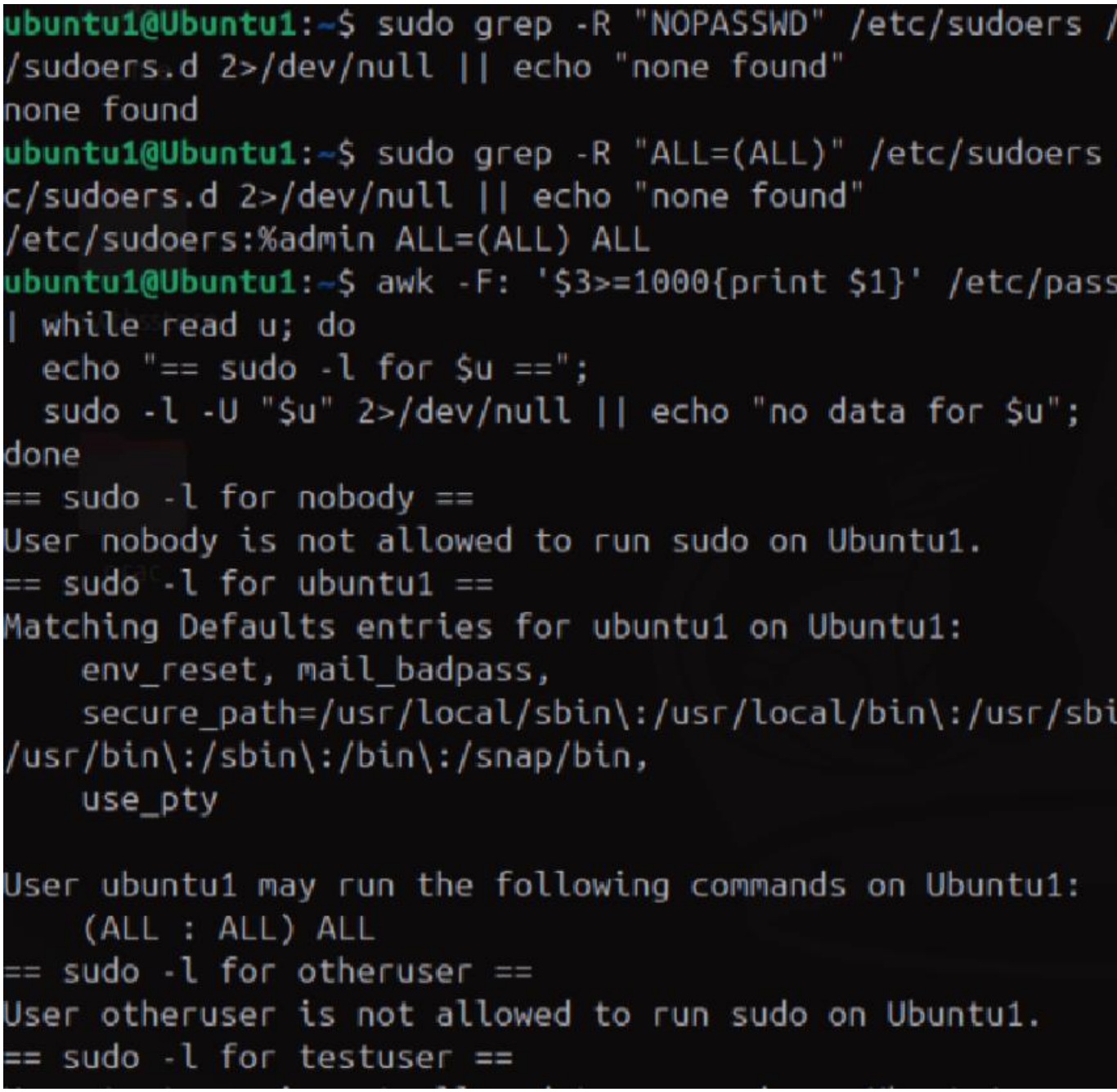
___Identifying overly permissive sudo configs (commands to find them)

Search for NOPASSWD, wildcard commands, or full ALL grants.

Find NOPASSWD entries
Find ALL=(ALL) grants to users/groups

Show per-user sudo rights for all non-system users

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



__Potential security concerns & concise improvements (apply these)

Concerns found by the commands above

NOPASSWD entries → command execution without authentication.

ALL=(ALL) grants → users can run any command as root.

Wildcard or path-writable command entries → attackers can replace binaries.

Excessive group membership (e.g., users in wheel/admin) → privilege creep.

Lack of logging/auditing → hard to trace misuse.

Recommended fixes (commands to implement or check)

1) Remove/replace NOPASSWD where not needed (use visudo to edit safely)

sudo visudo # edit /etc/sudoers safely

sudo visudo -f /etc/sudoers.d/custom # edit drop-in files safely

2) Replace ALL with specific allowed commands (example)

in visudo: alice ALL=(root) /usr/bin/systemctl, /usr/bin/journalctl

3) Force TTY & require authentication

in sudoers: Defaults requiretty (RHEL) and remove NOPASSWD entries

4) Enable sudo logging (if not already)

sudo mkdir -p /var/log/sudo && sudo chown root:root /var/log/sudo

Ensure /etc/sudoers has Defaults logfile=/var/log/sudo/sudo.log

5) Use groups and least-privilege:

sudo groupadd backupops 2>/dev/null || true

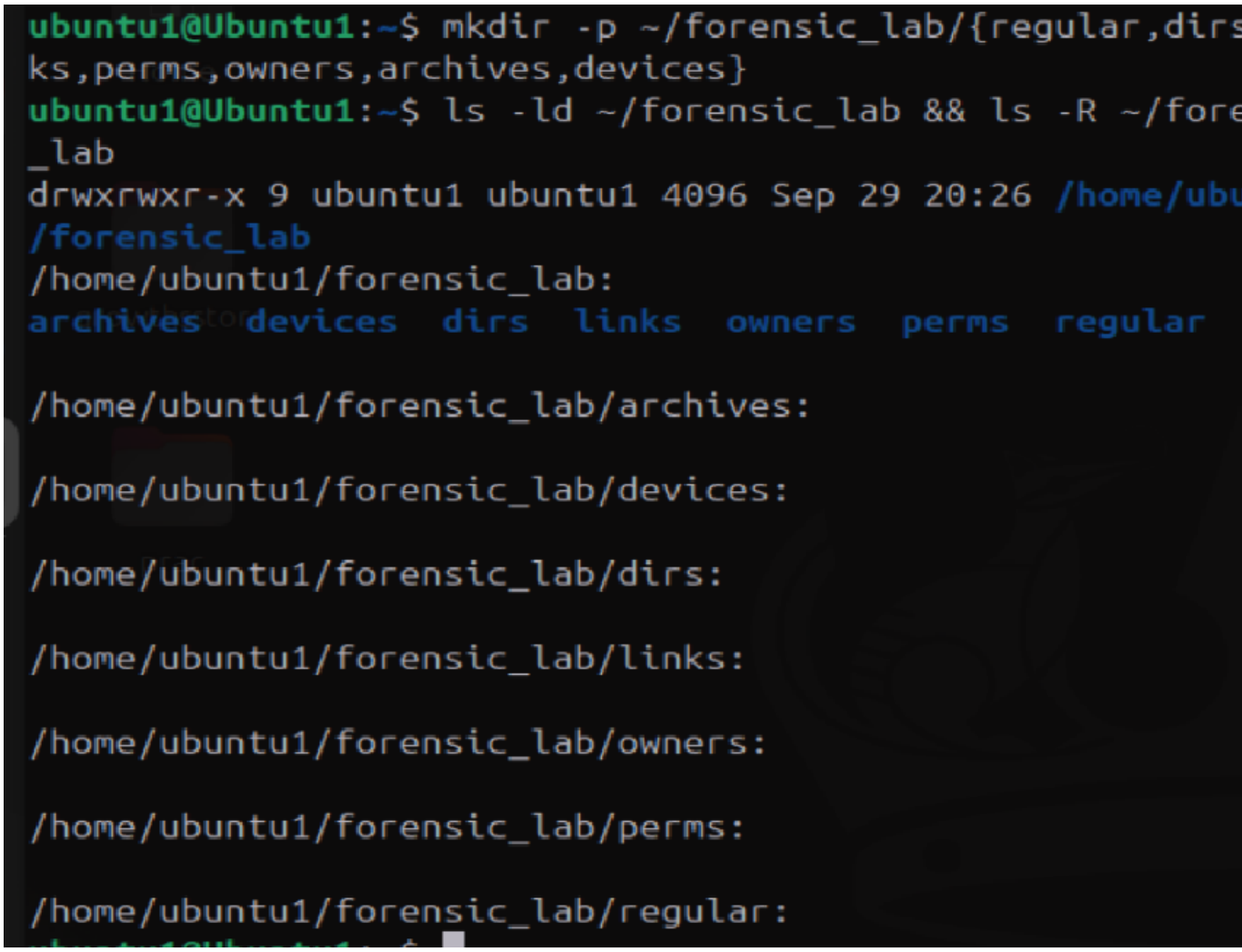
add only necessary commands to group via sudoers file for group 'backupops'

6) Enforce MFA for sudo via PAM (installation/config needed; policy change)

(policy/config step; not a single command)

Qn17: Creating forensic directory tree (all inside ~/forensic_lab)

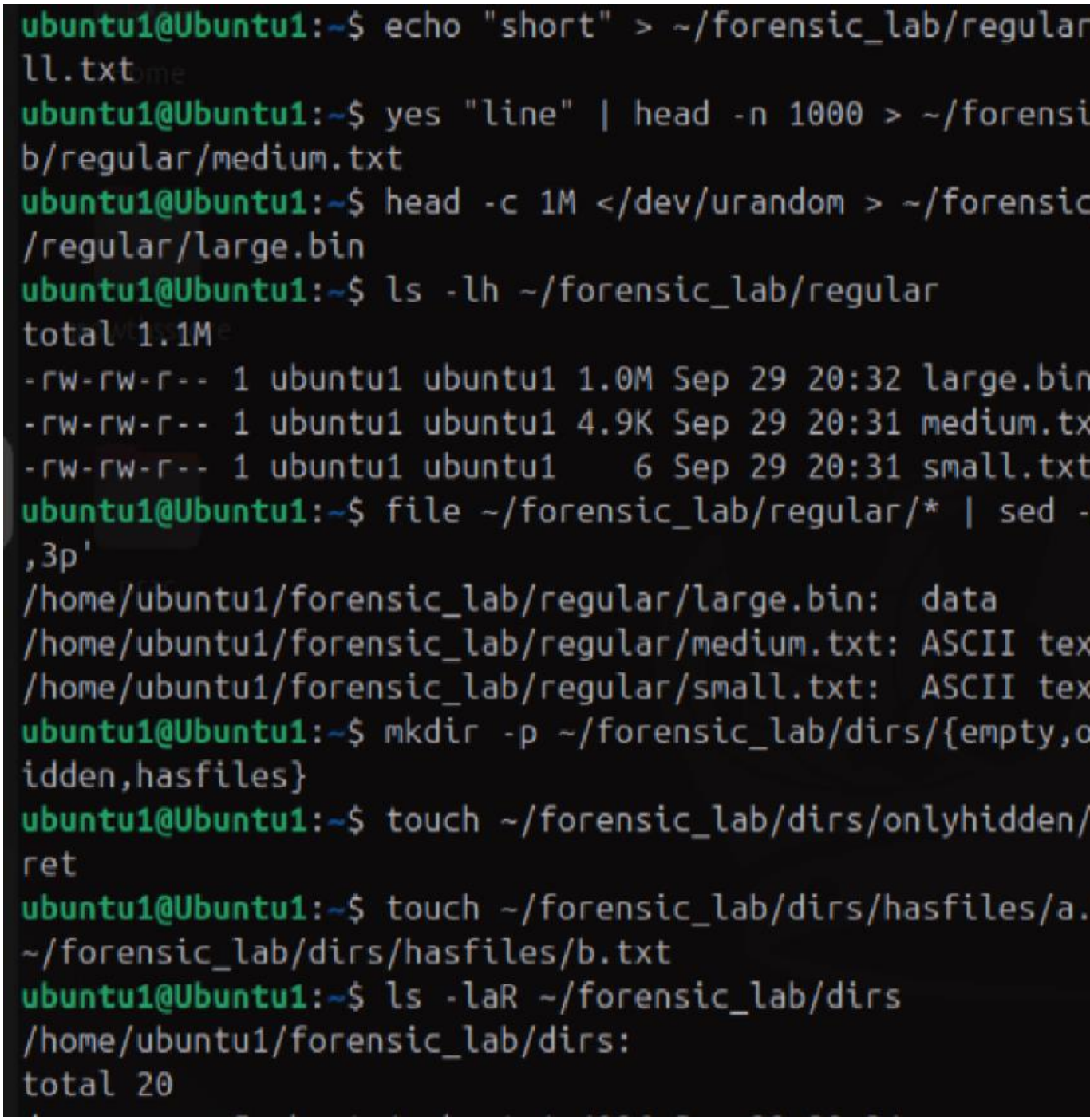
Name: Irumva Jason ID:27241 Date:Fri/26th/2025



___Regular files (small, medium, large)

Directories with content

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

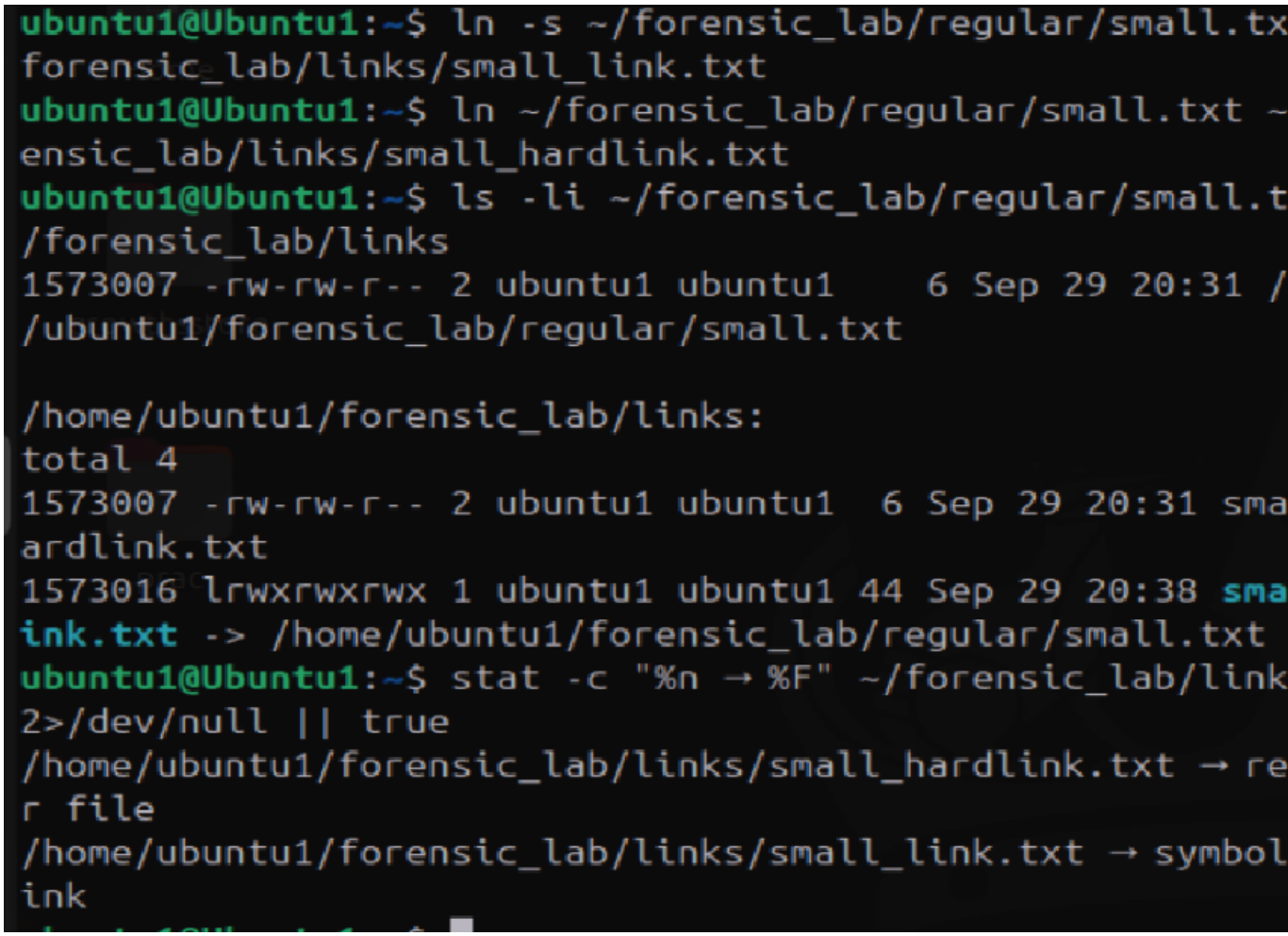
```
/home/ubuntu1/forensic_lab/dirs/empty:
total 8
drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 20:34 .
drwxrwxr-x 5 ubuntu1 ubuntu1 4096 Sep 29 20:34 ..

/home/ubuntu1/forensic_lab/dirs/hasfiles:
total 8
drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 20:34 .
drwxrwxr-x 5 ubuntu1 ubuntu1 4096 Sep 29 20:34 ..
-rw-rw-r-- 1 ubuntu1 ubuntu1    0 Sep 29 20:34 a.txt
-rw-rw-r-- 1 ubuntu1 ubuntu1    0 Sep 29 20:34 b.txt
```

___Symbolic link and hard link

Symlink

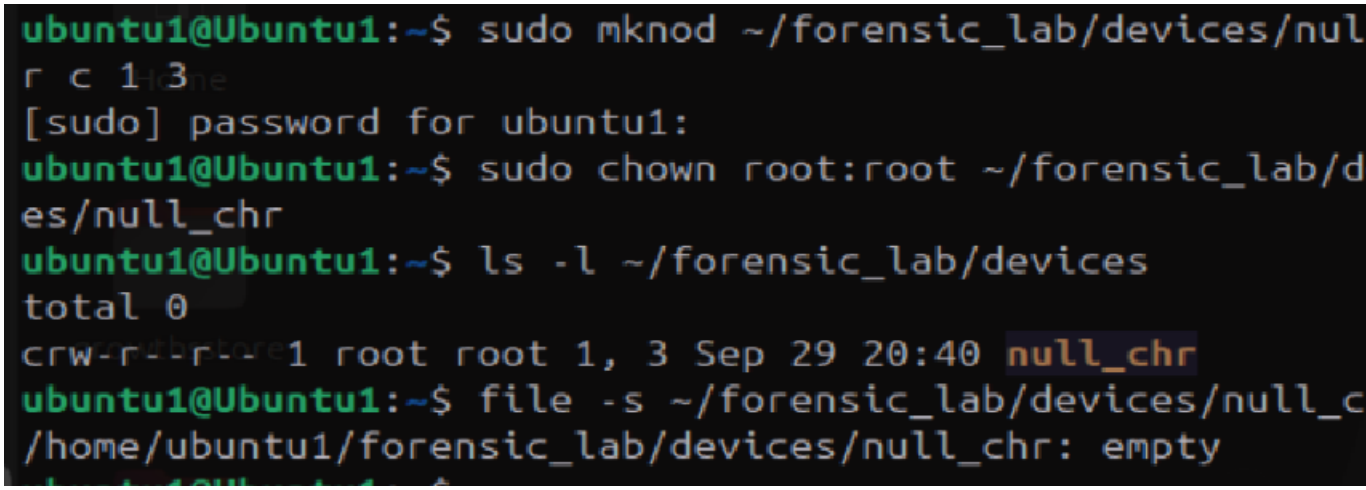
hardlink (same inode as small.txt)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



____Device file (requires root) — *simulated* character device (e.g., /dev/null-like)

creating a character device with major=1 minor=3 (null) — needs sudo

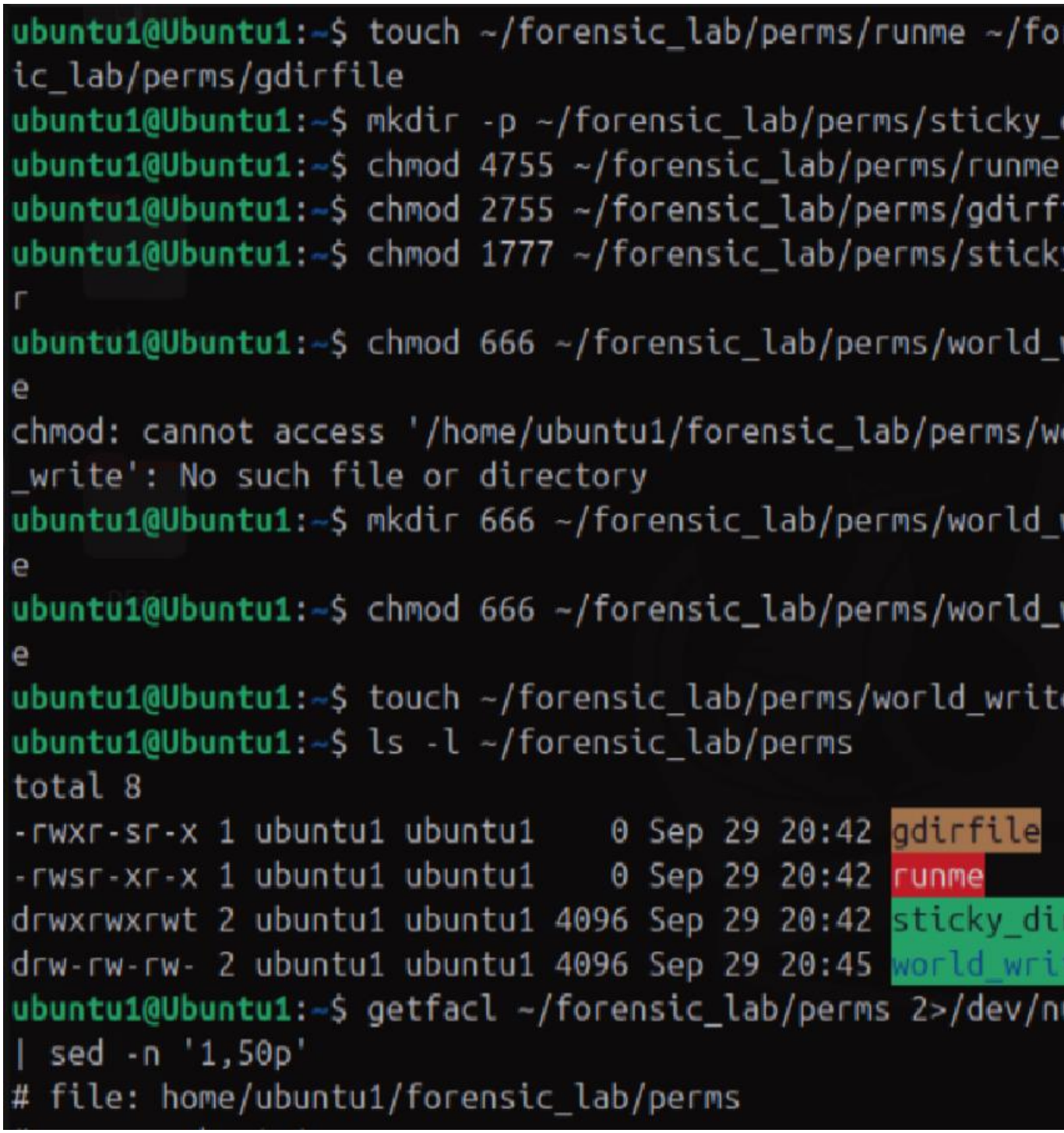


Name: Irumva Jason ID:27241 Date:Fri/26th/2025

__Permission combinations including special bits

creating target files/dirs.

Setting bits: setuid, setgid, sticky, world-writable…

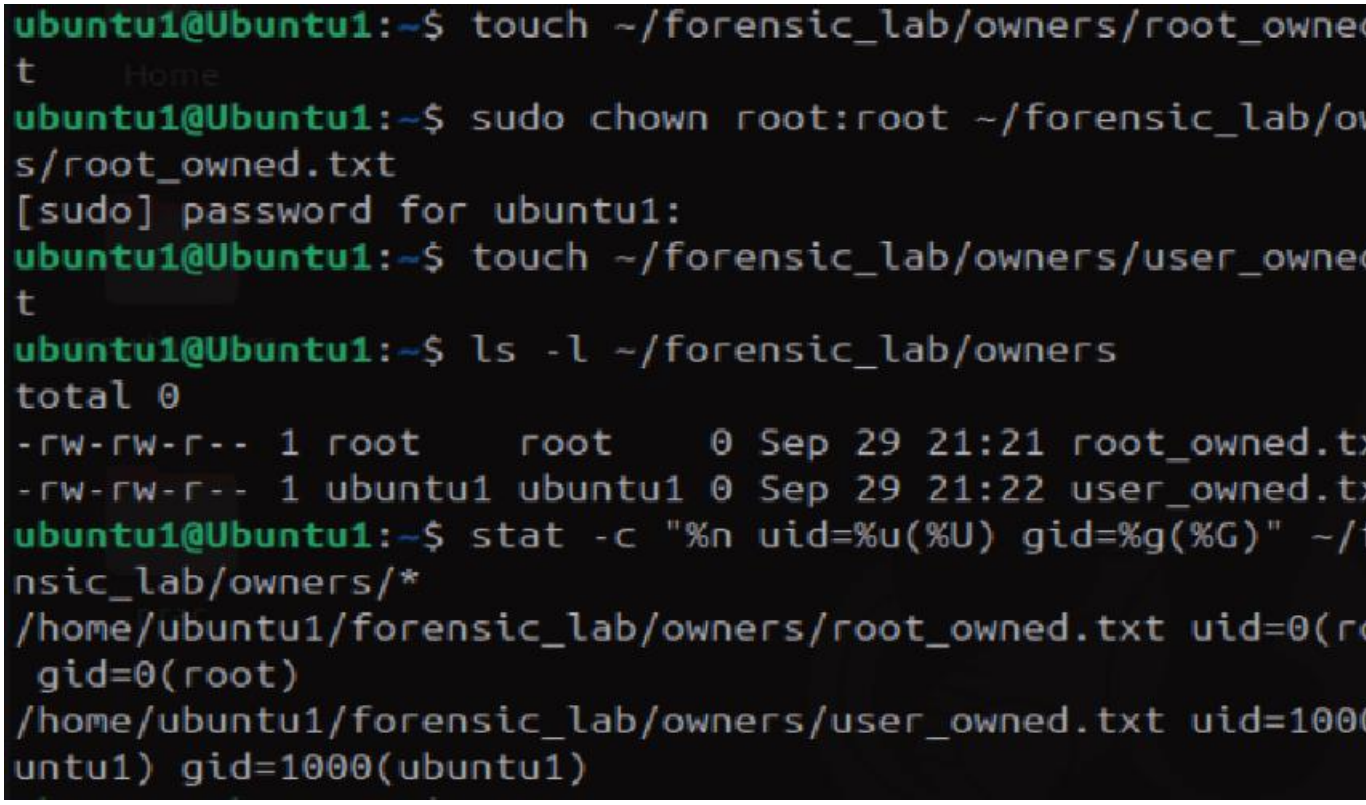


___Files with different ownership patterns

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

making a file owned by root (requires sudo)

a file owned by current user (already)



___Creating archives with different compression

creating sample files to archive

creating archives

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
.gz -C ~/forensic_lab/archives/src .
ubuntu1@Ubuntu1:~$ tar -cjf ~/forensic_lab/archives/test.
.bz2 -C ~/forensic_lab/archives/src .
ubuntu1@Ubuntu1:~$ tar -cJf ~/forensic_lab/archives/test.
.xz -C ~/forensic_lab/archives/src .
ubuntu1@Ubuntu1:~$ zip -r ~/forensic_lab/archives/test.zi
/forensic_lab/archives/src >/dev/null
ubuntu1@Ubuntu1:~$ tar -tzf ~/forensic_lab/archives/test.
.gz
./
./b.log
./a.conf
ubuntu1@Ubuntu1:~$ tar -tjf ~/forensic_lab/archives/test.
.bz2
./
./b.log
./a.conf
ubuntu1@Ubuntu1:~$ tar -tJf ~/forensic_lab/archives/test.
.xz
unzip -l ~/forensic_lab/archives/test.zip
./
./b.log
./a.conf
Archive:  /home/ubuntu1/forensic_lab/archives/test.zip
  Length      Date    Time    Name
```

```
ives/src/b.log
        5  2025-09-29 21:23   home/ubuntu1/forensic_lab/
ives/src/a.conf
---------                     -------
    16005                     3 files
ubuntu1@Ubuntu1:~$
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

___Commands to analyze each element (quick investigators' toolbox with results shown)

file type

detailed metadata (owner/perm/ctime/mtime/atime)

find setuid/setgid/sticky/world-writable files

list symlinks and their targets

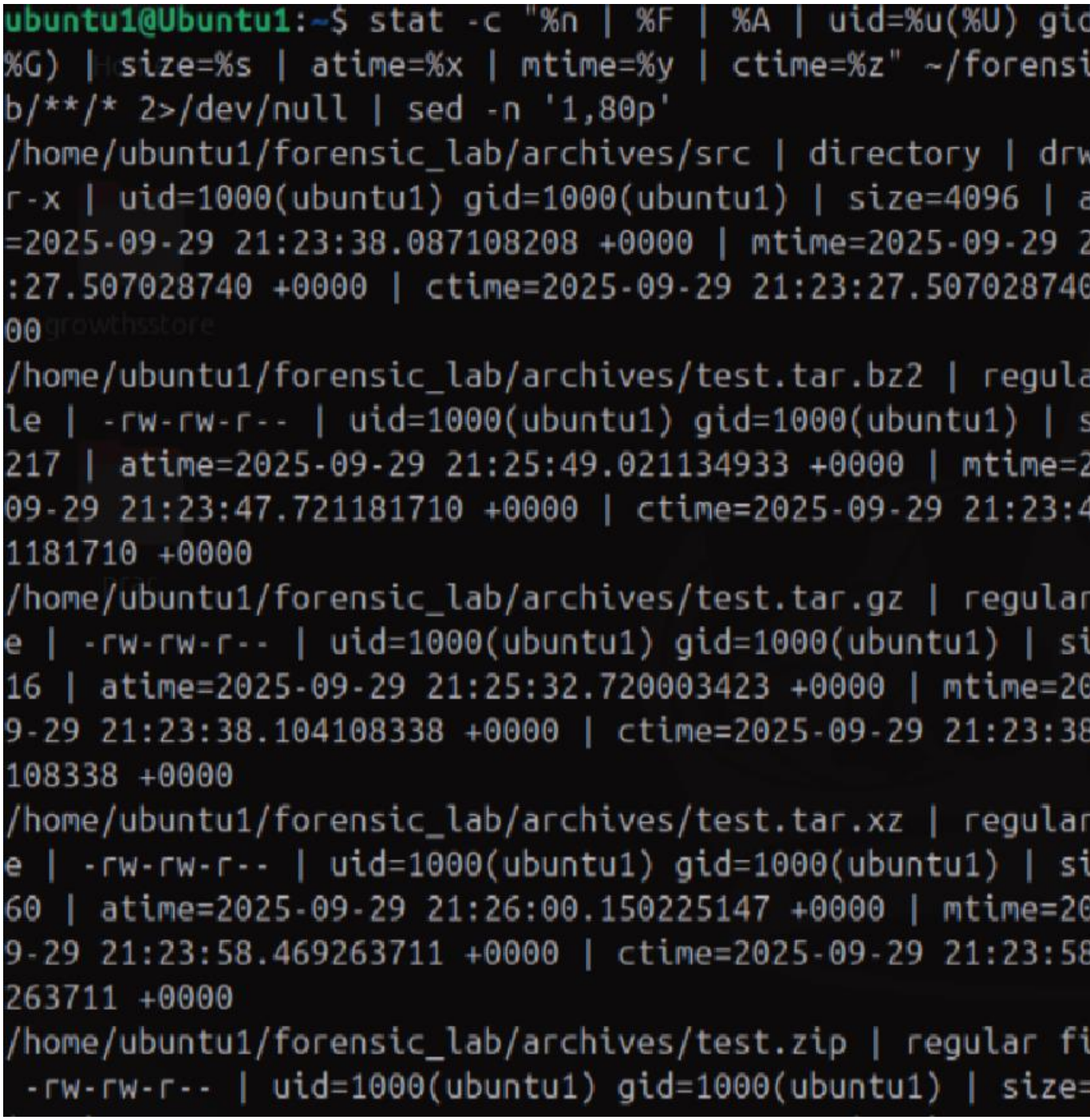
show inodes (to detect hardlinks)

examine archives without extracting (safe)

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ file -s ~/forensic_lab/**/* 2>/dev/nul
 sed -n '1,80p'
/home/ubuntu1/forensic_lab/archives/src:                dire
ry
/home/ubuntu1/forensic_lab/archives/test.tar.bz2:       bzip
ompressed data, block size = 900k
/home/ubuntu1/forensic_lab/archives/test.tar.gz:        gzip
mpressed data, from Unix, original size modulo 2^32 20480
/home/ubuntu1/forensic_lab/archives/test.tar.xz:        XZ c
ressed data, checksum CRC64
/home/ubuntu1/forensic_lab/archives/test.zip:           Zip
hive data, at least v1.0 to extract, compression method=s
e
/home/ubuntu1/forensic_lab/devices/null_chr:            empt
/home/ubuntu1/forensic_lab/dirs/empty:                  dire
ry
/home/ubuntu1/forensic_lab/dirs/hasfiles:               dire
ry
/home/ubuntu1/forensic_lab/dirs/onlyhidden:             dire
ry
/home/ubuntu1/forensic_lab/links/small_hardlink.txt: ASCI
ext
/home/ubuntu1/forensic_lab/links/small_link.txt:        symb
c link to /home/ubuntu1/forensic_lab/regular/small.txt
/home/ubuntu1/forensic_lab/owners/root_owned.txt:       empt
```

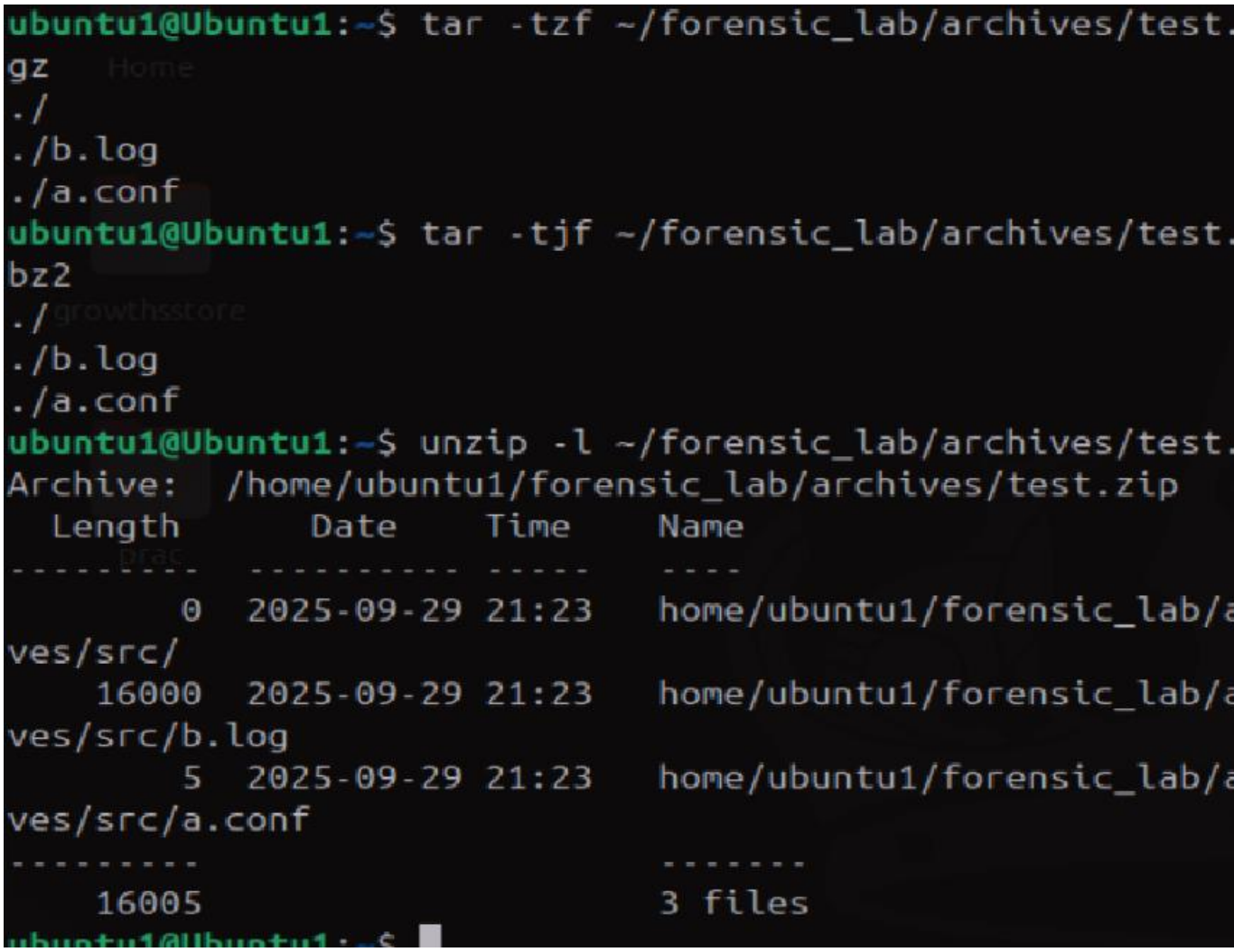
Name: Irumva Jason ID:27241 Date:Fri/26th/2025



Name: Irumva Jason ID:27241 Date:Fri/26th/2025

```
ubuntu1@Ubuntu1:~$ find ~/forensic_lab -perm /4000 -o -pe
2000 -o -perm -0002 -o -perm /1000 -ls
ubuntu1@Ubuntu1:~$ find ~/forensic_lab -type l -ls
  1573016      0 lrwxrwxrwx   1 ubuntu1  ubuntu1         4
p 29 20:38 /home/ubuntu1/forensic_lab/links/small_link.tx
 /home/ubuntu1/forensic_lab/regular/small.txt
ubuntu1@Ubuntu1:~$ ls -liR ~/forensic_lab | sed -n '1,120
/home/ubuntu1/forensic_lab:
total 28
1573005 drwxrwxr-x 3 ubuntu1 ubuntu1 4096 Sep 29 21:24 ar
es
1573006 drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 20:40 de
s
1572995 drwxrwxr-x 5 ubuntu1 ubuntu1 4096 Sep 29 20:34 di
1573000 drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 20:38 li
1573002 drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 21:22 ow
1573001 drwxrwxr-x 4 ubuntu1 ubuntu1 4096 Sep 29 20:45 pe
1572985 drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 20:32 re
r

/home/ubuntu1/forensic_lab/archives:
total 20
1573033 drwxrwxr-x 2 ubuntu1 ubuntu1 4096 Sep 29 21:23 sr
1573038 -rw-rw-r-- 1 ubuntu1 ubuntu1  217 Sep 29 21:23 te
ar.bz2
```

Name: Irumva Jason ID:27241 Date:Fri/26th/2025



____10) How an investigator uses this info (brief bullet points)

Use the outputs above to:

- Identify unexpected SUID/SGID binaries (may be used for privilege escalation).

- Detect world-writable files/dirs (can be abused to drop backdoors).

- Find device nodes in user-writable locations (suspicious; device nodes are rarely created in home dirs).

- Compare inodes to see hardlinks (hidden copies of files can be hidden via hardlinks).

- Follow symlink targets (symlink to sensitive files is a red flag).

- Check timestamps (mtime/ctime/atime) for tampering or timeline reconstruction.

- Inspect archive contents without extraction to find suspicious payloads.

Note: These are observations to draw from the commands I ran above.

Name: Irumva Jason ID:27241 Date:Fri/26th/2025

___File system artifacts that may indicate unauthorized access / compromise (brief)

- Unexpected setuid/setgid files added recently.
- Binaries with changed ctime/mtime but no package update.
- New device nodes or cron entries in nonstandard locations.
- World-writable directories where binaries/scripts are stored.
- Hidden files (dotfiles) with recent execution timestamps.
- Multiple hardlinks to suspicious binaries (hiding copies).
- Compressed archives with obfuscated names or nested payloads.
- Log files with gaps, truncated entries, or cleared logs (log tampering).