

?????? ??

FreeBSD 安装 配置 网络 数据库 系统 工具 软件 .

- [PostgreSQL 安装](#)
- [MySQL 安装 配置 DB 工具 , 备份 恢复](#)
- [FreeBSD 安装 apcupsd 配置](#)
- [Certbot : 安装 配置 使用](#)
- [Munge 安装](#)
- [Shell 安装 sh 配置 bash 使用](#)
- [alias 安装 bash shell 配置 使用 \(.bashrc 配置 .bash_profile\)](#)
- [exFAT 安装 使用](#)
- [ssh 安装 使用](#)
- [GitHub Pull 配置 merge 使用](#)
- [tar 安装](#)

PostgreSQL ????

[FreeBSD Wiki](#) 資料庫 .

2024 01 14 日 FreeBSD PostgreSQL 16 安裝 .

登入 root 帳號 . \$ 登入 root 帳號 . 登入 **sudo** 登入 .

????

1. 安裝 PostgreSQL 16 .

```
$ pkg install postgresql16-server postgresql16-client
```

2. 設定 PostgreSQL 16 的配置文件 .

```
$ pkg search postgresql
```

3. 設定 PostgreSQL 16 的配置文件 .

```
$ sysrc postgresql_enable="YES"
```

4. 設定 PostgreSQL 16 的配置文件 .

```
$ /usr/local/etc/rc.d/postgresql initdb
```

```
$ service postgresql start
```

5. 設定 PostgreSQL 16 的配置文件 .

```
$ sockstat -46 | grep 5432
```

<IPv4 地址 端口>

```
$ sockstat -4 | grep 5432 -> IPv4 地址 端口.
```

```
$ sockstat -6 | grep 5432 -> IPv6 地址 端口.
```

6. 設定 PostgreSQL 16 的配置文件 .

```
$ sudo passwd postgres
```

Enter new password:
Enter it again:
passwd: password updated successfully
You can now log in
as postgres using `su - postgres` or by specifying the name of the user with the `-l` option, i.e. `su -l postgres`
You can suppress this message by setting `passwd_update` to `no` in `/etc/passwd`.

```
# su - postgres (or sudo su - postgres if you have sudo installed)
$ createuser admin
$ createdb foo_db -0 admin
```

Now you can log in as the postgres user (psql is the preferred way) and create a database named foo_db.
To log in as postgres, run **exit** to return to the shell, then run **psql foo_db** to log in.

```
$ psql foo_db
foo_db=# alter role admin with encrypted password 'yourpassword';
foo_db=# grant all privileges on database foo_db to admin;
foo_db=# exit
$ exit
```

Now you can log in as the postgres user and create a database named foo_db.
To log in as postgres, run **exit** to return to the shell, then run **psql foo_db** to log in.

```
$ dropdb foo_db
```

Now you can log in as the postgres user and create a database named foo_db.
To log in as postgres, run **exit** to return to the shell, then run **psql foo_db** to log in.

????

Bind ?? ??

Now you can log in as the postgres user and create a database named foo_db.
To log in as postgres, run **exit** to return to the shell, then run **psql foo_db** to log in.
data16 is the name of the database.
The `listen_addresses` parameter in `/var/db/postgres/data16/postgresql.conf` controls which IP addresses the server listens to.
By default, it is set to `'*'`, which means it listens to all IP addresses.
You can change it to a specific IP address, or to a range of IP addresses.
For example, to listen to all IP addresses, you can set `listen_addresses = '*'`.
To listen to a specific IP address, you can set `listen_addresses = '127.0.0.1'`.
To listen to a range of IP addresses, you can set `listen_addresses = '127.0.0.1/24'`.

```
sudo nano /var/db/postgres/data16/postgresql.conf
```

Now you can log in as the postgres user and create a database named foo_db.
To log in as postgres, run **exit** to return to the shell, then run **psql foo_db** to log in.
data16 is the name of the database.
The `listen_addresses` parameter in `/var/db/postgres/data16/postgresql.conf` controls which IP addresses the server listens to.
By default, it is set to `'*'`, which means it listens to all IP addresses.
You can change it to a specific IP address, or to a range of IP addresses.
For example, to listen to all IP addresses, you can set `listen_addresses = '*'`.
To listen to a specific IP address, you can set `listen_addresses = '127.0.0.1'`.
To listen to a range of IP addresses, you can set `listen_addresses = '127.0.0.1/24'`.

```
listen_addresses = '*'
```




```

def md5(filename):
    """Return the MD5 hash of the file at filename.

    The MD5 hash is a 32-character hexadecimal string.

    """
    # Open the file in binary mode
    with open(filename, 'rb') as f:
        # Read the file in chunks
        data = f.read()
        # Calculate the MD5 hash
        md5_hash = hashlib.md5(data).hexdigest()
        # Return the hash
        return md5_hash

```

1. 数据库用户，数据库名 admin，foo_db 数据库。
 2. 数据库用户 0.0.0.0/0 (所有 IP 地址)。
 3. 数据库用户 IP 地址。

,



md5
trust









(4□) □ □□□ □□ . □□ □□ □□□□ □□ □□ □□□□ .

MySQL ??? DB??, ???

????

mysql > create database DB;

mysql > use DB;

??? ??

mysql > create user 'root'@'localhost' identified by 'root';
mysql >

```
create user rootID;
```

```
create user rootID@'localhost' identified by 'root';
```

```
create user 'rootID'@'localhost' identified by 'PASSWORD';
```

mysql > create user 'root'@'localhost' identified by 'root';
mysql >

?????? ??

mysql > create database DB;

```
mysql > create database DB; default character set utf8;
```

????

mysql > create user 'root'@'localhost' identified by 'PASSWORD';
mysql >

```
mysql> GRANT ALL privileges ON DB[*].* TO 'UserId'@'localhost' IDENTIFIED BY 'PASSWORD';
```

```
mysql > flush privileges;
```

```

[ ] [ ] [ ] [ ] , identified by 'PASSWORD' [ ] [ ] [ ] localhost[ ] [ ] [ ] [ ]
[ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] [ ] .

```

FreeBSD?? apcupsd ??

apcupsd 是 APC UPS 的 软件 , 通过 UPS 的 网络 接口 实现 对 UPS 的 监控 和 管理 。

??

APC UPS 的 AP9631 Network Management Card 是 通过 网络 接口 实现 对 UPS 的 监控 和 管理 的 。

通过 网络 接口 实现 对 UPS 的 监控 和 管理 的 软件 是 apcupsd 。

??

在 FreeBSD 中 安装 apcupsd 。

```
sudo pkg update
sudo pkg install apcupsd
```

安装 完成后 , 需要 编辑 配置文件 **/usr/local/etc/apcupsd/apcupsd.conf** 。

```
sudo nano /usr/local/etc/apcupsd/apcupsd.conf
```

编辑 完成后 , 保存 并 退出 。

```
# 配置 文件 的 名称 。
UPSDNAME APCUPS

# SNMP 的 名称 和 地址 。
UPSCABLE ether

# SNMP 的 社区 字符串 。
UPSTYPE snmp

# UPS 的 网络 接口 IP 地址 和 端口 。 默认 是 161 端口 和 apc, 社区 字符串 是 public
DEVICE 192.168.0.255:161:apc:public

# 是否 启用 网络 服务器 。
NETSERVER on
```



```
# 0.0.0.0 [] [] [] localhost[] [][].  
NISIP 127.0.0.1  
# NIS [] [] [] 3551[].  
NISPORT 3551  
# UPS[] [] [] [] [] [] [] [].  
UPSCCLASS shareslave  
# [] [] [].  
UPSMODE disable
```

[] [] [] [] [] [] /etc/rc.conf[] [] .

```
sudo service apcupsd enable  
sudo service apcupsd start  
or  
sudo sysrc apcupsd_enable="YES"  
sudo shutdown -r now  
# []...
```

[] [] [] apcaccess [] NIS[] [] .

```
$ apcaccess  
APC : 001,044,1028  
DATE : 2024-01-17 22:17:59 +0900  
HOSTNAME : blurblur  
VERSION : 3.14.14 (31 May 2016) freebsd  
UPSNAME : APCUPS # [] [] [] [] [].  
CABLE : Ethernet Link  
DRIVER : SNMP UPS Driver  
UPSMODE : ShareUPS Slave  
STARTTIME: 2024-01-17 22:17:57 +0900  
MODEL : Smart-UPS 1500 # [] [] [] [] [].  
STATUS : ONLINE # [] ONLINE[] [] [] [] [].  
LINEV : 205.0 Volts  
LOADPCT : 24.0 Percent  
BCHARGE : 100.0 Percent  
TIMELEFT : 52.0 Minutes  
MBATTCHG : 5 Percent  
MINTIMEL : 3 Minutes  
MAXTIME : 0 Seconds  
MAXLINEV : 205.0 Volts
```

```
MINLINEV : 204.0 Volts
OUTPUTV  : 205.0 Volts
SENSE     : High
DWAKE     : 1000 Seconds
DSHUTD    : 20 Seconds
DLOWBATT  : 5 Minutes
LOTRANS   : 195.0 Volts
HITRANS   : 265.0 Volts
ITEMP     : 23.0 C
ALARMDEL  : 30 Seconds
BATTV     : 27.0 Volts
LINEFREQ  : 59.0 Hz
LASTXFER  : No transfers since turnon
NUMXFERS  : 0
TONBATT   : 0 Seconds
CUMONBATT: 0 Seconds
XOFFBATT  : N/A
SELFTEST  : OK
STESTI    : OFF
STATFLAG  : 0x05000008
MANDATE   : 09/12/2019
SERIALNO  : 3S1937X12596
BATTDATE  : 12/15/2019
NOMOUTV   : 230 Volts
FIRMWARE  : UPS 09.3 (ID18)
END APC   : 2024-01-17 22:18:20 +0900
```

apcupsd.conf ????

❏ apcupsd.conf ❏❏ ❏❏ ❏❏ ❏❏❏❏ .

```
## apcupsd.conf v1.1 ##
#
# for apcupsd release 3.14.14 (31 May 2016) - freebsd
#
# "apcupsd" POSIX config file
#
```

```
# 0 00 0000 00 000 000000 apcupsd 000 00 00000 000.
# 0 00 0000 000000 000.
#
#
#
# ===== 00 00 00 00 =====
#
# UPSNAME xxx
# 00 0000 00 00 00 UPS 0000 0000 0 00000.
# 00 00 UPS0 00 00 00 000000.
# 0000 EEPROM0 00000 00000. 80 00000 000.
UPSNAME F00
#
# UPSCABLE < cable >
# UPS0 00000 00000 0000 0000 000000.
#
# < cable >0 00 0000 00000 0000 0000 00000:
# simple, smart, ether, usb
#
# 00 00 0000 00 0000 0000 00 00000.:
# 940-0119A, 940-0127A, 940-0128A, 940-0020B,
# 940-0020C, 940-0023A, 940-0024B, 940-0024C,
# 940-1524C, 940-0024G, 940-0095A, 940-0095B,
# 940-0095C, 940-0625A, M-04-02-2000
#
UPSCABLE ether
#
# apcupsd0 000000 0000 00 00000 00000 0 0000 00000 00
# UPS 0000 00000 UPSTYPE0 00000 0000(0000 0000 00 00).
# 00 000000 00 DEVICE0 00000 0000.
# USB UPS0 00 DEVICE 00000 00 0000.
# 00 UPS 0000 00 0000 00 00 0000 00000 0000.
#
#
# UPSTYPE DEVICE Description
# apcsmart /dev/tty** 00 0000(USB0 00)0 00000 00000 0000
# 0000 00 00 00 000000.
#
#
# usb < BLANK > 00000 0 UPS0 USB0000.
# 00 0000 00000 00 0000 0000000 00000
```

```

#                               0000 00 0000 000000.
#
# net      hostname:port      apcupsd 0000 00 0000 00 0000
#                               apcupsd 00 0000 00. 0000 000 0000
#                               UPS 000000 00 00 0000 0000 00
#                               0000 000000.
#
# snmp      hostname:port:vendor:community
#
#                               SNMP 00 UPS 0000 00 SNMP 0000 000000.
#                               hostname 000000 00 UPS IP 00 00 0000 000000.
#                               vendor "APC" 00 "APC_NOTRAP" 0 0 0000.
#                               "APC_NOTRAP" SNMP 00 0000 00000000
#                               000000 "APC" 000000. 0000 00 161000.
#                               000000 00 "private"0000.
#
# netsnmp   hostname:port:vendor:community
#
#                               OBSOLETE
#
#                               00 SNMP 000000 net-snmp 0000000 0000 000.
#                               0 00 000000 0000 0000 0000 0000 00
#                               'snmp' 000000 00 000000.
#
# dumb      /dev/tty**         simple-signaling UPS 00 0000 00
#
#                               00 00 00 000000.
#
# pcnet      ipaddr:username:passphrase:port
#
#                               AP9617 0000 00 00 000000 SNMP 000000
#                               0000 0 00 PowerChute 0000 00 00000000.
#                               password 0000 0000 00 000000. port UPS
#                               000000 0000 00 00(000000 3052)0000. 0 000000
#                               00 0000 0000 00 000000 3052 000000.
#
# modbus     /dev/tty**        MODBUS 000000 000000 00 SmartUPS
#
#                               0000 00 0000 0 00 00 000000.
# modbus     <BLANK>           MODBUS over USB 00 00 0000 00 0000
#
#                               UPS 00 0000 000000 apcupsd 00 0000
#                               00000000 0000(USB UPS 0 0 0000 00 0000).
#
UPSTYPE snmp
DEVICE IP:PORT:APC:PRIVATE

```

```

# POLLTIME <int>

#   apcupsd UPS  0000 0000 00(0)000.
#   0 0000 00 0000 UPS(UPSTYPE apcsmart, usb, dumb) 000000 0000
#   UPS(UPSTYPE net, snmp) 0000 000000. 0 0000 0000 CPU 00000 00000 00
#   00 00000 00 apcupsd 00000 000000. 00000 60 00000 0000 000000.
#POLLTIME 60


# LOCKFILE <path to lockfile>
#   00 00 0000 000000. 00 0000 0000 00000000. 0 000000
#   00 00000 00, apcupsd 0 000000 00000 00000.
#   00 0000 00 0000 DEVICE 000 000000.
#   Win32 0000 00000 00000.
LOCKFILE /var/spool/lock


# SCRIPTDIR <path to script directory>
#   apccontrol 0 0000 000000 00 00000.
SCRIPTDIR /usr/local/etc/apcupsd


# PWRFAILDIR <path to powerfail directory>
#   00 00 0000 0000 0000 00000000. 0 0000 apcupsd
#   0000 0000 0000 0 00000, OS 00 00000000 killpower
#   (UPS 00 00 00) 0 00000 0000 000000.
PWRFAILDIR /var/run


# NOLOGINDIR <path to nologin directory>
#   nologin 0000 0000 00000000. 0 0000 0000 0000
#   OS 00 0 00000 000000 0000 000000.
NOLOGINDIR /var/run


#
# ===== 00 0 00000 00 00000 =====
#


#   ONBATTERYDELAY 0000 0000 00000
#   00000 00000 0000 00000 00(0)000.
#
#   0, 0000 00000 00 powerout 0000 00000 apccontrol 000000.
#   0000 onbattery 0000 onbatterydelay 00 00000 apccontrol 000000.
#   00 00000 00 0000 00 00 0000 apccontrol powerout 00 00 00 0000

```

```
#   0 000 000 00 000 000.
```

```
ONBATTERYDELAY 6
```

```
#
```

```
# Note: BATTERYLEVEL, MINUTES 0 TIMEOUT0 00 000000
```

```
# 0 00 000 0000 000 000000.
```

```
#
```

```
# 00 00 00 000 00(UPS00 000)0 000 00 000 00,
```

```
# apcupsd0 000 000 000000.
```

```
BATTERYLEVEL 5
```

```
# 00 00 00 00 00(0 00)0 (UPS00 000000 000) 0
```

```
# 000 00, apcupsd0 000 000 000000.
```

```
MINUTES 3
```

```
# 00 00 UPS0 0000 0000 00 TIMEOUT0 00 00
```

```
# 0000 apcupsd0 000 000 000000.
```

```
# 00 000 0 0000 00000000.
```

```
#
```

```
# 000 UPS0 0000 00 0000 000 0000 0 0000
```

```
# 000000 00 00 0000. 000 00 00 0000 BATTERYLEVEL 000
```

```
# 000000 00 000 00 000 MINUTES 000 000 000 UPS0 0000 00 000000.
```

```
# 00 000 00 00 0 00 6000 0000 00 0000 000 0000 000 000000.
```

```
# 00 dumb UPS0 0000 00 0 00 0000 000 0 00 0000 00 0000 00 0000.
```

```
TIMEOUT 0
```

```
# 000 0000 000 00 00 signoff0 000 000 00(0)000.
```

```
# 00 00000000.
```

```
ANNOY 300
```

```
# 00 0 00000 00000 0000 000 000 0000000 00 00 00.
```

```
ANNOYDELAY 60
```

```
# 00 0 0000 0000 0 00 000 0000 000000.
```

```
# NOLOGON <string> [ disable | timeout | percent | minutes | always ]
```

```
NOLOGON disable
```

```
# KILLDELAY0 00 00, 000 000 0 000 00(0)0 00 0
```

```
# 000 00000 000 000 apcupsd0 00 000000.
```

```

# 0 000 00 0 apcupsd 0000 000 0 00 00000 0000 00 0000.
# KILLDELAY <seconds> 0 disables
KILLDELAY 0

#

# ==== 0000 00 000 000 ====
#

# NETSERVER [ on | off ] on enables, off disables the network
# information server. If netstatus is on, a network information
# server process will be started for serving the STATUS and
# EVENT data over the network (used by CGI programs).
NETSERVER on

# NISIP <dotted notation ip address>
# IP address on which NIS server will listen for incoming connections.
# This is useful if your server is multi-homed (has more than one
# network interface and IP address). Default value is 0.0.0.0 which
# means any incoming request will be serviced. Alternatively, you can
# configure this setting to any specific IP address of your server and
# NIS will listen for connections only on that interface. Use the
# loopback address (127.0.0.1) to accept connections only from the
# local machine.
NISIP 0.0.0.0

# NISPORT <port> default is 3551 as registered with the IANA
# port to use for sending STATUS and EVENTS data over the network.
# It is not used unless NETSERVER is on. If you change this port,
# you will need to change the corresponding value in the cgi directory
# and rebuild the cgi programs.
NISPORT 3551

# If you want the last few EVENTS to be available over the network
# by the network information server, you must define an EVENTSFILE.
EVENTSFILE /var/log/apcupsd.events

# EVENTSFILEMAX <kilobytes>
# By default, the size of the EVENTSFILE will be not be allowed to exceed
# 10 kilobytes. When the file grows beyond this limit, older EVENTS will
# be removed from the beginning of the file (first in first out). The

```

```
# parameter EVENTSFILEMAX can be set to a different kilobyte value, or set
# to zero to allow the EVENTSFILE to grow without limit.
EVENTSFILEMAX 10

#
# ===== Configuration statements used if sharing =====
#           a UPS with more than one machine

#
# Remaining items are for ShareUPS (APC expansion card) ONLY
#

# UPSCLASS [ standalone | shareslave | sharemaster ]
#   Normally standalone unless you share an UPS using an APC ShareUPS
#   card.
UPSCLASS shareslave

# UPSMODE [ disable | share ]
#   Normally disable unless you share an UPS using an APC ShareUPS card.
UPSMODE disable

#
# ===== Configuration statements to control apcupsd system logging =====
#

# Time interval in seconds between writing the STATUS file; 0 disables
STATTIME 0

# Location of STATUS file (written to only if STATTIME is non-zero)
STATFILE /var/log/apcupsd.status

# LOGSTATS [ on | off ] on enables, off disables
# Note! This generates a lot of output, so if
#       you turn this on, be sure that the
#       file defined in syslog.conf for LOG_NOTICE is a named pipe.
# You probably do not want this on.
LOGSTATS off

# Time interval in seconds between writing the DATA records to
# the log file. 0 disables.
```


DATETIME 0

```
# FACILITY defines the logging facility (class) for logging to syslog.
#       If not specified, it defaults to "daemon". This is useful
#       if you want to separate the data logged by apcupsd from other
#       programs.
#FACILITY DAEMON

#
# ===== Configuration statements used in updating the UPS EPROM =====
#

#
# These statements are used only by apctest when choosing "Set EEPROM with conf
# file values" from the EEPROM menu. THESE STATEMENTS HAVE NO EFFECT ON APCUPSD.
#

# UPS name, max 8 characters
#UPSNAME UPS_IDEN

# Battery date - 8 characters
#BATTDATE mm/dd/yy

# Sensitivity to line voltage quality (H cause faster transfer to batteries)
# SENSITIVITY H M L          (default = H)
#SENSITIVITY H

# UPS delay after power return (seconds)
# WAKEUP 000 060 180 300    (default = 0)
#WAKEUP 60

# UPS Grace period after request to power off (seconds)
# SLEEP 020 180 300 600    (default = 20)
#SLEEP 180

# Low line voltage causing transfer to batteries
# The permitted values depend on your model as defined by last letter
# of FIRMWARE or APCMODEL. Some representative values are:
#   D 106 103 100 097
#   M 177 172 168 182
```

```
# A 092 090 088 086
# I 208 204 200 196      (default = 0 => not valid)
#LOTRANSFER 208

# High line voltage causing transfer to batteries
# The permitted values depend on your model as defined by last letter
# of FIRMWARE or APCMODEL. Some representative values are:
# D 127 130 133 136
# M 229 234 239 224
# A 108 110 112 114
# I 253 257 261 265      (default = 0 => not valid)
#HITRANSFER 253

# Battery charge needed to restore power
# RETURNCHARGE 00 15 50 90 (default = 15)
#RETURNCHARGE 15

# Alarm delay
# 0 = zero delay after pwr fail, T = power fail + 30 sec, L = low battery, N = never
# BEEPSTATE 0 T L N      (default = 0)
#BEEPSTATE T

# Low battery warning delay in minutes
# LOWBATT 02 05 07 10     (default = 02)
#LOWBATT 2

# UPS Output voltage when running on batteries
# The permitted values depend on your model as defined by last letter
# of FIRMWARE or APCMODEL. Some representative values are:
# D 115
# M 208
# A 100
# I 230 240 220 225      (default = 0 => not valid)
#OUTPUTVOLTS 230

# Self test interval in hours 336=2 weeks, 168=1 week, ON=at power on
# SELFTEST 336 168 ON OFF (default = 336)
#SELFTEST 336
```


Certbot : ??? ???? ??

?? ??

- 1 11 1111 111 111 111 111 .
- 80111 1111 1111 111 1111 .

11 801 111 1111 1111 111 11111 .

FreeBSD111 11 111 11111 . IPv61 1111 -4 11 -61 1111 .

```
sudo sockstat -4 -l
```

1111 net-tools1 sudo apt install net-tools1 111 11111 .

```
netstat -tnlp | grep 80
```

NginX1 Apache 111 11111 1111 111111 .

???? ??? ??

111 11111 .

```
sudo certbot certificates
```

11 11111 111 111 1111 . 1111 11111 11 111 11 111 11111 .

```
sudo certbot delete
```

11111 1111 1111 111111 .

??? ??

11 111 1 1111 .

```
sudo certbot renew
```

11 11 1111 11 1111 1111 11 --dry-run 111 111 . 11111 1111 111 11 11111 11111 .

Shell? sh?? bash? ??

你 在 哪 里 安装 bash shell 呢 系统 默认 安装 了 没有 呢 .

你 安装 bash 系统 默认 安装 了 没有 呢 .

```
sudo pkg install bash
```

你 在 哪 里 安装 了 系统 默认 安装 了 没有 呢 , 你 安装 了 没有 呢 .
你 安装 了 bash 系统 默认 安装 了 没有 呢 .

```
$ which bash
/usr/local/bin/bash
```

/usr/local/bin/bash 系统 默认 安装 了 没有 呢 . 你 安装 了 chsh (change shell 你 安装 了 没有 呢) 你 安装 了 没有 呢 .
你 安装 了 系统 默认 安装 了 没有 呢 . 你 安装 了 系统 默认 安装 了 没有 呢 user id 你 安装 了 系统 默认 安装 了 没有 呢 . 你 安装 了 系统 默认 安装 了 没有 呢 .

```
$ chsh -s /usr/local/bin/bash      # 你 安装 了 系统 默认 安装 了 没有 呢
$ chsh -s /usr/local/bin/bash ID  # ID 你 安装 了 系统 默认 安装 了 没有 呢
```

你 安装 了 系统 默认 安装 了 没有 呢 你 安装 了 系统 默认 安装 了 没有 呢 .

```
chsh: user information updated
```

你 安装 了 系统 默认 安装 了 没有 呢 你 安装 了 系统 默认 安装 了 没有 呢 .
你 安装 了 系统 默认 安装 了 没有 呢 你 安装 了 系统 默认 安装 了 没有 呢 , 你 安装 了 系统 默认 安装 了 没有 呢 bash 你 安装 了 系统 默认 安装 了 没有 呢 .

alias? bash shell?? ?? ??? (.bashrc? ??? .bash_profile)

我 在 虚拟机 里面
FreeBSD 14.0 的 bash shell 中 .bashrc 文件 和 .bash_profile 文件 都没有
找到。

我 在 终端 输入 ls 命令 发现 在 根目录 下 有一个 alias 文件 。

```
$ nano .bash_profile
```

我 编辑 了 .bash_profile 文件 并 添加 了 以下 内容 。

```
alias ll='ls -G'  
alias lsg='ls -alG'
```

然后 我 在 终端 输入 -G 命令 发现 在 根目录 下 有一个 .ll 文件 和 一个 .lsg 文件 。

我 在 终端 输入 alias 命令 发现 ：

```
$ alias
```

我 发现 alias 命令 的 输出 是 ：

```
$ ualias <alias>
```

我 发现 在 .bash_profile 文件 中 添加 了 alias 命令 。

exFAT ?????? ??????

<https://www.micski.dk/2021/04/10/how-to-mount-exfat-formatted-sd-memory-card-on-freebsd/>

FreeBSD GUI .

```
$ sudo pkg install fusefs-exfat
$ sudo kldload fusefs
$ sudo nano /boot/loader.conf
# 
fusefs_load="YES"
$ sudo shutdown -r now
```

.

ssh ????? ?????

<https://cho001.tistory.com/20>

1. ssh key pair을 생성합니다.

```
# -b 4096 2048비트 키를 생성합니다.  
ssh-keygen -t rsa -b 4096
```

2. ssh-copy-id로 원격 호스트에 키를 복사합니다.

```
# <remote-host>에 키를 복사합니다.  
# 예: foobar@192.168.0.230  
$ ssh-copy-id -i ~/.ssh/id_rsa.pub <remote-host>
```

3. 원격 호스트에 접속합니다. (예: foobar@192.168.0.230)

4. scp를 사용하여 파일을 원격 호스트에 복사합니다.

GitHub Pull? merge ??? ???

1. 本地代码与远程仓库同步
2. 创建分支
3. 提交代码
4. 推送代码到远程仓库
5. 在远程仓库创建 Pull Request
6. 等待他人审核
7. 合并代码

```
$ sudo git fetch --all
$ sudo git reset --hard origin/master
# 本地 BookStack 仓库与远程仓库 release/master 分支同步。
$ sudo git pull
# 本地 BookStack 仓库与远程仓库 git pull origin release 分支同步。
```

tar ?????

tar 是 Linux 系统中用于打包和压缩文件的工具。它可以将多个文件或目录打包成一个 tar 文件，并可以选择性地压缩。tar 文件的扩展名通常是 .tar。压缩后的文件扩展名通常是 .tar.gz 或 .tar.bz2。tar 命令的基本语法如下：

```
$ tar -cvf- TargetDir | split -b 1024m - DestinationFile.tar
```

- TargetDir : 要打包的目录
- 1024m : 每个分片的文件大小 (默认为 1GB)
- DestinationFile.tar : 打包后的文件名称

打包后的文件名称可以是 File.tara File.tarb File.tarc 等。打包后的文件名称可以是 File.tara File.tarb File.tarc 等。

tar 命令

```
$ cat CompFile.tar* | tar xvf -
```

tar 命令的基本语法如下：

tar 命令的基本语法如下：
tar : <https://superad.tistory.com/74>