


# FreeBSD

[a38x](#), [software](#), [freebsd](#)

## Overview

<b>Name:</b>	FreeBSD
<b>Picture:</b>	
<b>Active</b>	Yes
<b>Purpose</b>	General Linux
<b>Ignition install</b>	No
<b>Compatibility</b>	All A38x MicroSoms
<b>More information</b>	
<b>Maintained by</b>	FreeBSD team
<b>Forum</b>	
<b>Website</b>	<a href="http://freebsd.org">freebsd.org</a>

## Description

FreeBSD is an operating system for a variety of platforms which focuses on features, speed, and stability. It is derived from BSD, the version of UNIX® developed at the University of California, Berkeley. It is developed and maintained by a large community.

## Prepare your uSD card

### Configure partitions of uSD card

Adjust the card to the Clearfog boot from SD requirements - let's assume the card is present in the FreeBSD host system as `/dev/mmcsd0`:

```
# dd if=/dev/zero of=/dev/mmcsd0
# gpart create -s MBR mmcsd0
# gpart add -t \!12 -b 4096 -s 32m mmcsd0
# gpart set -a active -i 1 mmcsd0
# gpart add -t \!12 -s 32m mmcsd0
# gpart set -a active -i 2 mmcsd0
# gpart add -t freebsd -s 2048m mmcsd0
# gpart create -s BSD mmcsd0s3
# gpart add -t freebsd-ufs -a 2m mmcsd0s3
```

```
# newfs_msdos /dev/mmcsd0s2  
# newfs /dev/mmcsd0s3a
```

## Mount uSD card

```
# mkdir /mnt/boot /mnt/rootfs  
  
# mount -t msdosfs /dev/mmcsd0s2 /mnt/boot  
# mount /dev/mmcsd0s3a /mnt/rootfs
```

## Install FreeBSD on uSD card

This part must be done on the FreeBSD host (via virtual machine or natively). It's impossible to build it directly on other OS.

## Download sources

```
$ pkg install git  
$ git clone https://github.com/freebsd/freebsd.git  
$ cd freebsd
```

## Export variables for cross-compilation

Some variables (shareown, makeobjdirprefix, etc) should be adjusted to your needs.

User in SHAREOWN variable must exist.

```
$ setenv MAKEOBJDIRPREFIX ~/obj  
$ setenv ARCH arm  
$ setenv TARGET_ARCH armv6  
  
$ setenv DESTDIR /mnt/rootfs  
$ setenv SHAREOWN user  
$ setenv SHAREGRP wheel  
$ setenv INSTALL_OPTS "-DWITHOUT_SHAREDOCS -DWITHOUT_EXAMPLES -  
DWITHOUT_GAMES -DWITHOUT_HTML -DWITHOUT_INFO -DWITHOUT_MAN"
```

## Build and install kernel

Use -j option to set cores/threads. In my case 4 threads.

```
$ make -j4 kernel-toolchain  
$ make -j4 buildkernel KERNCONF=ARMADA38X  
$ make installkernel KERNCONF=ARMADA38X
```

## Build and install world

```
$ make builddtb FDT_DTS_FILE=armada-388-clearfog.dts
$ make -j4 buildworld -DWITH_FDT
$ make installworld $INSTALL_OPTS
$ make distrib-dirs
$ make distribution
```

At this point you have compiled and installed FreeBSD with the kernel at `/mnt/rootfs/boot/kernel/` and dtb's in `/mnt/rootfs/boot/dtb/` directories of the mounted storage.

## Copy loader to boot partition

```
# cp /mnt/rootfs/boot/ubldr.bin /mnt/boot/
```

## Configure fstab

```
# echo "/dev/mmc0s3a / ufs rw 1 1" >> /mnt/rootfs/etc/fstab
```

## Environment variables for loader + tuning the kernel

In file `/boot/loader.conf` you can set kernel variables, which are passed further by loader.

Execute lines below to turn on L1 cache prefetch in FreeBSD kernel.

```
# echo "hw.cpu.quirks.actlr_set = 0x4" >> /mnt/rootfs/boot/loader.conf
# echo "hw.cpu.quirks.actlr_mask = 0x4" >> /mnt/rootfs/boot/loader.conf
```

Later once the kernel is running on the target machine, above variables values can be verified, using `sysctl`.

```
# sysctl -a | grep hw.cpu
```

## Unmount uSD card

```
# umount /mnt/boot /mnt/rootfs
```

Following part of tutorial will cover all things necessary to build u-boot

## U-Boot

Remember switch to linux machine.

## Get toolchain and set env variables for cross compilation

```
$ cd
$ wget
https://releases.linaro.org/archive/14.11/components/toolchain/binaries/arm-
linux-gnueabi/gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.xz
```

```
$ tar xvf gcc-linaro-4.9-2014.11-x86_64_arm-linux-gnueabi.tar.xz
$ export CROSS_COMPILE="$HOME/gcc-linaro-4.9-2014.11-x86_64_arm-linux-
gnueabi/bin/arm-linux-gnueabi-"
```

```
$ export ARCH=arm
```

## Download and build u-boot

Assuming, that the uSD card is present in the system as /dev/mmcblk0:

```
$ git clone https://github.com/SolidRun/u-boot-armada38x.git
$ cd u-boot-armada38x
$ make armada_38x_clearfog_config
$ make u-boot.mmc
# dd if=u-boot.mmc of=/dev/mmcblk0 bs=512 seek=1
```

Plug SD card to board and turn the Clearfog on.

## U-boot configuration

Open board serial console, reset SoC and wait for:

Hit any key to stop autoboot:

Stop In U-Boot by pressing any key. Then set proper booting command in bootcmd variable:

```
setenv fdtfile armada-388-clearfog.dtb
setenv bootcmd 'fatload mmc 0:2 900000 ubldr.bin; go 900000;'
setenv bootdelay 5 #or smaller
saveenv
```

Now you could restart board and wait for system:

```
reset
```

After less then minute you could login to FreeBSD:

```
FreeBSD/arm (Amnesiac) (ttyu0)
login:
```

## Finalizing

### Set root password

After login to FreeBSD (root with no password) you can set password by typing:

```
passwd
```

Then you have to type new password 2 times.

### Create ordinary user

To create new user use:

```
adduser
```

It works in interactive mode, so it will ask you few questions and suggest default values

### Grow rootfs partition

If size of your SD card is larger than size of partition you have created you can type following command:

```
service growfs onestart
```

It will resize rootfs to maximum size.

## Official SolidRun Images

The latest images can be found here:

<http://images.solid-build.xyz/A38X/freebsd/>

## Known Issues

Switching between Clearfog Base and Clearfog Pro need some software changes (Uboot etc.)

## External Links

- [FreeBSD](#)

From:

<https://wiki.solid-run.com/> - **Wiki | SolidRun**

Permanent link:

<https://wiki.solid-run.com/doku.php?id=products:a38x:software:os:freebsd>

Last update: **2017/07/25 09:29**

