CP 211 Practical Test (2025/2026) - Question 2 Script Explanation

Reference Question (Question 2: User Account Verification Script)

- Accept a username as input.
- Check whether the username exists in /etc/passwd.
- If the user exists, display the user ID information.
- If the user does not exist, display: "User does not exist".

Script Under Review: cp211_practical_solutions/q2.sh

```
1  #!/usr/bin/env bash
2  set -euo pipefail
3
4  read -r -p "Enter username: " username
5
6  if awk -F: -v user="$username" '$1 == user {found=1; exit} END {exit !found}' /etc/passwd; then
7    id "$username"
8  else
9    echo "User does not exist"
10 fi
```

How the Script Solves Each Required Part

(i) Accepts a username as input

- Line 4 uses:
  ```
  read -r -p "Enter username: " username
  ```
- Command details:
  - read: reads user input from the terminal.
  - -p: prints a prompt before reading.
  - -r: keeps backslashes literal (safer input handling).
  - username: variable that stores what the user typed.

(ii) Checks whether the username exists in /etc/passwd

- Line 6 uses awk on /etc/passwd:
  ```
  awk -F: -v user="$username" '$1 == user {found=1; exit} END {exit !found}' /etc/passwd
  ```
- Command details:
  - awk: text-processing tool.
  - `-F:` sets the field separator to colon because /etc/passwd is colon-separated.
  - -v user="$username": passes shell variable into awk.
  - $1 == user: compares first field (username column) with input username.
  - found=1; exit: stop early once match is found.
  - END {exit !found}: returns status 0 if found, 1 if not found.

(iii) If user exists, display user ID information

- Line 7 runs:
  ```
  id "$username"
  ```
- Command details:
  - id shows UID, GID, and group memberships for that user.
  - This is exactly the required "user ID information" command.

(iv) If user does not exist, display required message

- Line 9 runs:
  ```
  echo "User does not exist"
  ```
- Command details:
  - echo prints the exact required message.

Control Flow Summary

- The if statement checks the exit status of awk.
- Exit status 0 (username exists) -> run id "$username".
- Non-zero exit status (username missing) -> print "User does not exist".

Additional Script Safety (Not explicitly required, but good practice)

- Line 2: set -euo pipefail
  - -e: stop script on command failure.
  - -u: treat unset variables as errors.
  - -o pipefail: fail pipeline if any stage fails.

Example Run 1 (existing user)

- Input: root
- Output (example format):

```
uid=0(root) gid=0(root) groups=0(root)
```

Example Run 2 (missing user)

- Input: user_not_present_123
- Output:

```
User does not exist
```

Conclusion

- The script in cp211_practical_solutions/q2.sh correctly satisfies all Question 2 requirements.
- It also uses robust shell options for safer execution.