System Programming
(ELEC462)

Lab #1

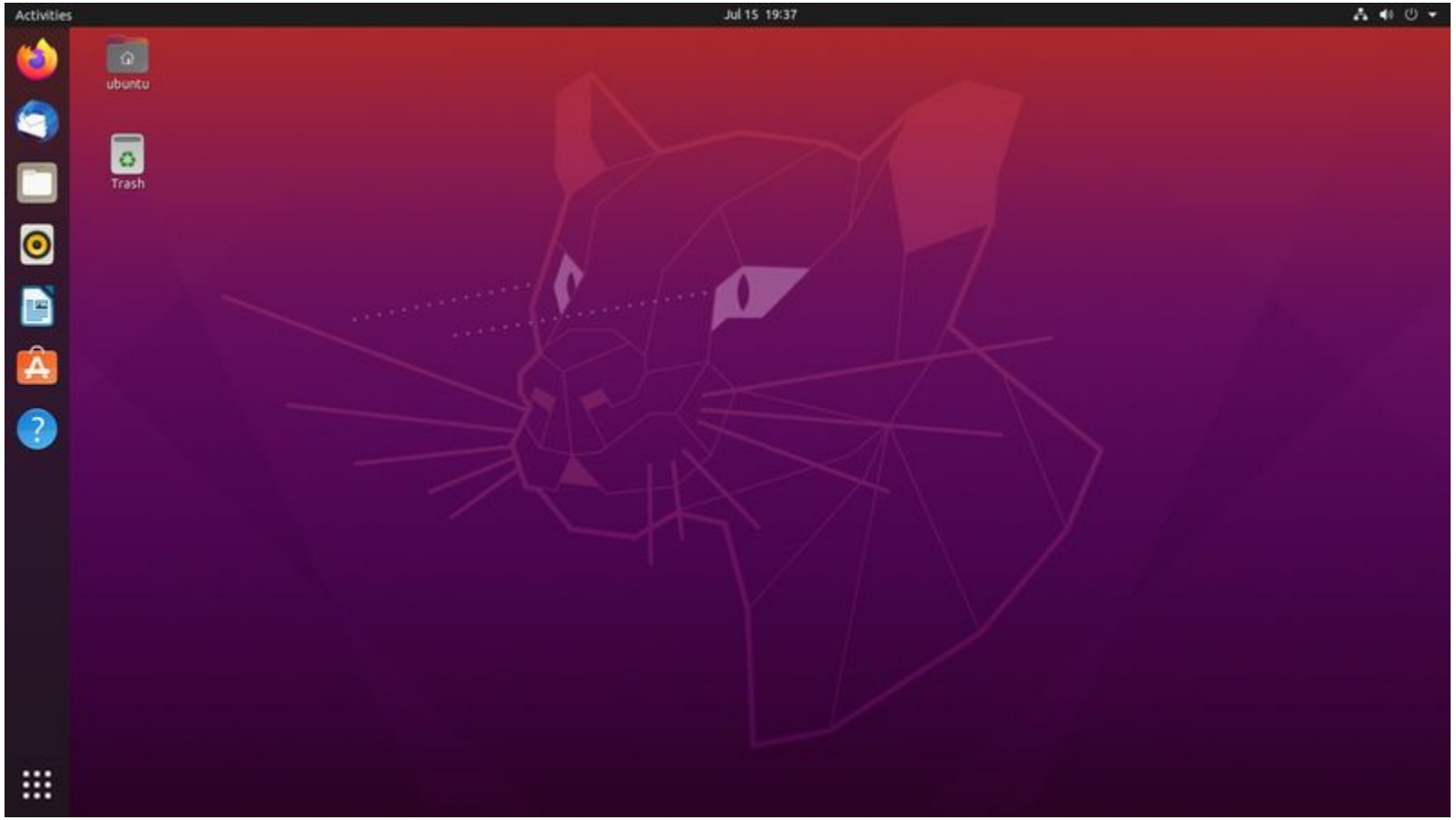
Dukyun Nam

KNU

Contents

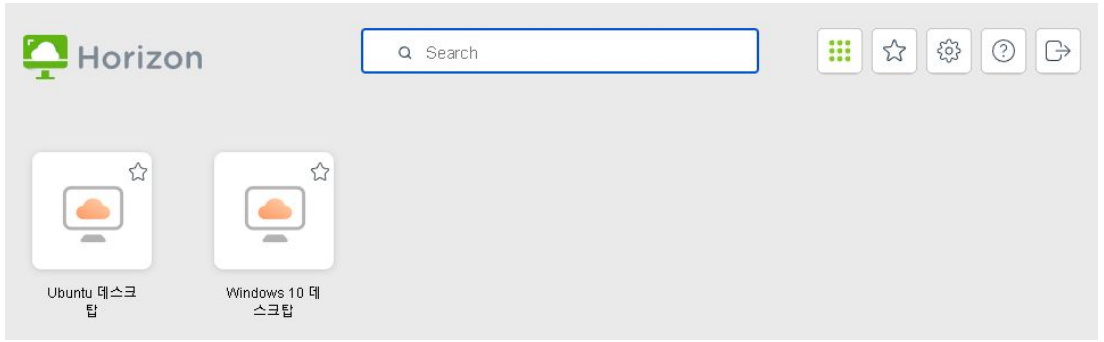
Part 1: Use Ubuntu

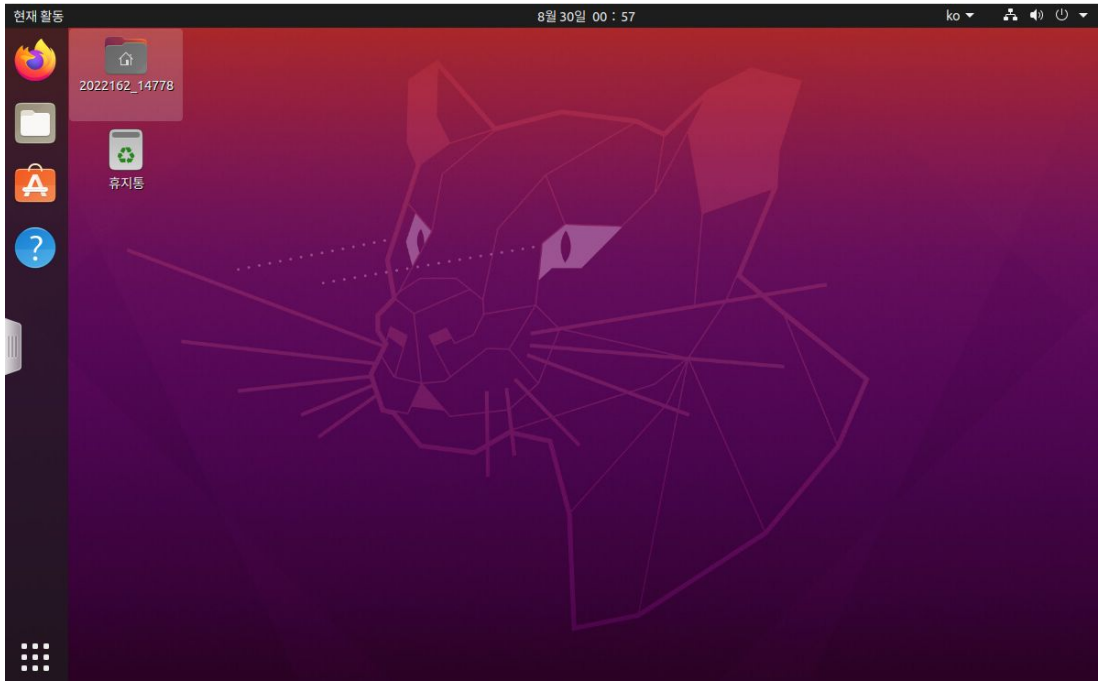
- Option 1) Use a Linux box itself



Part 1: Use Ubuntu (cont.)

- Option 2) Virtual lab on LMS
  - Virtual Desktop Infrastructure





Part 1: Use Ubuntu (cont.)

- Option 3) Windows Subsystem for Linux
  - https://docs.microsoft.com/en-us/windows/wsl/
  - “Windows Subsystem for Linux (WSL) lets developers run a GNU/Linux environment directly on Windows, unmodified, without the overhead of a traditional virtual machine or dual-boot setup.”

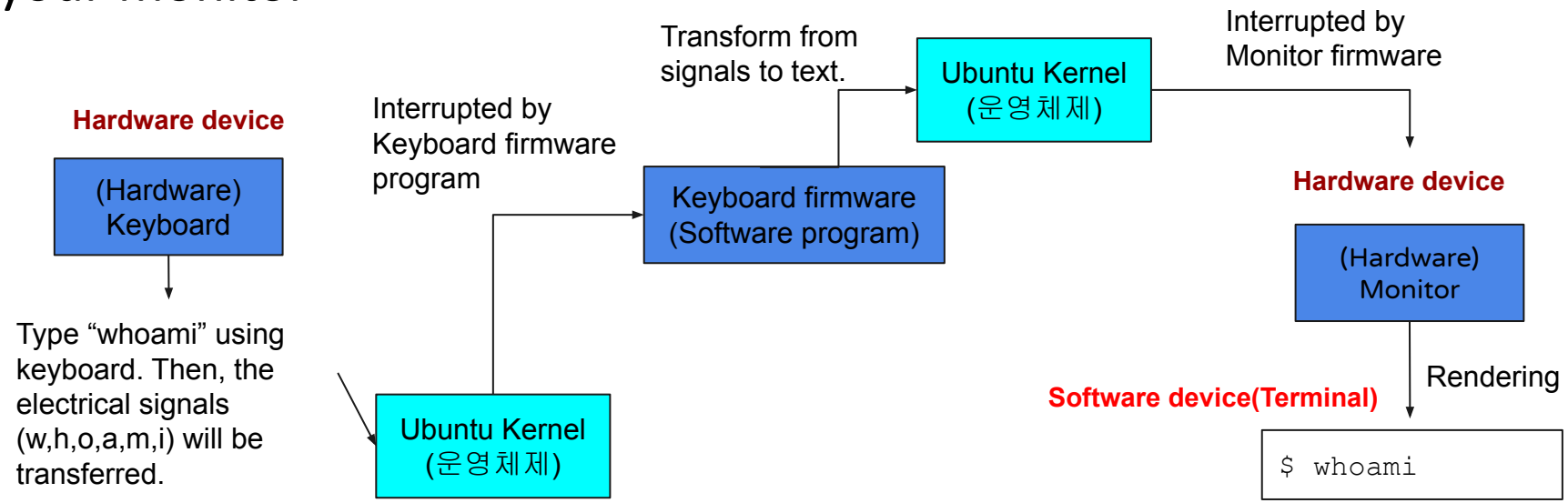
```
# Install WSL in an administrator PowerShell or Windows Command Prompt
C:\Users\user> wsl --install

# To see the available Linux distros
C:\Users\user> wslconfig.exe /l
```

Part 2: Hello World

- Procedure
  - Step 1: Open up a terminal
  - Step 2: Make and go to a specific directory
  - Step 3: Create a file using an editor program, called `vim`
  - Step 4: Write some code in `vim` editor
  - Step 5: Quit the editor
  - Step 6: Compile the code
  - Step 7: Run the code on Terminal

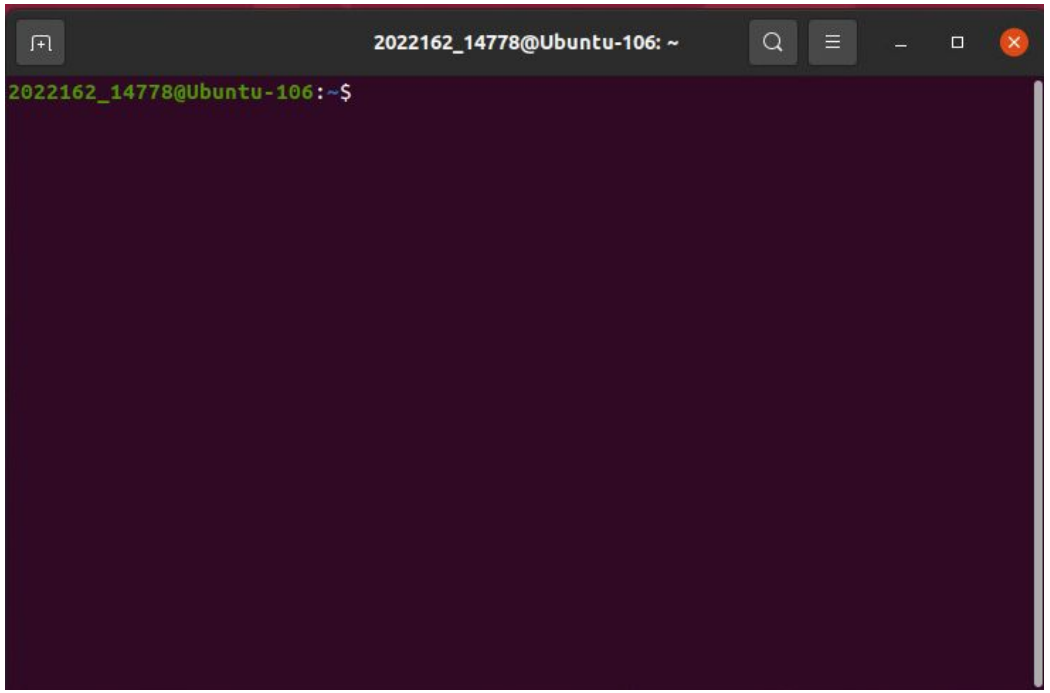
Part 2: Hello World (cont.)

- What is *Terminal*?
  - Terminal is called *device*
    - e.g., Devices: mouse, keyboard, speaker, and wireless network(WiFi)
  - If you type some text on Terminal using keyboard, the text will appear to your monitor



Part 2: Hello World (cont.)

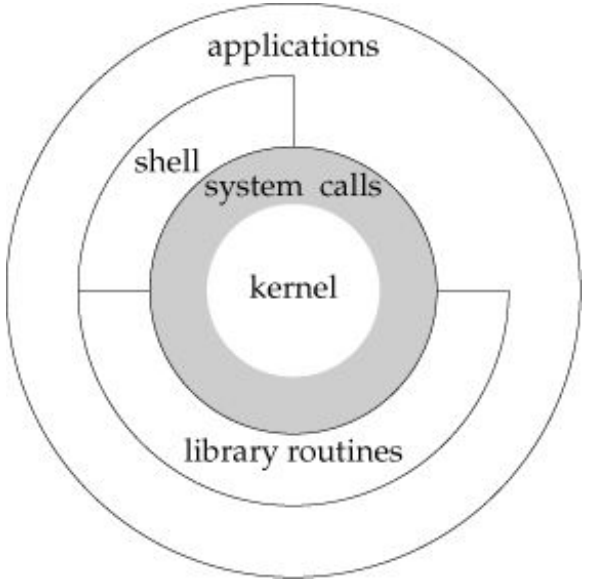
- Step 1) Open up a terminal
  - *username@hostname:<path>*$
    - Prompt ($): accept a "command line" typed by the user terminated by the `Enter` key




```
$ whoami
$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 20.04.5 LTS
Release:        20.04
Codename:       focal

$ uname -a
```

Part 2: Hello World (cont.)

- What is a “shell”?
  - After logging in, Linux/Unix starts another program called the shell
  - The shell interprets commands the user types and manages their execution
    - The shell communicates with the internal part of the operating system called the **kernel**
    - The most popular shells are: tcsh, csh, korn, and bash
    - The differences are most times subtle
  - Shell commands are **CASE SENSITIVE**!



Part 2: Hello World (cont.)

- Step 2~7) Live demo
  - Sample commands

```
# Writing a helloworld program
$ mkdir lab1
$ cd lab1
$ vi helloworld.c
$ gcc helloworld.c
$ ./a.out
```

Lab #1: Submission

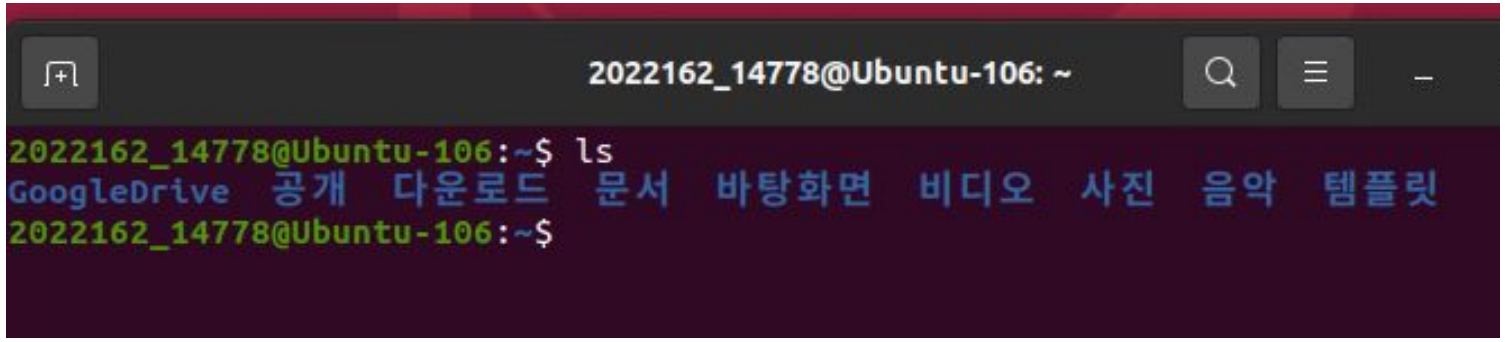
- How to submit?
  - Make sure you have your code and screenshot in a directory (`lab1`): `helloworld.c` and `helloworld.jpg(png)`
  - Change your directory name (`lab1`) to another using a series of the following commands:
    - `cd ..`
    - `mv lab1 lab1_s<Your_Student_ID>`
      - Assume your ID is 2022000000.
      - An example command: `mv lab1 lab1_s2022000000`
  - Zip your folder:
    - `zip -r lab1_s2022000000.zip lab1_s2022000000`
  - Upload the zipped directory (`lab1_s2022000000.zip`) into LMS

Optional

Mount a Remote Directory

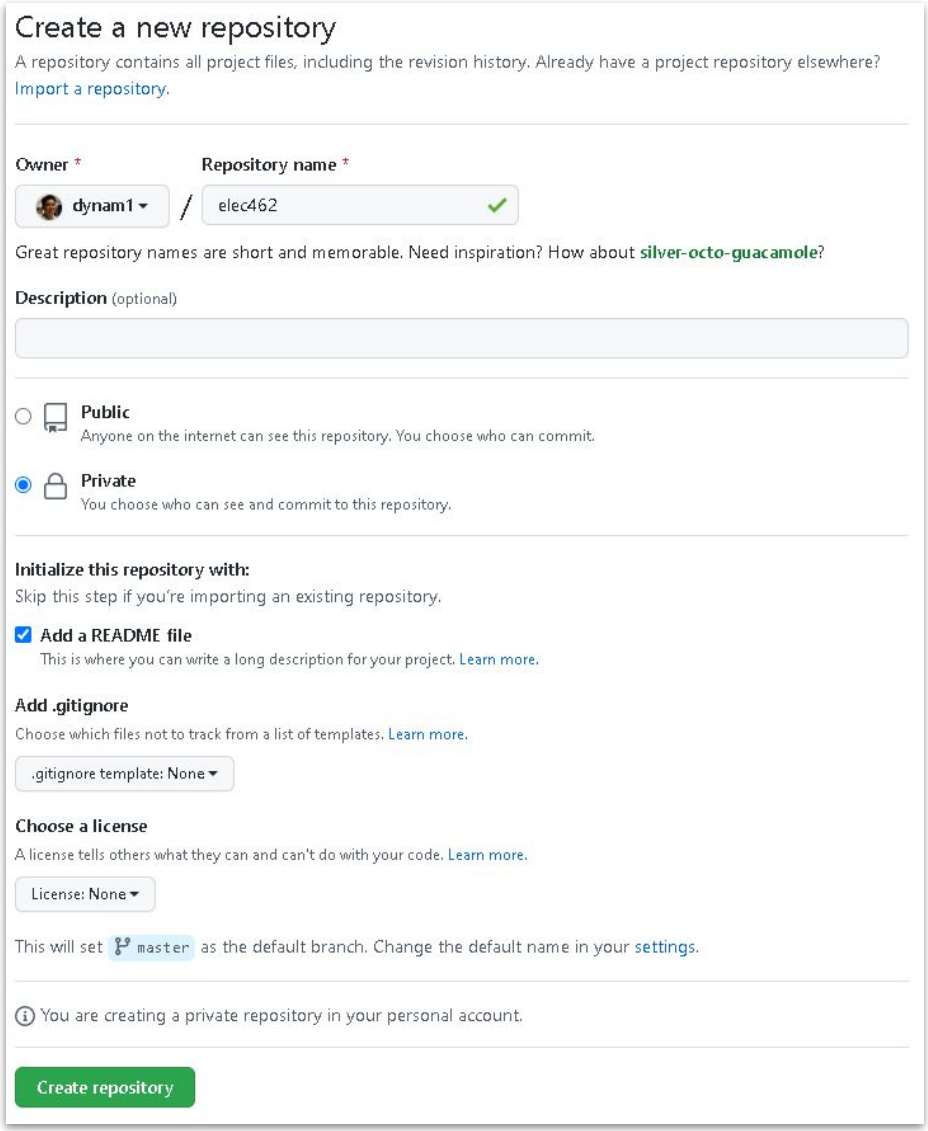
- Option 1) Google Drive
  - Use *google-drive-ocamlfuse* which is one of filesystems in a user space

```
$ sudo add-apt-repository ppa:alessandro-strada/ppa
$ sudo apt update
$ sudo apt install google-drive-ocamlfuse
$ mkdir ~/GoogleDrive
$ google-drive-ocamlfuse ~/GoogleDrive
```



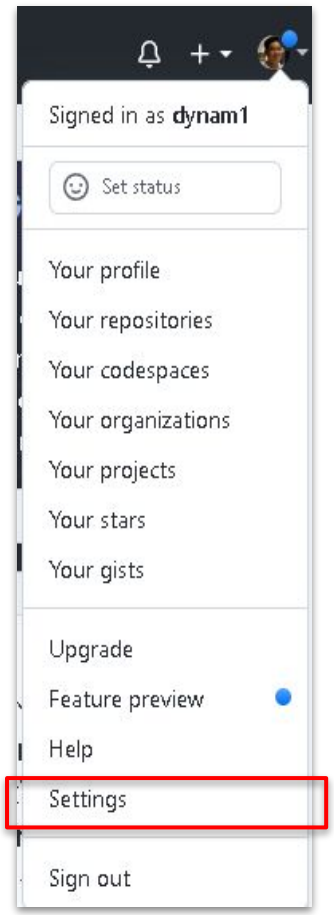
Mount a Remote Directory (cont.)

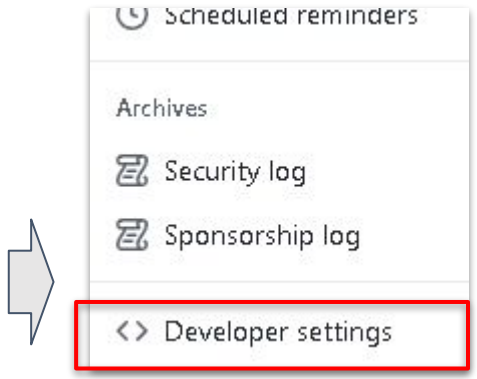
- Option 2) GitHub
  - 1. Sign up for GitHub
  - 2. Create a private repository
    - Repository name: “elec462”
  - 3. Generate a personal access token
  - 4. Clone a repository with the token
  - 5. Use it

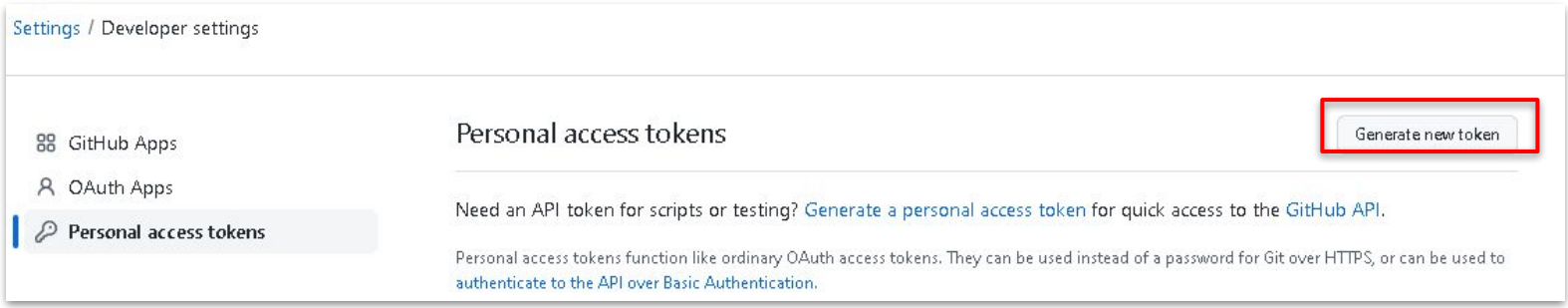


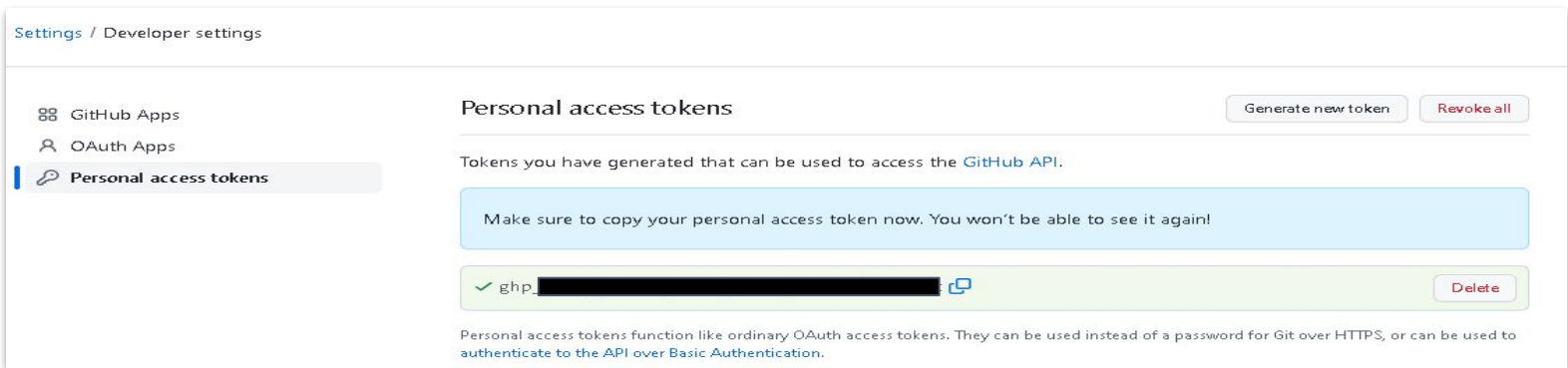
Mount a Remote Directory (cont.)

- Generate a personal access token









Mount a Remote Directory (cont.)

- Config

```
$ git config --global user.email " dukyun.nam@gmail.com"
$ git config --global user.name " Dukyun Nam"
$ git config credential.helper store
```

- Clone

```
# example: git clone https://github.com/dynam1/elec462.git
$ git clone https://github.com/ username/repo.git
Username: your_username
Password: your_token
```

- Commit & push

```
$ git add .
$ git commit -m “lab1”
$ git push
```