Missing Semester Lecture 5

**Job control**

1. From what we have seen, we can use some ps aux | grep commands to get our jobs’ pids and then kill them, but there are better ways to do it. Start a sleep 10000 job in a terminal, background it with Ctrl-Z and continue its execution with bg. Now use [pgrep](https://www.man7.org/linux/man-pages/man1/pgrep.1.html) to find its pid and [pkill](http://man7.org/linux/man-pages/man1/pgrep.1.html) to kill it without ever typing the pid itself. (Hint: use the -af flags).

Sleep 10000

<ctrl -z>

bg %1

pgrep -af sleep

pkill sleep

1. Say you don’t want to start a process until another completes, how you would go about it? In this exercise our limiting process will always be sleep 60 &. One way to achieve this is to use the [wait](https://www.man7.org/linux/man-pages/man1/wait.1p.html) command. Try launching the sleep command and having an ls wait until the background process finishes.

sleep 60 &

pgrep sleep | wait && ls

However, this strategy will fail if we start in a different bash session, since wait only works for child processes. One feature we did not discuss in the notes is that the kill command’s exit status will be zero on success and nonzero otherwise. kill -0 does not send a signal but will give a nonzero exit status if the process does not exist. Write a bash function called pidwait that takes a pid and waits until the given process completes. You should use sleep to avoid wasting CPU unnecessarily.

**Terminal multiplexer**

1. Follow this tmux [tutorial](https://www.hamvocke.com/blog/a-quick-and-easy-guide-to-tmux/) and then learn how to do some basic customizations following [these steps](https://www.hamvocke.com/blog/a-guide-to-customizing-your-tmux-conf/).

**Aliases**

1. Create an alias dc that resolves to cd for when you type it wrongly.
2. Run history | awk '{$1="";print substr($0,2)}' | sort | uniq -c | sort -n | tail -n 10 to get your top 10 most used commands and consider writing shorter aliases for them. Note: this works for Bash; if you’re using ZSH, use history 1 instead of just history.

**Dotfiles**

Let’s get you up to speed with dotfiles.

1. Create a folder for your dotfiles and set up version control.
2. Add a configuration for at least one program, e.g. your shell, with some customization (to start off, it can be something as simple as customizing your shell prompt by setting $PS1).
3. Set up a method to install your dotfiles quickly (and without manual effort) on a new machine. This can be as simple as a shell script that calls ln -s for each file, or you could use a [specialized utility](https://dotfiles.github.io/utilities/).
4. Test your installation script on a fresh virtual machine.
5. Migrate all of your current tool configurations to your dotfiles repository.
6. Publish your dotfiles on GitHub.

**Remote Machines**

Install a Linux virtual machine (or use an already existing one) for this exercise. If you are not familiar with virtual machines check out [this](https://hibbard.eu/install-ubuntu-virtual-box/) tutorial for installing one.

1. Go to ~/.ssh/ and check if you have a pair of SSH keys there. If not, generate them with ssh-keygen -o -a 100 -t ed25519. It is recommended that you use a password and use ssh-agent , more info [here](https://www.ssh.com/ssh/agent).
2. Edit .ssh/config to have an entry as follows

Host vm

User username\_goes\_here

HostName ip\_goes\_here

IdentityFile ~/.ssh/id\_ed25519

LocalForward 9999 localhost:8888

1. Use ssh-copy-id vm to copy your ssh key to the server.
2. Start a webserver in your VM by executing python -m http.server 8888. Access the VM webserver by navigating to http://localhost:9999 in your machine.
3. Edit your SSH server config by doing sudo vim /etc/ssh/sshd\_config and disable password authentication by editing the value of PasswordAuthentication. Disable root login by editing the value of PermitRootLogin. Restart the ssh service with sudo service sshd restart. Try sshing in again.
4. (Challenge) Install [mosh](https://mosh.org/) in the VM and establish a connection. Then disconnect the network adapter of the server/VM. Can mosh properly recover from it?
5. (Challenge) Look into what the -N and -f flags do in ssh and figure out what a command to achieve background port forwarding.