

Installing Applications in FreeBSD

tsaimh(2022-2024, CC BY-SA)

lctseng (2019-2021, CC BY-SA)

? (1996-2018)

國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

Handbook and Manual pages

- Complete guide and be found at
 - <https://www.freebsd.org/doc/handbook/ports.html>
 - https://www.freebsd.org/doc/zh_TW/books/handbook/ports.html
 - [ports\(7\)](#)
 - [pkg\(7\)](#), [pkg\(8\)](#)

Before we start (1)

- Permission issue
 - **root**: the superuser
 - *In Unix-like system, root is the conventional name of the user who has all rights or permissions (to all files and programs) in all modes (single-user or multi-user)*
- Don't execute any command as root directly
 - It's **DANGEROUS**
- However sometimes you still need to be root to do something
 - Install software
 - Manage system settings
 - Create/modify/delete users

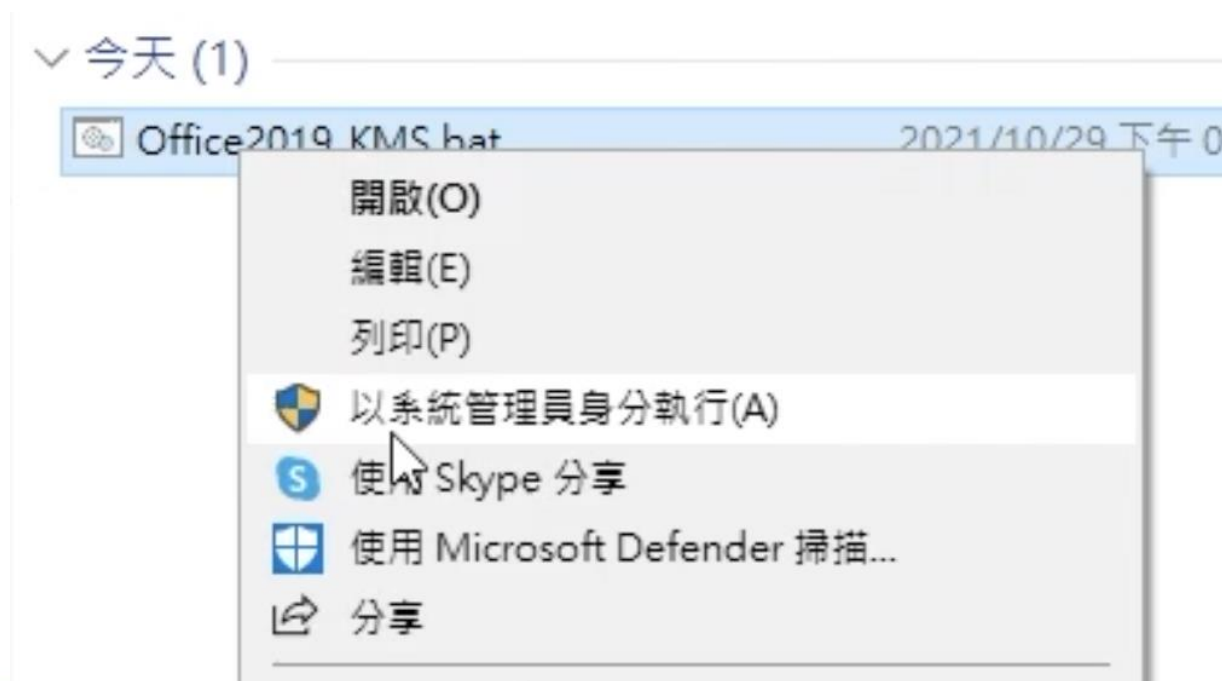
Before we start (2)

- Become root
 - Console login with root
 - By default, you cannot login as root via SSH
- Change current user
 - Don't need to login with console
 - Use command "su -", and then type root's password
 - Only user in "wheel" group can use "su -" to change to root
 - To see which account you are using, use "whoami"

```
$ whoami
tsaimh
$ su -
Password:
$ whoami
root
```

Before we start (3)

- As mentioned before, don't run as root directly
- Can we execute with root's credential only for some specific commands?
 - Like 'Run as administrator' in Windows
 - Is there similar commands in Unix-like system/FreeBSD?



Before we start (4)

- Run commands with other user's permission
- "sudo" command
 - Only simplest explanation here for basic usage
 - "sudo" syntax and other details will be explained in later chapters
 - Here only tell you how to simply enable 'sudo'
- How to enable sudo?
 - "sudo" is not a command in the base, needs to be installed manually

Enable "sudo" (1)

- Install the package
 - Check Internet connection
 - `$ ping -c4 8.8.8.8`
 - Become root
 - `$ su -`
 - Install the package of sudo
 - `$ pkg install sudo`
 - This will install 'sudo' from Internet
 - Type 'Y' (means yes) when it asks for confirmation

Enable "sudo" (2)

- Allowing your user to execute "sudo"
 - Switch to root first
 - If you are not familiar with the default editor 'vi', type the following command to change your editor for this time (skip this step otherwise)

```
$ setenv EDITOR ee
```

 - Will explain this in later chapters
 - This will allow you using a notepad-like editor

Before we start – Enable "sudo" (3)

- Allowing your user to execute "sudo"
 - Type "**visudo**" to edit the sudoers file (i.e., /usr/local/etc/sudoers)
 - Specify who can use "sudo"

```
##  
## User privilege specification  
##  
root ALL=(ALL) ALL  
tsaimh ALL=(ALL) ALL
```

- Save the file and exit, back to normal user
 - Use "logout" command or press Ctrl+D

Before we start – Using "sudo"

- Now, you can prepend "sudo" before commands to run them as root
 - But please **think carefully before you hit enter**
- Execute commands with "sudo"
 - sudo whoami
 - You have **root's credential**
 - sudo pkg install vim
 - Install software without becoming root directly
 - You need to re-type **your password**
 - Don't need to re-type within 5 minutes

```
% sudo -l
User tsaimh may run the
following commands on bsd4:
  (ALL : ALL) ALL
% whoami
tsaimh
% sudo whoami
Password:
root
tsaimh@bsd4:~ %
```

Install software: Overview (1)

- Package (Pre-built binary programs)
 - Like installers (.msi) in Windows
 - "package" (.txz) on FreeBSD
 - rpm on RedHat Linux, deb on Debian Linux
- Package Manager
 - install/remove/upgrade packages
 - Other Unix-like systems
 - rpm, yum, dpkg, apt, dnf, pacman ...
 - FreeBSD
 - `pkg`

Install software: Overview (2)

- Install from source
 - Managed source collection
 - FreeBSD Ports
 - With dependency checking and FreeBSD specified patches
 - Others
 - Download source tarball (.tar.gz) from websites
 - Checkout from VCS (git/svn)
 - No dependency checking

Install software: Comparison (1)

Method	Description	Dependency Checking
Packages	Pre-built ports, contains pre-compiled copies of all the commands for the software with default settings , as well as any configuration files or documentation.	Yes
Ports	A collection of files designed to automate the process of compiling an software from source code and additional patches (a set of Makefile, patches, description files, ...)	Yes
Tarball VCS	fetch it, configure the installation options, and compile it by yourself .	No

Install software: Comparison (2)

Method	Benefits
Packages	<ul style="list-style-type: none">● Packages do not require any additional compilation● Benefit for slow machines
Ports	<ul style="list-style-type: none">● Optimization<ul style="list-style-type: none">○ You can tweak the compilation options to generate code that is specific to a different processor● Customization<ul style="list-style-type: none">○ Some software have compilation time options relating to what they can and cannot do
Tarball VCS	<ul style="list-style-type: none">● Some software cannot be found in ports collection<ul style="list-style-type: none">○ Newly created projects, latest versions, ...● Some latest versions of software may have new configurations that do not exist in ports (cannot configure it through the ports easily)

Package System (1)

- pkg
 - New generation of FreeBSD package system
- Install new software
 - Fetch packages from a repository
 - Need root permission (sudo)
 - Automatically update the database
 - By default invoking either of `pkg install` or `pkg upgrade` will cause repository catalogues to be updated automatically
 - Perform dependency check
 - Will install software that required by new software

Package System (2)

- Install new software
 - `pkg install <names of packages...>`
 - `pkg install vim-console tmux`
- Upgrade currently installed software
 - `pkg upgrade <names of packages...>`
 - `pkg upgrade vim-console`
 - `pkg upgrade`
 - Upgrade all installed software
 - This will also update the database

Package System (3)

- Update packages database only
 - `pkg update`
- Delete a package
 - `pkg delete <names of packages>`
- Search
 - `pkg search <keyword>`
 - Search package repository catalogues

Package System (4)

- Show information about installed packages

- **pkg info**

- Show all installed packages
- Use "grep" to find specific packages
 - `pkg info | grep vim`

- **pkg info <name of package>**

- Show detailed information
- `pkg info vim-console`

```
% pkg info
...
sudo-1.9.15p5_4      Allow others to run
commands as root
tmux-3.3a_3         Terminal Multiplexer
vim-9.1.0470        Improved version of
the vi editor (console flavor)
% pkg info vim
vim-9.1.0470
Name      : vim
Version   : 9.1.0470
Installed on : Thu Sep  5 16:50:47 2024 CST
Origin    : editors/vim
Architecture : FreeBSD:14:amd64
Prefix    : /usr/local
Categories : editors
...
```

Package System (5)

- Show version of installed packages
 - **pkg version**
 - `pkg version -v`

```
$ pkg version -v
sudo-1.9.15p5_4      = up-to-date with port
tmux-3.3a_3         = up-to-date with port
vim-9.1.0470        < needs updating (port has 9.1.0708)
```

Port System

- We should...
 - Obtain the ports collection
 - List of ports available to be installed into system
 - Find the application
 - Change to the directory for the port
- Ports will
 - Fetch the source tarball
 - Ask for configuration friendly
 - Compile the source code to a **package**
 - Install the application via the just built package
- Deinstall process

Obtaining the Ports Collection (2/3)

- [git \(1\)](#)
 - Install git command line tool
 - `sudo pkg install git`
 - Checkout from a given repository
 - `sudo git clone https://git.FreeBSD.org/ports.git /usr/ports`
 - <https://docs.freebsd.org/en/books/handbook/ports/#ports-using>
 - **portsnap(8)** is deprecated since FreeBSD 14.0 Release, and now **git** is recommended.

Obtaining the Ports Collection (3/3)

- Port directory
 - `/usr/ports/<category>/<name>`

```
$ ls /usr/ports/  
CHANGES          archivers          emulators          math              shells  
CONTRIBUTING.md  astro             finance            misc              sysutils  
COPYRIGHT         audio             french             multimedia        textproc  
GIDs              benchmarks        ftp                net               ukrainian  
Keywords          biology           games              net-im            vietnamese  
MOVED             cad               german             net-mgmt          www  
...  
  
$ ls /usr/ports/editors/vim  
Makefile          distinfo          files              pkg-descr         pkg-plist
```

Ports system – Find your application

- `cd /usr/ports`
- For the first time, run "`sudo make fetchindex`" to fetch index for searching
- `make search name=program name`
- `make search key=string`


`/usr/ports/INDEX-14`

```
/usr/ports % sudo make fetchindex
Password:
/usr/bin/env fetch -am -o /usr/ports/INDEX-14.xz
https://download.FreeBSD.org/ports/index/INDEX-14.xz
/usr/ports/INDEX-14.xz                1826 kB  1105 kBps    01s
/usr/ports %
```

Ports system – Search through INDEX

```
% make search name=vim-9.1
Port: vim-9.1.0708
Path: /usr/ports/editors/vim
Info: Improved version of the vi editor (console flavor)
Maint: adamw@FreeBSD.org
B-deps: gettext-runtime-0.22.5 gettext-tools-0.22.5 indexinfo-0.3.1 libffi-
3.4.6 libtextstyle-0.22.5 mpdecimal-4.0.0 pkgconf-2.3.0,1 python311-3.11.9_1
readline-8.2.13
R-deps: gettext-runtime-0.22.5 indexinfo-0.3.1 libffi-3.4.6 mpdecimal-4.0.0
python311-3.11.9_1 readline-8.2.13 xxd-9.1.0708
WWW: https://www.vim.org/
% grep vim-9\..1 INDEX-14
vim-9.1.0708|/usr/ports/editors/vim|/usr/local|Improved version of the vi editor
(console flavor)|/usr/ports/editors/vim/pkg-descr|adamw@FreeBSD.org|editors|gettext-
runtime-0.22.5 gettext-tools-0.22.5 indexinfo-0.3.1 libffi-3.4.6 libtextstyle-0.22.5
mpdecimal-4.0.0 pkgconf-2.3.0,1 python311-3.11.9_1 readline-8.2.13|gettext-runtime-
0.22.5 indexinfo-0.3.1 libffi-3.4.6 mpdecimal-4.0.0 python311-3.11.9_1 readline-
8.2.13 xxd-9.1.0708|https://www.vim.org/| | |
```


Ports system - psearch

- [psearch\(1\)](#)
 - Simple but useful tool to find ports
 - ports-mgmt/psearch
 - or pkg install psearch
 - **psearch <name of port>**
 - psearch vim

```
% psearch vim
audio/vitunes      Curses-based media player with vim-like keybinds
devel/clewn        Clewn provides Gdb support within Vim
editors/cream      Gvim extension with many features
editors/vim        Improved version of the vi editor (console flavor)
...
```

Ports system – Compile and Install

- Type "make install clean" to install your application

- make config (/var/db/ports/)
- make fetch (/usr/ports/distfiles/)
- make checksum
- make extract
- make patch
- make configure
- make build
- make install
- make clean
 - Remove the expanded source code.
- make distclean
 - Clean downloaded distribution files (tarball)



make (all)

```
% make config  
===> Switching to root credentials  
to create /var/db/ports/shells_zsh  
Password:
```

Ports system (4)

- The ports system uses [fetch\(1\)](#) to download the files
 - **MASTER_SITES** environment variable
 - /etc/make.conf (can copy from /usr/share/examples/etc/make.conf)

```
MASTER_SITE_BACKUP?= \
    http://FreeBSD.cs.nctu.edu.tw/distfiles/${DIST_SUBDIR}/
MASTER_SITE_OVERRIDE?= ${MASTER_SITE_BACKUP}
```

- Options for ports
 - make config
 - Won't build or install the port
 - Use this to re-configure ports (otherwise, it uses old one instead)
 - hidden options (not shown in 'make config')
 - Edit the Makefiles under that port directory

Ports system (5)

- I have installed the application but **Command not found...**
 - Logout, and then login.
 - If you use (t)osh or zsh
 - rehash

Upgrading Ports using Portmaster

- ports-mgmt/portmaster
 - A utility for easily upgrading and installing ports

```
$ cd /usr/ports/ports-mgmt/portmaster && make install clean
```

- Install or upgrade a port
 - `portmaster <category>/<name>`
 - `portmaster sysutils/lsof`
 - `/usr/ports/UPDATING`
 - **Read before attempting any port upgrades!!!**
- Useful options
 - `-B, -d, -a, -r, -y, -H, -w`
 - `portmaster -dyBwH editors/vim`
 - `/usr/local/etc/portmaster.rc`

Security

- Show security issues about installed packages
 - No matter from port or from package
 - **pkg audit**
 - Upgrade these packages to mitigate security problems

```
$ pkg audit -F
Fetching vuln.xml.xz: 100%    1 MiB    1.1MB/s    00:01
0 problem(s) in 0 installed package(s) found.
```

```
$ pkg audit
python38-3.8.10 is vulnerable:
  Python -- multiple vulnerabilities
  WWW: https://vuxml.FreeBSD.org/freebsd/145ce848-1165-11ec-ac7e-08002789875b.html
```

Install from source (1)

- Compile the source files first and then install
 - Tarball, a pack of source code
 - `tar -xzf certain-source.tar.gz`
 - `cd certain-source`
 - `./configure [options ...]`
 - `./configure --help`
 - `make`
 - `make install` (root permission)

Install from source (2)

- Compile the source files first and then install
 - Checkout master branch from VCS
 - `git clone --depth=1 https://github.com/curl/curl.git`
 - `cd curl`
 - `./buildconf`
 - `./configure --enable-debug`
 - `make`
 - `sudo make install`

Security Considerations (1)

- How to find secure source
 - Check the official site, read the announcement and change log
 - Verify the checksum (tarball)
 - Fetch via https or ssh (VCS)

Security Considerations (2)

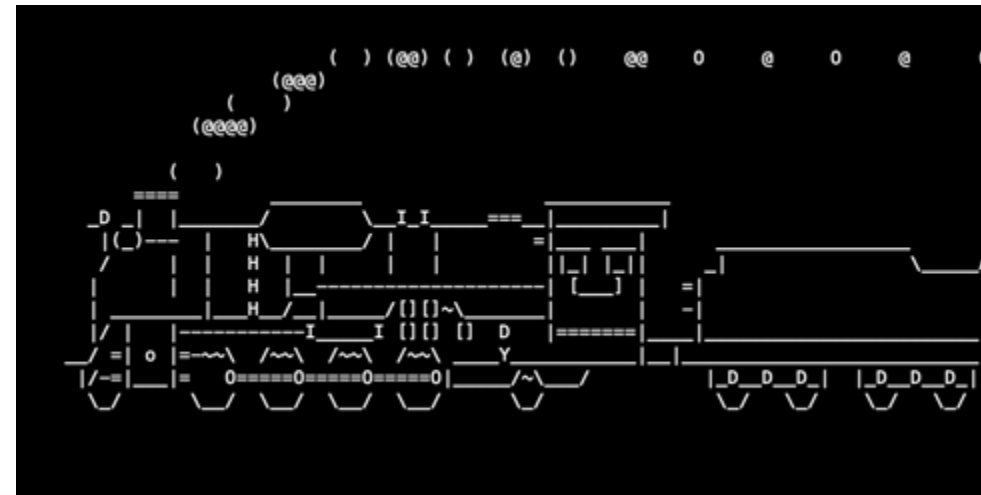
- Why "curl *URL* | sh" is bad?
 - Example: "curl get.pow.cx | sh"
 - Why do you think this is good?
 - Search: "curl pipe bash"
 - Even the file does not contain evil code, broken connection may turn it.
 - "rm -rf /tmp/foo.bar" becomes "rm -rf /"
 - Instead: download the script, read it, execute it.

Deinstall Applications

- Two methods
 - pkg delete
 - Find the package name via pkg info
 - Dependency check
 - Disable dependency check
 - -f : force
 - `pkg delete -f <names of packages>`
 - **make deinstall**
 - Change to the port's directory
 - make deinstall
 - Delete it anyway
 - Similar to "pkg delete -f"

Try to install from ports/pkg

- tmux
- vim-console, emacs
- mutt
- wget, curl
- lftp
- lynx, w3m
- expect
- zsh, bash
- ~~st~~



Appendix

Package management in other Unix-like systems

國立陽明交通大學資工系資訊中心

Information Technology Center, Department of Computer Science, NYCU

Package Manager Rosetta Stone

- Package commands in the most common systems
- <https://wiki.freebsd.org/PackageManagerRosettaStone>

Search package	<code>yum search <u>pattern</u></code>	<code>pkg search <u>pattern</u> (only by name)</code>
Install package	<code>yum install <u>packagename</u></code>	<code>pkg install <u>packagename</u></code>
Delete single package	<code>yum remove <u>packagename</u></code>	<code>pkg delete -f <u>packagename</u></code>
Delete package and dependencies	<code>yum autoremove <u>packagename</u></code>	<code>pkg delete <u>packagename</u></code>
List installed packages	<code>rpm -qa</code> <code>yum info</code>	<code>pkg info</code>
List files installed by a package	<code>rpm -ql <u>packagename</u></code>	<code>pkg info -l <u>packagename</u></code>
Upgrade single package with dependencies	<code>yum upgrade <u>packagename</u></code> <code>yum upgrade-to <u>versionedpackagename</u></code>	<code>pkg upgrade <u>packagename</u></code>
Upgrade all packages	<code>yum update</code> <i>(also see yum(8) for 'yum upgrade')</i>	<code>pkg upgrade</code>