OpenBSD

- Openbsd-as-a-Desktop-OS
- Install Pandoc via Cabal OpenBSD 6.7
- OpenBSD virtualization OpenBSD 6.7

Openbsd-as-a-Desktop-OS

Upgrade OpenBSD

Patch Base System

```
syspatch
```

Upgrade system

```
sysupgrade
```

Upgrade Packages

```
pkg_add -u
```

Package Management

Search for a package

```
pkg_info -Q <searchterm>
```

Get Info about package

```
pkg_info <packagename>
```

install package

```
pkg_add <packagename>
```

System Enhancement

If you have an SSD you can edit `/etc/fstab` and add attribute `softdep`

Download and Configure Ports tree

```
ftp https://cdn.openbsd.org/pub/OpenBSD/$(uname -r)/{ports.tar.gz,SHA256.sig}
signify -Cp /etc/signify/openbsd-$(uname -r | cut -c 1,3)-base.pub -x SHA256.sig ports.tar.gz
tar xzvf /tmp/ports.tar.gz -C /usr
```

Install Pandoc via Cabal OpenBSD 6.7

Install Cabal

Cabal is a packagemanager for Haskell very much like pip is for Python. You can install it from ports:

```
doas pkg_add -i ghc cabal-install
```

Prepare system to compile programs via cabal

Enable wxallowed in /home and /tmp:

You can do so via editing your /etc/fstab like this (this is my personal fstab. don't blindly copy paste, you have to insert the wxallowed in the right place in your own /etc/fstab.

```
.b none swap sw
.a / ffs rw,softdep,noatime 1 1
.k /home ffs rw,softdep,wxallowed,noatime,nodev,nosuid 1 2
.d /tmp ffs rw,softdep,wxallowed,noatime,nodev,nosuid 1 2
.f /usr ffs rw,softdep,wxallowed,noatime,nodev 1 2
.g /usr/X11R6 ffs rw,softdep,noatime,nodev 1 2
.h /usr/local ffs rw,softdep,wxallowed,noatime,nodev 1 2
.j /usr/obj ffs rw,softdep,noatime,wxallowed,nodev,nosuid 1 2
.i /usr/src ffs rw,softdep,noatime,nodev,nosuid 1 2
.e /var ffs rw,softdep,noatime,nodev,nosuid 1 2
```

then reboot.

(optional) enable multithreating

If you want to enable hyperthreating tempoary, do:

```
doas sysctl hw.smt=1
```

if you want to make it permanent edit your /etc/sysctl.conf and paste the following at the end:

```
hw.smt=1
```

notice: without rebooting you have once to enable it manually, after the fist reboot, it will be active

```
sysctl hw.smt=1
```

you can check that hyperthreating is active, with programs like top or htop

raise ulimit:

put the following into your

```
~/.profile
```

```
ulimit -d 4096*1024
```

If you haven't relogged, you have to manually paste this everytime you open a terminal in which you want to compile, since it's a command for your shell.

```
ulimit -d 4096*1024
```

Correct you PATH

edit this into you `.profile`

```
PATH=$HOME/bin:/bin:/sbin:/usr/bin:/usr/sbin:/usr/X11R6/bin:/usr/local/bin:/usr/local/sbin:/usr/games:$HOME/.cabal/bin/:.
```

install pandoc or pandoc-citeproc

if you want to convert LATEX you probably want pandoc-citeproc, it automatically pulls pandoc as dependency:

as a user (not root) execute:

```
ulimit -d 4096*1024
cabal update
cabal install pandoc-citeproc
```

This will need a lot of time and resources to compile, be pacient. Pandoc should now be able to be compiled.

After compilation is finished, don't be surpised if nothing happens when you execute pandoc and / or getting an error, read clearly, you may have the binary successfully compiled and it just couldn't symlink. you probably can find pandoc in:

~/.cabal/bin/pandoc

if nothig happens when you execute this binary, it probably does work as inteded anyways, try to give it an argument like:

```
~/.cabal/bin/pandoc -v
```

Congratulations! You have successfully compiled and installed pandoc in OpenBSD

OpenBSD virtualization

OpenBSD 6.7

Setup

`/etc/rc.conf.local`

```
apmd_flags="-A"
dhcpd_flags=vether0
vmd_flags=
ntpd_flags="-s"
```

/etc/hostname.vether0

```
inet 192.168.30.1 255.255.255.0 NONE
```

/etc/dhcpd.conf

```
# Network:        192.168.11.0/255.255.255.0
# Domain name:    vmm.local
# Name servers:   192.168.11.1
# Default router: 192.168.11.1
# Addresses:      192.168.30.100 - 192.168.30.200

shared-network VMM-LOCAL {
    subnet 192.168.30.0 netmask 255.255.255.0 {
        range 192.168.30.100 192.168.30.200;

        option subnet-mask 255.255.255.0;
```

```
        option broadcast-address 192.168.30.255;
        option routers 192.168.30.1;
        option domain-name-servers 192.168.11.1;

#        host vm1 {
#            hardware ethernet 00:20:91:00:00:01;
#            fixed-address vm1.vmm.local;
#        }
    }
}
```

/etc/sysctl.conf

```
net.inet.ip.forwarding=1
```

/etc/pf.conf

```
set skip on lo

block return    # block stateless traffic
pass            # establish keep-state

# By default, do not permit remote connections to X11
block return in on ! lo0 proto tcp to port 6000:6010

ext_if="em0"
int_if="{ vether0 tap0 }"
set block-policy drop
set loginterface egress
match in all scrub (no-df random-id max-mss 1440)
match out on egress inet from !(egress:network) to any nat-to (egress:0)
pass out quick inet
pass in on $int_if inet
pass in on egress inet proto tcp from any to (egress) port 22
```

/etc/vm.conf

```
switch "local" {

   add vether0
   add tap0

}

vm "vm1.vm" {
    memory 512M
    kernel "/bsd.rd"
    disk "/vmm/vm1.img"
    interface {
        switch "local"
        lladdr 00:20:91:00:00:01
    }
}
```

Commands

```
vmmctl status
```

```
vmctl console 1
cu /dev/ttyp0
```

```
vmctl create /vmm/vm1.img -s 500M
vmctl start -c -b /bsd.rd -m 512M -i 1 -d /vmm/vm1.img
```

```
# X11 Forwarding
ssh -Y vm programmname
```