Quick FreeBSD Tricks

In several places, I have simply inserted the code as a script to be downloaded. I tend to keep things like this around so I can just run a simple command instead of copying/pasting an entire line.

Hardware Devices

To locate all attached drives, scan dmesg.boot

findDrives.sh

```
#! /bin/sh

egrep '(ad|cd|da)[0-9]' /var/run/dmesg.boot | sort
```

Or, simply run

```
geom disk list
```

List USB devices

```
usbconfig
```

Format a Drive

Following example will set up an empty drive (/dev/da23) with GPT as the schema and one slice (partition). The partition will have a label, *sneakernet*, then be formatted using ufs.

When you label a slice, it can be found in /dev/gpt/ and mounted using that alias.

```
# create the GPT schema
gpart create -s GPT da23
# show what is there (nothing yet)
gpart list /dev/da23
# add a new slice of type UFS, and give it a label, sneakernet
# use all of the available disk space for this (ie, only one slice)
gpart add -t freebsd-ufs -l sneakernet da23
# show what is up
gpart list /dev/da23
# format the partition with UFS
newfs /dev/da23p1
# it should now show up as an alias in /dev/gpt
```

```
ls /dev/gpt
# make a place to mount it
mkdir /mnt/sneakernet
# mount it
mount /dev/gpt/sneakernet /mnt/sneakernet
# unmount it
umount /mnt/sneakernet
# clean up
rmdir /mnt/sneakernet
```

Monitoring

CPU Temperatures

Following command will show you the temperature of each core of a processor

```
sysctl -a | grep temperature | grep cpu
```

iotop substitute

iotop is a well known utility under Linux, but not available for FreeBSD. However, the following command will do the same thing (does not apparently work for iSCSI devices)

```
top -m io -o total
```

watch substitute

Under Linux, watch repeats a command over and over, so it is useful for monitoring long running processes. The FreeBSD command *cmdwatch* does the same thing, with the same flags.

```
cmdwatch zpool iostat -v
```

Package Management

setting pkg to not ask permission

I was used to Debian's apt-get, and used the -y (answer “Yes” to all questions) parameter. Looking for something similar for pkg, I ran across http://dan.langille.org/2013/12/06/bootstrapping-installing-pkg-on-freebsd-unattended-and-without-answering-yes/ which showed a possible answer; set an environmental variable as part of the call.

```
env ASSUME_ALWAYS_YES=YES pkg install p5-libwww
```

will install LWP (p5-libwww) without waiting for you to select “Yes”

clean pkg cache

After a while, your pkg cache will use more and more space on your disk, with copies of packages you have already installed. The following command cleans that cache.

```
pkg clean
```

User Administration

Administrative Permission

By default, a new user is **not** able to become root. To do this, you must add them to the *wheel* group. Use the following command

```
pw user  mod  username -G wheel
```

where *username* is the username of the user who has access

Changing Shell

I just like *bash* for my shell. While it is not the standard for BSD, it is much more powerful than the standard *sh*, so I like to use it for my personal account. Once bash is installed, set it as the default shell for a user.

```
chsh -s bash username
```

Warning: do not modify the root account's shell. You will break your system. If you want a bash script to run as root, be sure to include

```
#! /usr/bin/env bash
```

at the head of your scripts.

zfs tricks

- zfs unshare -a
- zfs share -a
- showmount -e

From:
https://wiki.linuxservertech.com/ - **Unix Server Tech Knowledge Base**

Permanent link:
https://wiki.linuxservertech.com/unix/freebsd/quick_tips

Last update: **2025/11/24 03:45**