VE482
Introduction to Operating Systems

LAB 2

October 1, 2021

Yihua Liu 518021910998

---

1 Minix 3

- In Minix 3, how to manage software, i.e. install, remove, update, etc.? [6]
  - Install: `pkgin install foo`
  - Remove: `pkgin remove foo`
  - Update: `pkgin update`
  - Upgrade: `pkgin upgrade`
  - Upgrade all: `pkgin full-upgrade`
  - List available packages: `pkgin avail`
  - List installed packages: `pkgin list`
  - Remove orphan dependencies: `pkgin autoremove`
  - Delete downloaded packages from the cache directory: `pkgin clean`
- What is the purpose of the commands `ifconfig`, `adduser`, and `passwd`?
  - `ifconfig` - configure a TCP/IP device: initializes a TCP/IP setting the IP address and/or netmask. It will report the address and netmask set [1].
  - `adduser` - add a user to the system [2]: creating and populating a home directory if necessary. The arguments to adduser are the user name, the group name, and the user's home directory [4].
  - `passwd` - modify a user's password: changes the user's password [5].

2 Working on a remote server

- Setup an SSH server on Minix 3. From Linux (using ssh) or Windows (using Putty) log into Minix 3. Note: the network need to be properly setup on the Virtual Machine (VM).
  On Minix 3:

```
        pkgin update
        pkgin install openssh
        user add -m -g users yihua  # optional, "users" is
        ↪  the group name (the same as "other")
```

On WSL Ubuntu:

```
        sudo apt update
        sudo apt upgrade
        sudo apt install openssh-client
        ssh root@192.168.36.136  # default port is 22
```

On PowerShell Core 7:

```
        ssh root@192.168.36.136  # default port is 22
```

- What is the default SSH port? Change this port for port 2222. Log into Minix 3 using this new SSH server setup. The default SSH port is 22. To change this port for port 2222, on Minix 3:

```
        vi /usr/pkg/etc/ssh/sshd_config  # delete # and
        ↪  change the line to Port 2222
```

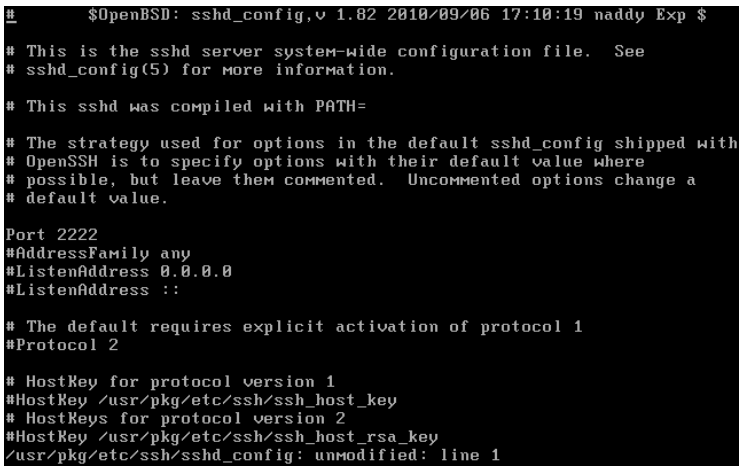

Figure 1. Screenshot of `sshd_config` on Minix 3.2.1.

On WSL Ubuntu or PowerShell Core 7:

```
1    ssh root@192.168.36.136 -p 2222
```

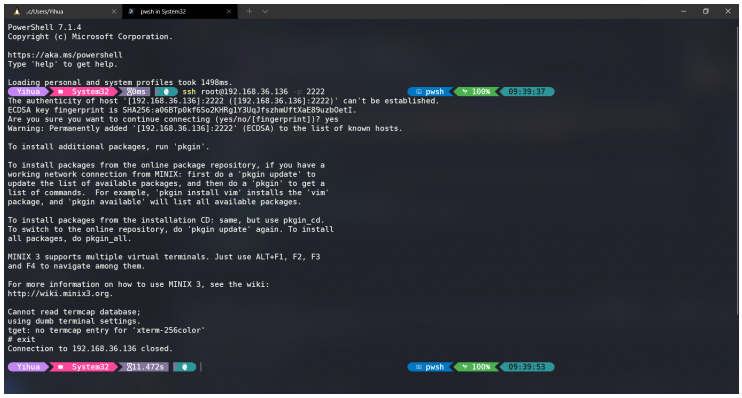

Figure 2. Screenshot of PowerShell Core 7 logging into Minix 3.

- List and explain the role of each the file in the `$HOME/.ssh` directory. In `$HOME/.ssh/config`, create an entry for Minix 3.
  - authorized_keys: a list of public SSH keys that is used to match users' private SSH keys to establish connections
  - config: SSH client configuration
  - id_rsa: private SSH key
  - id_rsa: public SSH key
  - known_hosts: a list of hosts that users have logged into

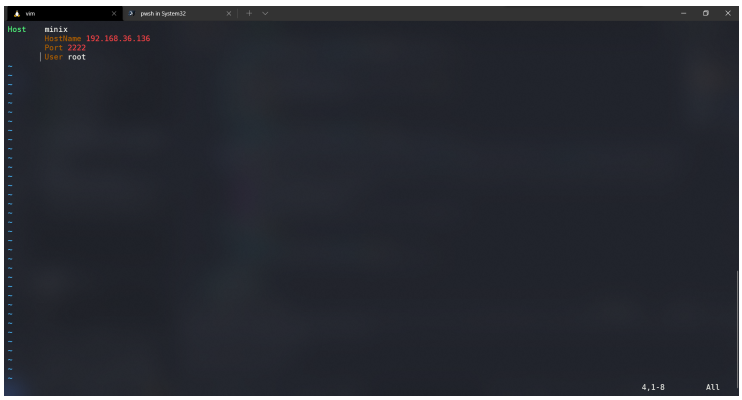

Figure 3. Screenshot of `/.ssh/config` creating an entry for Minix 3.

- Briefly explain how key-only authentication works in SSH. Generate a key-pair on the host system and use it to log into Minix 3 without a password. Key-only authentication works in SSH is based on asymmetric

cryptography. Given user-generated SSH key pairs (a private SSH key and a public SSH key) done by ssh-keygen, the public key will be sent to the server by ssh-copy-id. Then, the server will store the public key, marks it as authorized, and allows access to anyone who can prove they have the corresponds private key [7].
On WSL Ubuntu:

```
vim ~/.ssh/config  # add minix host
ssh minix
```

~/.ssh/config:

```
Host    minix
        HostName 192.168.130.138
        Port 2222
        User root
```

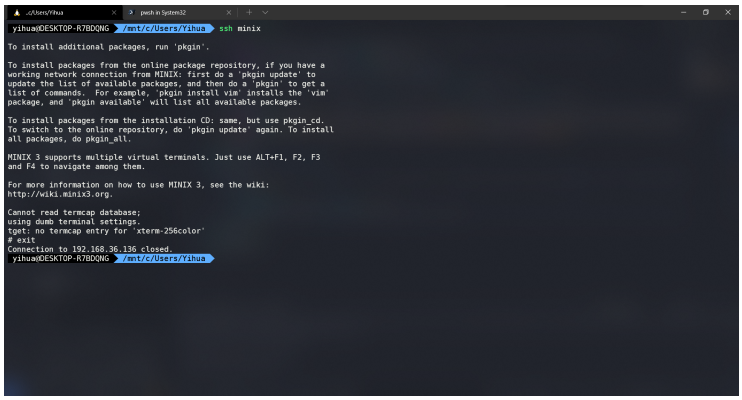

Figure 4. Screenshot of WSL Ubuntu logging into Minix 3.

- On Canvas, submit your public key in a *separate file*. Name it "student-id.pub", e.g. "5143709219.pub". This public key will be used to grant you access to the VE482 course server. Note: always remember that the private keys should remain *private*, and as such should never be disclosed.

3 Basic git

- Setup git on your computer, we will use it for the rest of the semester.
- Search the use of the following `git` commands [3]:
  - `help` - Display help information about Git.
  - `branch` - List, create, or delete branches.

  - `merge` - Join two or more development histories together: Incorporates changes from the named commits (since the time their histories diverged from the current branch) into the current branch.
  - `tag` - Create, list, delete or verify a tag object signed with GPG.
  - `commit` - Record changes to the repository: Create a new commit containing the current contents of the index and the given log message describing the changes.
  - `init` - Create an empty Git repository or reinitialize an existing one: This command creates an empty Git repository - basically a `.git` directory with subdirectories for `objects`, `refs/heads`, `refs/tags`, and template files.
  - `push` - Update remote refs along with associated objects: Updates remote refs using local refs, while sending objects necessary to complete the given refs.
  - `add` - Add file contents to the index: This command updates the index using the current content found in the working tree, to prepare the content staged for the next commit.
  - `log` - Show commit logs: Shows the commit logs. List commits that are reachable by following the `parent` links from the given commit(s), but exclude commits that are reachable from the one(s) given with a ^ in front of them.
  - `clone` - Clone a repository into a new directory: Clones a repository into a newly created directory, creates remote-tracking branches for each branch in the cloned repository (visible using `git branch --remotes`), and creates and checks out an initial branch that is forked from the cloned repository's currently active branch.
  - `checkout` - Switch branches or restore working tree files: Updates files in the working tree to match the version in the index or the specified tree.
  - `pull` - Fetch from and integrate with another repository or a local branch: Incorporates changes from a remote repository into the current branch.
  - `diff` - Show changes between commits, commit and working tree, etc: Show changes between the working tree and the index or a tree, changes between the index and a tree, changes between two trees, changes resulting from a merge, changes between two blob objects, or changes between two files on disk.
  - `fetch` - Download objects and refs from another repository: Fetch branches and/or tags (collectively, "refs") from one or more other repositories, along with the objects necessary to complete their histories.
  - `reset` - Reset current HEAD to the specified state.
- Setup your `git` repository on the VE482 server.

References

[1] Kees J. Bot. *IFCONFIG(8)*. 2013.

[2] Alistair G. Crooks. *USERADD(8) NetBSD System Manager's Manual*. Jan. 13, 2009.

[3] Git. *Git Reference*. 2021. URL: `https://git-scm.com/docs`.

[4] lionelsambuc. *Managing User Accounts*.

[5] Robert Morris and Ken Thompson. *PASSWD(1) NetBSD General Commands Manual*. Feb. 25, 2005.

[6] NetBSDfr. *pkgin, a binary package manager for pkgsrc*. NetBSD. URL: `https://pkgin.net/`.

[7] SSH.COM. *What is SSH Public Key authentication?* 2021.