

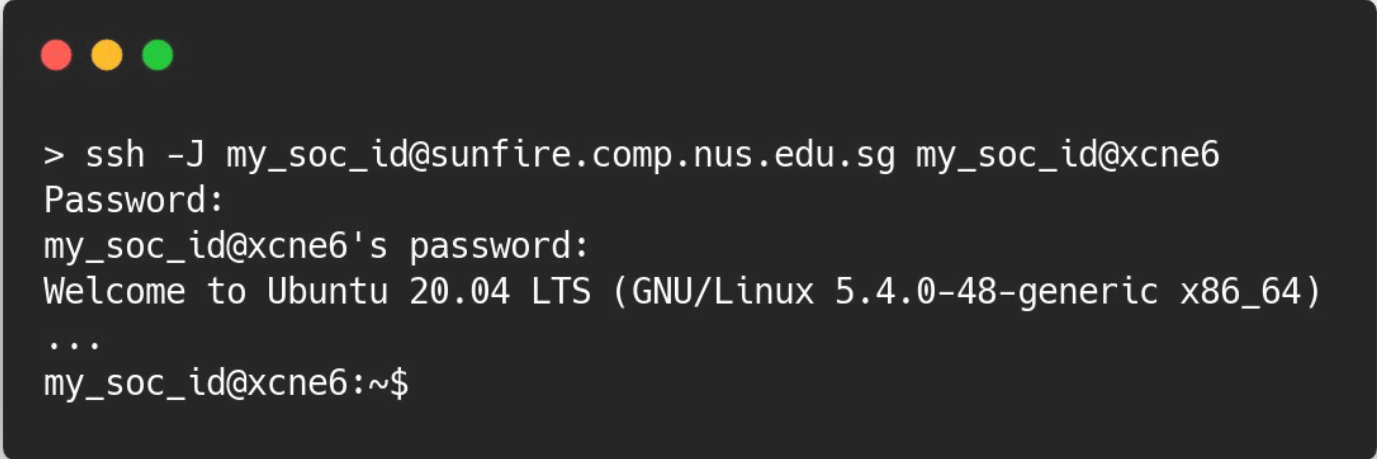
CS2106 Lab 1

Some slides to get your lab going!

What your lab development process should look like:

1. Code out your solutions using a Ubuntu VM or directly on the xcne servers.
2. Test your code while coding using the .in and .out files provided.
 - a. Please do this testing using your VM or the xcne servers!
 - b. It may seem like coding on e.g. a Mac may work as well now, but in later labs, there will be C and shell commands that **do not work** (or function differently) unless you're in a Linux environment as specified in the lab handout.
3. Check for memory issues using `valgrind`.
4. Do one final test on xcne servers (both test cases and `valgrind`).
5. Zip up your files, then run the submission checking shell script to make sure everything's alright.

Accessing the xcne servers



```
> ssh -J my_soc_id@sunfire.comp.nus.edu.sg my_soc_id@xcne6
Password:
my_soc_id@xcne6's password:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-48-generic x86_64)
...
my_soc_id@xcne6:~$
```

Using sunfire as a jump host with ProxyJump (-J flag)
(Recommended, no need SoC VPN)

You can also use the same -J flag for sftp when transferring files.

To avoid repeatedly typing this long command, add it into ~/.bash_aliases or refer to the alternative on the next slide.

Accessing the xcne servers (Alternative)

```
Host sunfire
  HostName sunfire.comp.nus.edu.sg
  User my_soc_id

Host xcne7
  HostName xcne7
  CheckHostIp no
  ProxyJump sunfire
  User my_soc_id

~
~
~
"~/ssh/config" 10L, 161C
```

Alternatively, add the ProxyJump into the `~/ssh/config` file
(create it if it doesn't exist)

*You can then use the alias `xcne7` for `ssh` and `sftp`.
E.g. `ssh xcne7` or `sftp xcne7`*

(Optional) Configuring your SSH keys

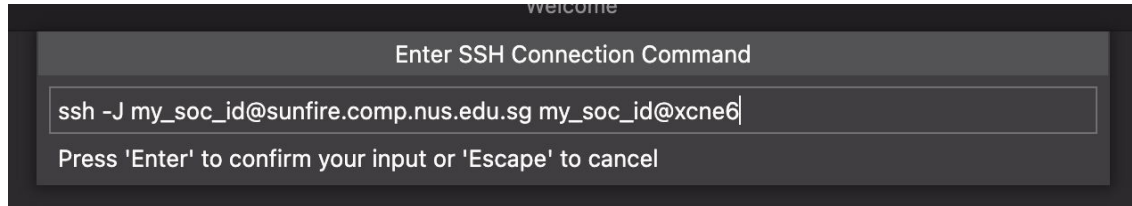
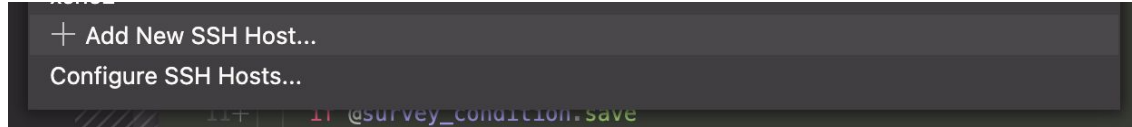
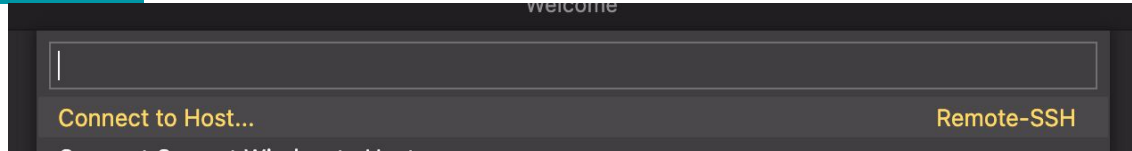
You can also set up SSH keys to avoid having to type your password twice.

GitHub has a pretty good guide on creating your first SSH key (which you can use for your GitHub account as well!): [Generating a new SSH key and adding it to the ssh-agent](#)

Once created, add your public key to the `~/.ssh/authorized_keys` file (create one if it doesn't exist) on both the sunfire server and the xcne server. You should now be able to directly SSH into xcne without typing any password! (except maybe for the password to your SSH key)

(Recommended) Using VS Code Remote Development

Developing on Remote Machines using SSH and Visual Studio Code The Visual Studio Code Remote



Exercise 2: Circular Linked List

Rationale: The linux kernel actually uses doubly circular linked lists extensively (why doubly or circular, and not just a regular singly linked list?)

Tips:

- Check out `ex2.c` (the provided runner) for some hints on how to use `malloc`, initialise structs, `free`, and how to iterate through the circular linked list.
- Consider coding out reusable helper functions to reduce code duplication. This will help with debugging!
- Remember to watch out for edge cases, e.g. when a deletion results in 0 nodes left, when you're inserting into index 0 (the new head), etc.

Exercise 3: Function Pointers

Rationale: Very interesting technique (though it might seem weird at first glance). Quite similar in concept to higher order functions in other languages!

Tips:

- Refer to `ex2.c` (the provided runner) for help on how to structure your runner.
- You will be reading from a file instead of `stdin`! Some commands that might be helpful are `fopen`, `fscanf` and `fclose`.
- You can print to `stderr` using `fprintf` if the file specified does not exist. You can also `exit` with a non-zero exit code when that happens, to signal an error has occurred.
- Remember to initialise everything related to `func_list` in `function_pointers.c`!

Exercise 5 & 6: Shell Scripts

Rationale: These two exercises aim to familiarise you with bash commands.

Clarification: User Processes in Exercise 5 refers to the number of processes that **the current user** (who is running the shell script) is running.

Tips:

- Exercise 6 is quite straightforward. Simply check out the man page of `strace`, i.e. run `man strace` in your shell to find out more about the command.
- For Exercise 5, a bit more googling and reading of man pages will be required. The later values also require quite a bit of piping to produce.