

The background features a dark blue gradient with faint, light blue concentric circles and degree markings (40, 150, 160, 170, 180, 190, 200, 210, 220, 230, 240, 250, 260) on the left side, suggesting a technical or scientific theme.

CS 2080: PROGRAMMING WITH UNIX

DR. ARMIN MOIN, ASSISTANT PROFESSOR

QUANTUM-CLASSICAL AI AND SOFTWARE ENGINEERING (QAS) LAB

DEPARTMENT OF COMPUTER SCIENCE

UNIVERSITY OF COLORADO COLORADO SPRINGS (UCCS)

SUMMER 2024

LECTURE 4: LINUX BASH SHELL COMMANDS I + LINUX FILE PERMISSIONS

- Outline:
 - Switching user
 - Installation
 - Information
 - Communication
 - Comparison
 - File management and permissions
 - Miscellaneous
 - SSH keys

SWITCHING USER

- *su*
 - Switch to a different user, e.g., *su alice*
 - Switch to the superuser (root): *su* or *su -*
- *sudo*
 - Switch to the superuser (root) without the need to enter the root password, if the user is listed in */etc/sudoers*
 - To add a user to the sudoers list: *usermod -aG sudo username*
- Reminder: On your AWS EC2 VMs, set the root password through *sudo passwd root*
- *whoami*
 - Print the name of the current user

INSTALLATION

- *sudo apt install*: Install a new software package
- *apt search*: Search for a software package
- */etc/apt/sources.list*
- Ubuntu sources have moved to */etc/apt/sources.list.d/ubuntu.sources*

INSTALLATION

- `/etc/apt/sources.list`
 - Main - Canonical-supported free and open-source software.
 - Universe - Community-maintained free and open-source software.
 - Restricted - Proprietary drivers for devices.
 - Multiverse - Software restricted by copyright or legal issues.

UPGRADE

- *sudo apt update*
- *sudo apt list --upgradable*
- *sudo apt upgrade*

INFORMATION

- *man: Get information on a command (You will need to read the man pages!)*
 - *E.g., man su*
- *info: The GNU info system for online documentation.*
 - *E.g., info su*
- *help: Information on built-in commands.*
 - *E.g., help **echo***
 - *Write help to see a list of these commands!*

COMMUNICATION

- *ssh*: Connect to another system securely via SSH
- *ftp*: Interactive file transfer program
- *login*: Sign on to Linux (open a new session)
- *slogin*: Sign on to remote Linux using secure shell (i.e., SSH)
- *mailx*: Read or send emails (if not installed, first run *sudo apt install mailutils*)
- *wget*: Download resources from the Internet
- *curl*: Transferring from/to a server)

COMPARISON

- *cmp*: Compare two files byte by byte (by default, reports the location of the first mismatch)
- *comm*: Compare items in two sorted files line by line
- *diff*: Compare two files line by line; or compare two directories
 - Options: *--ignore-case*, *--ignore-blank-lines* or *-B*, *--ignore-space-change* or *-b*, *--ignore-all-space* or *-w*
- *sdiff*: Compare two files side by side
- *diff3*: Compare three files

FILE MANAGEMENT AND PERMISSIONS

- *chmod*: Change access modes on files (Use the `-R` option for directories and their content)
 - User (u), group (g), others (o)
 - Read (r), write (w), execute (x)
 - `-rwxrwxrwx`

FILE MANAGEMENT AND PERMISSIONS - EXAMPLES

- *E.g., `chmod u-x` -> `-rw-rwxrwx`*
- *E.g., `chmod o-r` -> `-rw-rwx-wx`*
- *E.g., `chmod u+x` -> `-rwxrwx-wx`*

FILE MANAGEMENT AND PERMISSIONS

- Alternative to r, w, and x: Numeric codes:
 - 0 = No permission
 - 1 = Execute
 - 2 = Write
 - 4 = Read

FILE MANAGEMENT AND PERMISSIONS

- For example, for -rw-----

User: r and w, thus $4+2=6$

Group: 0

Others: 0

Hence: 600

FILE MANAGEMENT

- *chown*: Change file owner (Use the `-R` option for directories and their content)
- *chgrp*: Change file group (Use the `-R` option for directories and their content)
- *cksum*: Print a file checksum, POSIX standard algorithm
- *md5sum*: Print a file's checksum using the Message Digest 5 (MD5) algorithm
- *cp*: Copy files (*cp -r* for copying directories)

FILE MANAGEMENT

- *cd*: Change directory (requires execution permissions in that directory)
- *file*: Determine a file's type (e.g., ASCII text)
- *head*: Show the first few lines of a file (use the `-n` option to specify the number of lines)
 - E.g., `head -n 10`
- *tail*: Show the last few lines of a file (use the `-n` option to specify the number of lines)
 - E.g., `tail -n 10`

FILE MANAGEMENT

- *more*: Display files by screenful
- *less*: Similar to more, but more sophisticated
- *ls*: List files or directories
 - `ls -l`
 - `ls -a`

FILE MANAGEMENT

- *ln*: Create file name aliases
 - By default, a hard-link (must be on the same filesystem). Still works after moving the target!
 - If *-s* or *-symbolic* used, a separate file (sort of a “shortcut”) created. Moving would break it!
- *locate*: *locate* or *plocate* is a fast way to find a file based on its name (matching pattern)
- *mkdir*: Create a directory

FILE MANAGEMENT

- *mv*: Move or rename files or directories
- *rm*: Remove files (use *-r* or *rmdir* for removing directories)
- *pwd*: Print the working directory's path

FILE MANAGEMENT

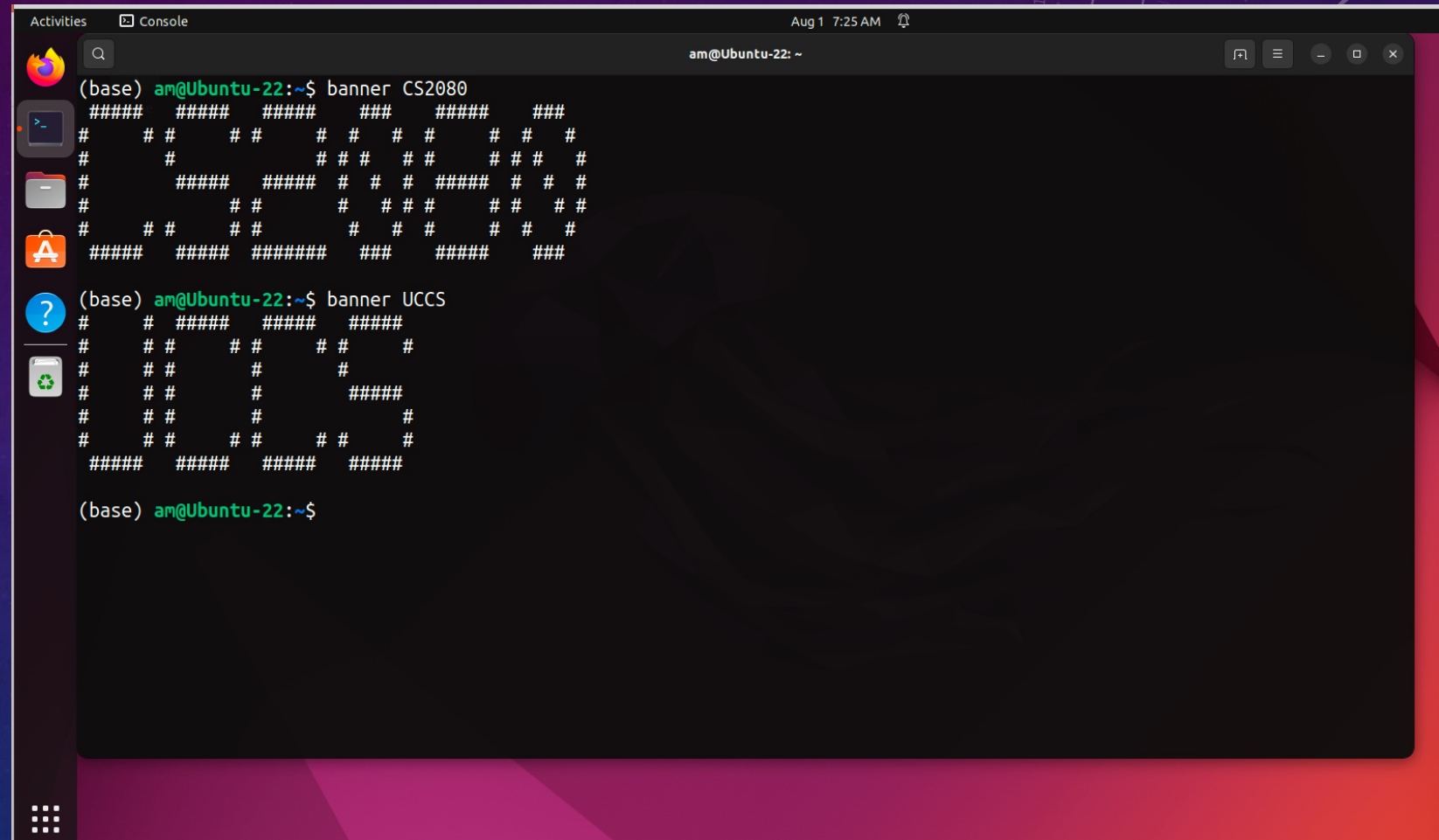
- *scp*: Copy files to a remote system securely using SSH
- *csplit*: Break files at specific locations
 - *E.g., csplit f1.txt 2 (i.e., from line 2)*
- *split*: Split files evenly
 - Default size: 1000 lines

FILE MANAGEMENT

- *wc*: Count a file's lines, words, and characters (use `-m` for characters, `-w` for words, and `-l` for lines) – by default lines
- *cat*: Concatenate files or print the content (use `>>` to append or `>` to substitute content)
 - `cat f1`
 - `cat f1 >> f2`
 - `cat f1 > f2`
- *nano*: Simple text editor (We'll talk about editors in Lecture 7 and Lab 5)

MISCELLANEOUS

- banner



The screenshot shows a terminal window titled "am@Ubuntu-22: ~" with the date and time "Aug 1 7:25 AM". The terminal displays the output of the 'banner' command for two different identifiers: CS2080 and UCCS. The output for CS2080 is a 6x12 grid of characters, and the output for UCCS is a 6x12 grid of characters. The terminal also shows the prompt "(base) am@Ubuntu-22:~\$" and the command "banner CS2080" and "banner UCCS".

```
(base) am@Ubuntu-22:~$ banner CS2080
#####  #####  #####  ###  #####  ###
#  #  #  #  #  #  #  #  #  #  #  #  #
#  #  #  #  #  #  #  #  #  #  #  #
#  #####  #####  #  #  #  #####  #  #  #
#  #  #  #  #  #  #  #  #  #  #  #
#####  #####  #####  ###  #####  ###

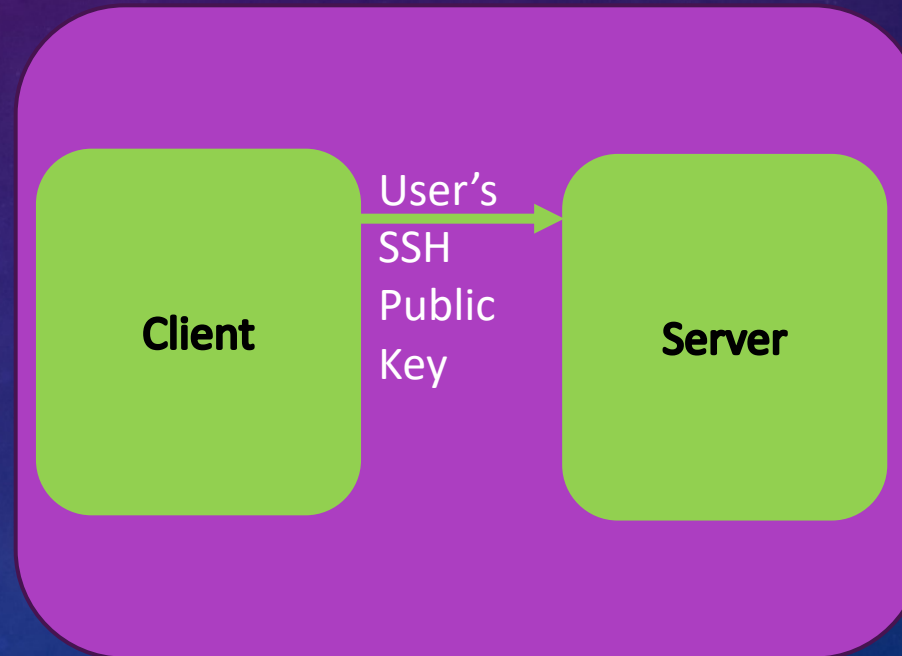
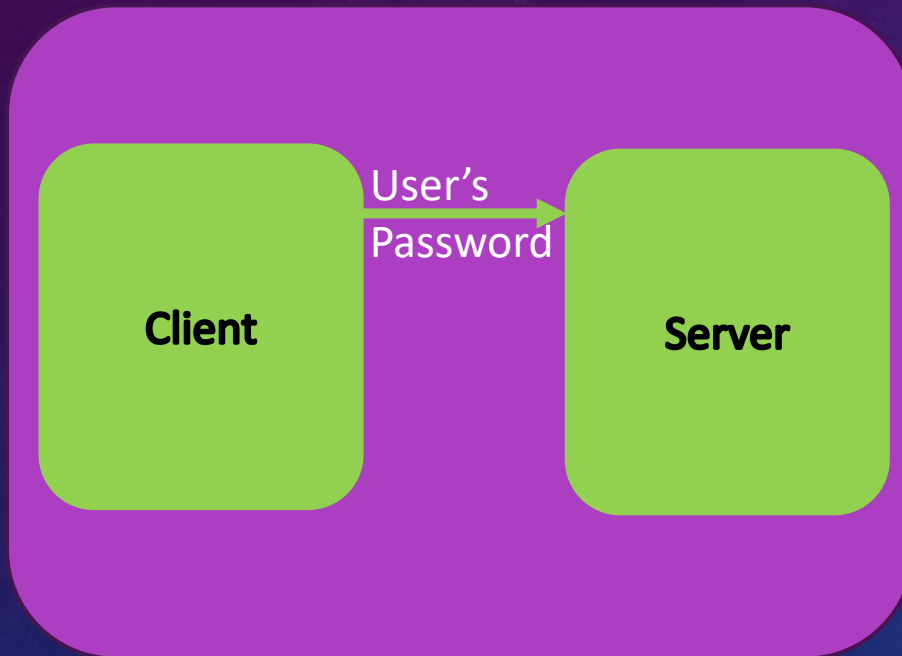
(base) am@Ubuntu-22:~$ banner UCCS
#  #  #####  #####  #####
#  #  #  #  #  #  #
#  #  #  #  #  #
#  #  #  #  #  #####
#  #  #  #  #  #
#  #  #  #  #  #
#####  #####  #####  #####

(base) am@Ubuntu-22:~$
```

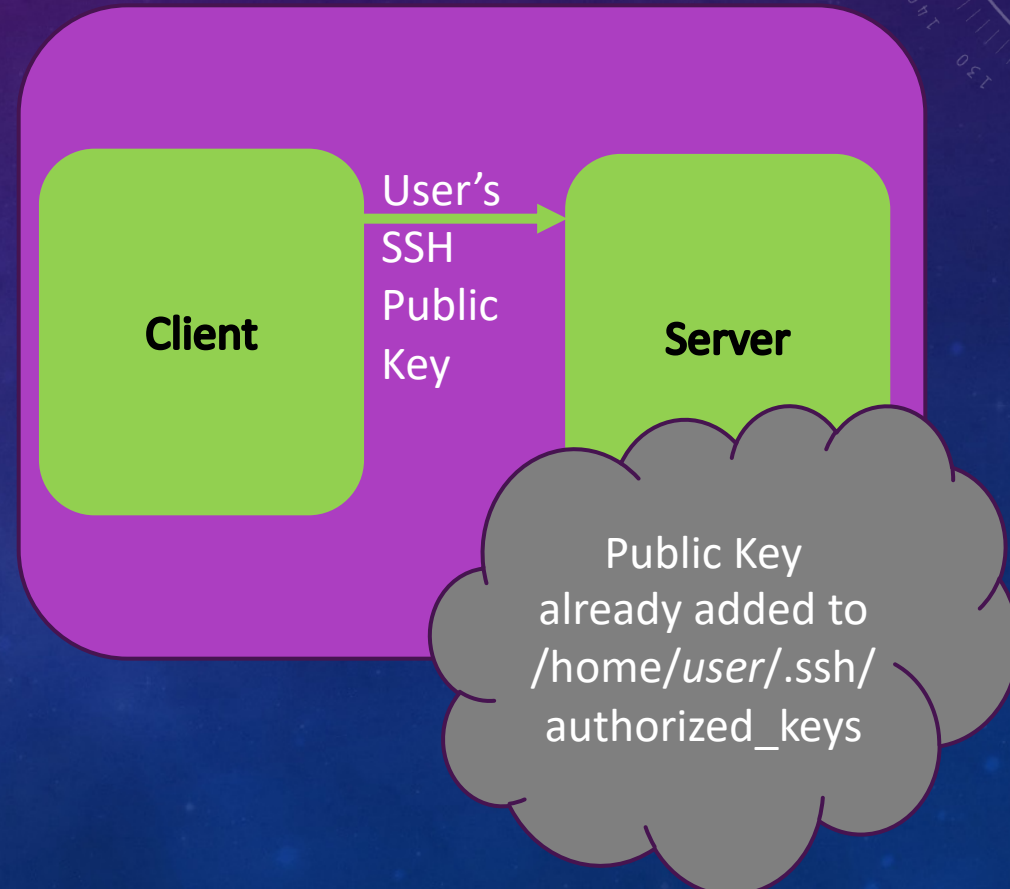
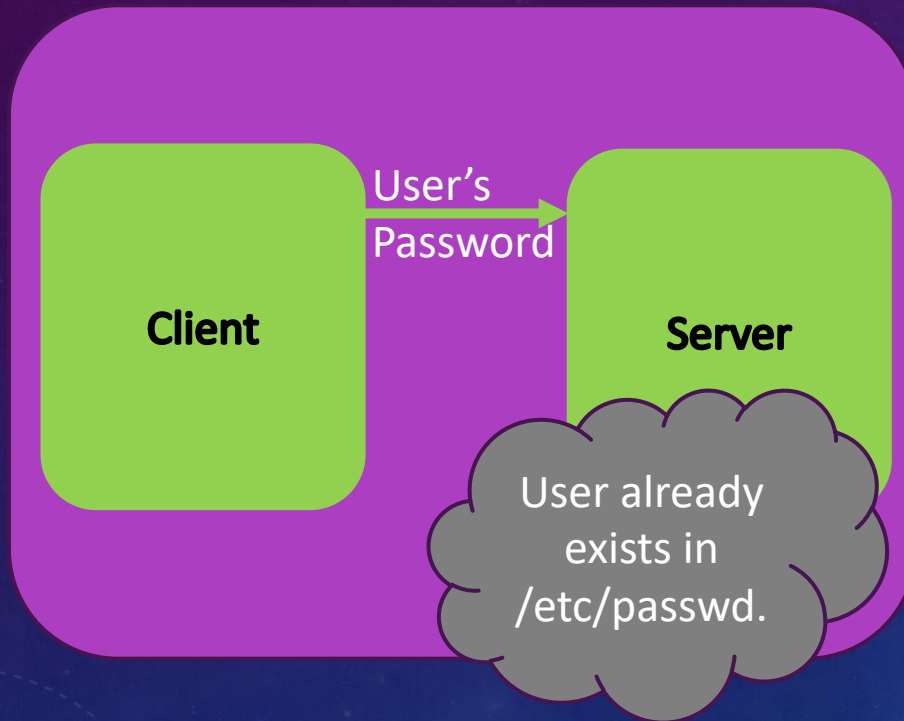
SSH KEYS

- A key pair: public key + private key for each machine
- Pairing enabled by a cryptographic algorithm
- Public key: To be shared globally
- Private key: To be kept secret (never share)
- The concept is known as **asymmetric** cryptography or **public key** cryptography
- Use cases: password-less (more secure) authentication and digital signatures (e.g., OpenPGP)
- Here, we focus on the former (**authentication**)

SSH KEYS: AUTHENTICATION WITH AND WITHOUT THEM



SSH KEYS: AUTHENTICATION WITH AND WITHOUT THEM



CREATING AN SSH KEY PAIR/IMPORTING A PUBLIC KEY

- Creating an SSH key pair
 - AWS Dashboard: We'll see how to do this in Lab 2.
 - The *ssh-keygen* command in the terminal
 - Other tools (e.g., PuTTY, see <https://www.putty.org>)
- Importing an existing SSH public key (We'll see how to do this in Lab 2)
 - AWS Dashboard
 - Append the public key to `/home/user/.ssh/authorized_keys`

QUESTIONS?

See you!