



OpenBSD



- **Come e dove recuperare OpenBSD**

- comprare il set dei cd, in questo modo si contribuisce a tenere in vita il progetto. Per ordinazioni in Europa:
<https://https.openbsd.org/cgi-bin/order.eu>
- ottenere i file dai server (http, ftp, afs, rsync) per poi crearsi l'iso o per installare il tutto direttamente dalla rete:
<http://www.openbsd.org>
<ftp://ftp.openbsd.org> login: anonymous

- **Installare il sistema**

- 1 Prima installazione veloce tramite rete più cd.iso

Scaricare il cd.iso all' indirizzo <ftp://ftp.openbsd.org/pub/OpenBSD/4.2/i386/cd42.iso> .

Mi riferisco alla versione 4.2 della i386, cambiate 4.2 per scaricare una versione differente o impostate snapshots per avere la versione in sviluppo. Cambiate i386 se la vostra macchina è una Sparc o altro (vedi sul sito ufficiale le architetture supportate).

Prepararsi il disco contenente l' immagine iso.

Prepararsi la slice e/o la partizione.

OpenBSD vuole stare nel primo settore del disco, posizionandolo in altri settori nei casi di più partizioni può creare un “bad magic”.

Quindi prepararsi il disco con fdisk. Per fare un esempio di disco con tre partizioni (OpenBSD, Linux, windows):

prima partizione primaria OpenBSD	(a6)
seconda partizione primaria windows	(c)
terza partizione primaria Linux	(83)
quarta partizione primaria o secondaria linux swap	(82)

Date le giuste id alle partizioni :

a6 per OpenBSD,

c per windows,

82 linux/swap

83 Linux.

Per creare le partizioni con fdisk:

menhir-# fdisk /dev/hda

....

Command (m for help):

digitando **m** avremmo la lista dei comandi.

Se volessimo stampare a video la tabella digitiamo **p** :

Command (m for help): p

Device	Boot	start	End	Blocks	Id	System
--------	------	-------	-----	--------	----	--------

I comandi che ci servono in questo caso sono:

d delete - cancella la partizione

n new - crea una nuova partizione

t table - assegnazione id

Riporto un esempio di creazione per la partizione OpenBSD, da questa si può prendere spunto per creare le altre partizioni.

Digitaremmo il comando **n** per creare la partizione.

Command (m for help): n

in questo punto ci viene chiesto se vogliamo una partizione primaria o secondaria, scegliamo la primaria per OpenBSD digitando **p**

Command action

e extended

p primary partition (1-4)

p

Scegliamo il primo settore:

Partition number (1-4): 1

diamo il primo cilindro, lasciamo l'opzione di default:

First cylinder (3401-7936, default 3401):

Impostiamo la grandezza:

Last cylinder or +sizeM or +sizeK (3401-7936, default 7936):

qua sta all'utente la scelta. Se dovessimo dare 2500 Mb digitiamo **+2500M** e diamo invio.

Diamo il giusto id, in questo caso **a6** con il comando **t**:

```
Command (m for help): t
Partition number (1-4): 1
Hex code (type L to list codes): a6
Changed system type of partition 1 to a6 (OpenBSD)
```

Stampiamo a video la tabella:

```
Command (m for help): p
```

Device	Boot	start	End	Blocks	Id	System
/dev/hda1		1	203	102311+	a6	OpenBSD

(non prendete in considerazione questa tabella perchè i valori non sono corretti è solo una schermata d'esempio.)

Create le altre partizioni, se ne avete il bisogno, nello stesso modo ma impostando l'id corrispettivo al sistema.

Se volete dedicare tutto il disco ad OpenBSD potete impostare il disco durante l'installazione o direttamente con fdisk lasciando solo una partizione primaria OpenBSD (a6).

OpenBSD chiede uno spazio minimo quindi sta a voi decidere quanto disco volete dedicare a questo sistema. Per darvi un' idea su uno zaurus con il sistema completo, dove sono presenti i sorgenti più i vari ports, lavoro bene con 2 Gb di disco anche se ho lavorato con meno spazio ma trovo problemi quando compilavo dai ports.

Al boot avviato del cd.iso arriveremmo alla seguente schermata:

```
9xxC SCSI
bha3: model BT-958, firmware 5.07B
bha3: sync, parity
scsibus1 at bha3: 8 targets
pcn0 at pci0 dev 17 function 0 "AMD 79c970 PCnet-PCI" rev 0x10, AM79c970A, rev 0
: irq 10, address 00:0c:29:99:91:f9
"Ensoniq AudioPCI97" rev 0x02 at pci0 dev 18 function 0 not configured
isa0 at pcib0
isadma0 at isa0
pckbc0 at isa0 port 0x60/5
pckbd0 at pckbc0 (kbd slot)
pckbc0: using irq 1 for kbd slot
wskbd0 at pckbd0: console keyboard, using wsdisplay0
mpx0 at isa0 port 0xf0/16: using exception 16
pccom0 at isa0 port 0x3f8/8 irq 4: ns16550a, 16 byte fifo
pccom1 at isa0 port 0x2f8/8 irq 3: ns16550a, 16 byte fifo
fdc0 at isa0 port 0x3f0/6 irq 6 drq 2
biomask fbe5 netmask ffe5 ttymask ffe7
rd0: fixed, 3800 blocks
wd0: no disk label
dksum: wd0 matches BIOS drive 0x80
root on rd0a
rootdev=0x1100 rrootdev=0x2f00 rawdev=0x2f02
erase ^?, werase ^W, kill ^U, intr ^C, status ^T
(I)nstall, (U)pgrade or (S)hell? I_
```

Digitando **I** avvieremo l'installazione, **U** avvieremo l'upgrade, **S** avvieremo la shell.

Se disponete di un collegamento wifi date il comando **s** ed impostate il collegamento che vi servirà più avanti:

```
# ifconfig wi0 192.168.0.2 netmask 255.255.255.0 nwid access-point-ssid nwkey 0xvostrachiavehexa64bit
```

digitando **exit** usciremo dalla shell e ritorniamo al menù d'installazione.

Digitiamo **I** per procedere con l'installazione.

Ci verrà chiesto il terminale:

```
Terminal type? [vt220]
```

lasciamo il terminale proposto di default.

Selezioniamo la mappatura della tastiera:

```
kbd(8) mapping? ('L' for list) [none]
```

digitando **L** avremmo la lista, selezioniamo la mappatura adatta che in questo caso sarà **it**.
In futuro si potrà cambiare la mappatura con il comando:

```
bash-# wscnctl -w keyboard.encoding=it
```

cambiate **it** con la mappatura adatta alle vostre esigenze.

Ci sarà chiesto se siamo sicuri di continuare:

```
Proceed with install? [no]
```

Se siamo sicuri digitiamo **yes**.

Scelta del disco d'installazione:

```
Available disks are: wd0.  
Which one is the root disk? (or 'done') [wd0]
```

Se non abbiamo altri dischi (come in questo caso), lasciate l'opzione di default **wd0**.

Se dovessimo avere altri dischi avremmo la possibilità di scegliere il disco di destinazione tra quelli rilevati, esempio:

Available disks are: wd0, sd0, wd1.

Ci verrà chiesto se vogliamo dedicare l'intero disco ad OpenBSD:

```
Do you want to use *all* of wd0 for OpenBSD? [no]
```

In questo caso digitiamo **no**.

Se volessimo dedicare l'intero disco digiteremmo **yes**.

Impostiamo la slice. Avremmo qualcosa del genere:

```
Disk: wd0 geom etry: 2482/255/63 [39873330 sectors]
Offset: 0 Signature: 0xAA55
Starting Ending LBA Info:
#: id C H S - C H S [ start: size ]
-----
0: 00 0 0 0 - 0 0 0 [ 0: 0 ] unused
1: 00 0 0 0 - 0 0 0 [ 0: 0 ] unused
2: 00 0 0 0 - 0 0 0 [ 0: 0 ] unused
*3: A6 0 1 1 - 2481 254 63 [ 63: 39873267 ]OpenBSD
```

Avendo già impostato la slice con fdisk il sistema lo riconosce in automatico.
Se non avessimo impostato la slice bisogna crearla in questo punto (vedi fdisk).
Per uscire da questa schermata e salvare se avete effettuato delle modifiche al prompt digitate **w** e poi **q** che sta per w=write=scrivi, q=quit=chiudi.

Impostare la partizione OpenBSD:

```
# Inside MBR partition 3: type A6 start 63 size 8385867
Treating sectors 63-8385930 as the OpenBSD portion of the disk.
You can use the 'b' command to change this.
Initial label editor (enter '?' for help at any prompt)
>
```

Digitando **?** Avremmo la lista dei comandi.

I comandi che ci servono in questo passo sono:

p m p stampa la tabella, l'aggiunta dell'opzione m stampa le grandezze in Mb.
d delete, cancella
a crea

Al prompt digitiamo **p** per visualizzare la tabella:

```
> p
device: /dev/rwd0c
type: ESDI
disk: ESDI/IDE disk
label: VMware Virtual I
bytes/sector: 512
sectors/track: 63
tracks/cylinder: 16
sectors/cylinder: 1008
cylinders: 8322
total sectors: 8388608
free sectors: 8385867
rpm: 3600

16 partitions:
#          size          offset  fstype [fsize bsize  cpg]
a:      8385867           63  unused     0     0
c:      8388608            0  unused     0     0
>
```

Cancelliamo la partizione **a** con il comando:

```
> d a
```

Attenzione a non cancellare la partizione **c**

Creiamo la stessa partizione ma lasciando spazio per un'altra partizione (**b**) swap.

```
> a
partition: [a]
offset: [63]
size: [8385867] 3000M
Rounding to nearest cylinder: 6143697
FS type: [4.2BSD]
mount point: [none] /
```

Creazione dello swap:

```
> a
partition: [b]
offset: [6143760]
size: [2242170] 512M
Rounding to nearest cylinder: 1048320
FS type: [swap]
>
```

Visualizzate le partizioni con il comando **p m** per verificare le grandezze.

Se volete impostare nuovamente le partizioni cancellate con **d** le partizioni **a** e **b**

```
> d a
```

```
> d b
```

e ricreatele con il comando **a**.

Salvate i cambiamenti con il comando **q** :

Ci verrà chiesto se vogliamo scrivere la nuova label. A questo punto se risulta essere tutto giusto diamo il comando **y** per scrivere la tabella :

```
> q
Write new label?: [y] y
No more disks to initialize.

OpenBSD filesystems:
wd0a /
```

Un' ulteriore conferma per la scrittura del disco.

Diamo **yes** per continuare :

```
The next step *DESTROYS* all existing data on these partitions!
Are you really sure that you're ready to proceed? [no] yes
```

Impostazione del network.

Impostiamo l' hostname, l' interfaccia, l' indirizzo, il DNS e il route:

```
System hostname? (short form, e.g. 'foo') thirdtry
Configure the network? [yes]
Available interfaces are: pcn0.
Which one do you wish to initialize? (or 'done') [pcn0]
Symbolic (host) name for pcn0? [thirdtry]
The media options for pcn0 are currently
    media: Ethernet autoselect (autoselect)
Do you want to change the media options? [no]
IPv4 address for pcn0? (or 'none' or 'dhcp') dhcp
Issuing hostname-associated DHCP request for pcn0.
DHCPDISCOVER on pcn0 to 255.255.255.255 port 67 interval 2
DHCPOFFER from 192.168.0.1
DHCPREQUEST on pcn0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.98 -- renewal in 300 seconds.
IPv6 address for pcn0? (or 'rtsol' or 'none') [none]
No more interfaces to initialize.
DNS domain name? (e.g. 'bar.com') [mshome.net]
DNS nameserver? (IP address or 'none') [192.168.0.1]
Use the nameserver now? [yes] yes
Default IPv4 route? (IPv4 address, 'dhcp' or 'none') [dhcp]
Edit hosts with ed? [no]
Do you want to do any manual network configuration? [no]
```

In questo caso verrà dato un indirizzo dinamico impostando l'interfaccia in dhcp.
Se volessimo impostare un indirizzo statico digitare l'indirizzo al posto di dhcp.
Cambiate hostname, domain name ed il nameserver in base alle vostre configurazioni.

Impostare la password di root:

```
Password for root account? (will not echo)
Password for root account? (again)
```

Iniziare l'installazione del set di pacchetti tgz:

```
Let's install the sets!
Location of sets? (cd disk ftp http or 'done') [cd] http
HTTP/FTP proxy URL? (e.g. 'http://proxy:8080', or 'none') [none]
```

Scegliere la locazione dei pacchetti (cd, ftp, http), se non avete scaricato i pacchetti in un cd scegliere ftp o http, in questo caso è stato scelto http.

Alla domanda:

```
Display the list of known http servers? [no]
```

se si sceglie yes verrà visualizzata la lista dei server, scegliendo no bisogna impostare la locazione dei packages. Noi scegliamo yes.

scegliere il server (se preferite i primi server, date il comando q per chiudere la lista).

Scegliamo il server **39** e confermiamo:

```
Server? (IP address, hostname, list#, 'done' or '?') 39
Using  openbsd.mirrors.tds.net/pub/OpenBSD Madison, WI, USA
Server? (IP address, hostname, list#, 'done' or '?') [openbsd.mirrors.tds.net]
```

Confermiamo la directory dove verranno prelevati i pacchetti:

```
Server directory? [pub/OpenBSD/4.2/i386]
```

A questo punto selezioniamo i pacchetti di sistema da installare.

I pacchetti di sistema disponibili sono:

bsd, bsd.rd, bsd.mp, etc42.tgz, misc42.tgz, comp42.tgz, man42.tgz, game42.tgz, xbase42.tgz, xetc42.tgz, xshare42.tgz, xfont42.tgz e xserv42.tgz

Per aggiungere un pacchetto digitare nel prompt il nome presente nella lista, per togliere un pacchetto digitare – prima del nome del pacchetto (es: -bsd.rd o -game42.tgz nel caso del pacchetto game), se si vuole installare tutto il set digitate **all** .

Fatte le giuste scelte dare il comando **done**, che dopo un'ulteriore conferma farà partire il download ed installazione dei pacchetti scelti.

Finito il download digitate nuovamente **done** se non avete altre locazioni dove scaricare altri pacchetti di sistema.

Scegliere se far partire il demone sshd, ntpd e se volete far girare X window :

```
Start sshd(8) by default? [yes]
Start ntpd(8) by default? [no]
Do you expect to run the X Window System? [no]
Change the default console to com0? [no]
Saving configuration files...done.
Generating initial host.random file...done.
```

Impostare il time zone:

```
What timezone are you in? ('?' for list)
```

Nel mio caso: **Europe/Rome**

Dopo il making device l'installazione sarà conclusa. Alla shell # diamo **reboot** ed avremmo sulla macchina OpenBSD installato.

2 Preparazione Zaurus modelli sl-c3x00 per installazione OpenBSD in dual boot

[Prefazione]

L'installazione e' stata fatta su uno sharp Zaurus modello sl-c3000 con sopra cacko , per l'installazione non fa' differenza con la Qtopia o qualsiasi altra rom che permetta il dual boot. Installeremo OpenBSD tenendo Linux. Siamo alla versione 3.7 di OpenBSD vicino alla 3.8 quindi seguo con la snapshot.

- Se non presente installare un terminale tipo qkonsole, terminal, embedddkonsole, etc.
- Recuperatevi una scheda wifi compactflash e/o qualcosa che permette un collegamento ai server OpenBSD. Per conoscere l'hardware supportato da OpenBSD vedi <http://www.openbsd.org/zaurus.html#hardware>

[Start]

Collegatevi in rete con una scheda wireless e/o quant'altro. Andiamo su "Add/Removes Software" nella sezione "Settings" di qtopia o cacko. Clickiamo con la stilo su "Install packages via networks". Clickiamo con la stilo sull'icona gialla (una sorta di levetta) situata in basso a destra.

Aggiungiamo un nuovo server clickando su "new".

Su "Name:" mettiamo OpenBSD o quello che si vuole.

Su "URL:" aggiungiamo questo link:

```
ftp://ftp.openbsd.org/pub/OpenBSD/snapshots/zaurus
```

possiamo cambiare "snapshots" con la versione che vogliamo.

In alto nella sezione "Servers" abbiamo il nome che abbiamo scelto per il server aggiunto, in questo caso OpenBSD.

Sempre con la stilo selezioniamo OpenBSD e chiudiamo clickando su "ok".

Selezioniamo il pacchetto openbsd ed installiamolo.

Apriamo terminal o qualche altra cosa che ci dia la possibilita' di utilizzare la shell.

```
zaurus $  
saliamo di livello (root)
```

```
zaurus $ su  
zaurus #
```

prepariamo il disco.

```
zaurus # fdisk /dev/hda
```

...

Command (m for help):

vediamo la nostra tabella:

Command (m for help): p

Device	Boot	start	End	Blocks	Id	System
/dev/hda1	*	1	203	102311+	83	Linux
/dev/hda2	*	204	1019	411264	83	Linux
/dev/hda3		1020	7936	1200024	c	w95

FAT32 (LBA)

vediamo che ci sono 3 partizioni:

```
hda1  linux
hda2  linux
hda3  w95 fat32 (LBA)
```

ora dobbiamo:

- cancellare hda3 con "d"
- ricrearla ma lasciando spazio per una quarta partizione con "n"
- dare l'id ad hda3 con "t" scegliendo "c" che sta per w95 fat32 (LBA)
- creare la quarta partizione per openbsd con "n"
- dare l'id ad hda4 con "t" scegliendo "a6" che sta per OpenBSD.

cancelliamo hda3

Command (m for help): d
Partition number (1-4): 3

Command (m for help):

vediamo la nostra tabella

Command (m for help): p

Device	Boot	start	End	Blocks	Id	System
/dev/hda1	*	1	203	102311+	83	Linux
/dev/hda2	*	204	1019	411264	83	Linux

ora ricreiamo hda3

scegliamo una partizione primaria,
assegnamo il numero della partizione in questo caso 3,
e gli diamo una dimensione:

Command (m for help): n

Command action

- e extended
- p primary partition (1-4)

p

[scegliamo una partizione primaria]

Partition number (1-4): 3

[scegliamo la partizione 3]

First cylinder (1020-7936, default 1020):

[qua diamo invio]

Last cylinder or +sizeM or +sizeK (1020-7936, default 7936): 3400

[qua possiamo scegliere la dimensione che interessa]

Command (m for help):

diamo l'id ad hda3:

Command (m for help): t

Partition number (1-4): 3

[scegliamo la partizione a cui vogliamo dare l'id]

Hex code (type L to list codes): c

[scegliamo quella che ci serve mettendo c che sta per W95 FAT32 (LBA)]

Changed system type of partition 3 to c (W95 FAT32 (LBA))

Command (m for help):

ora creiamo la nostra partizione per openbsd:

Command (m for help): n

Command action

e extended

p primary partition (1-4)

p

[scegliamo una partizione primaria]

Partition number (1-4): 4

[scegliamo la partizione 4]

First cylinder (3401-7936, default 3401):

[qua diamo invio]

Last cylinder or +sizeM or +sizeK (3401-7936, default 7936):

[qua possiamo scegliere la dimensione che interessa]

Command (m for help):

diamo l'id ad hda4:

Command (m for help): t

Partition number (1-4): 4

[scegliamo la partizione a cui vogliamo dare l'id]

Hex code (type L to list codes): a6

[scegliamo quella che ci serve mettendo "a6" che sta per OpenBSD]

Changed system type of partition 4 to a6 (OpenBSD)

Command (m for help):

rivediamo la nostra tabella:

Command (m for help): p

Device	Boot	start	End	Blocks	Id	System
/dev/hda1	*	1	203	102311+	83	Linux
/dev/hda2	*	204	1019	411264	83	Linux
/dev/hda3		1020	7936	1200024	c	W95 FAT32 (LBA)
/dev/hda4		3401	7936	2286144	a6	OpenBSD

salviamo

Command (m for help): w

ed usciamo

Command (m for help): q

Chiudiamo la shell ed andiamo nella sezione "Applications" di Qtopia o cacko. Clickiamo con la stilo sull'icona con puffy chiamata "Install". a questo punto parte il boot d'installazione openbsd e va seguita un'installazione tipica.

Una volta conclusa l'installazione fate un reboot.

[Problemi]

al boot se dovrete avere problemi tipo non riconoscimento del disco seguite quanto segue:

```
boot> r
```

cosi' dovrebbe partire il boot di linux
se cosi' non fosse tipo messaggio di errore "HDD1 error...etc,etc."
date un ^C
e vedrete partire il boot di linux.

● Kernel – compilazione

Prendere i sorgenti dai cd ufficiali o scaricare i sorgenti dal server openbsd.org o tramite sincronizzazione cvs.

I file da scaricare sono **src.tar.gz** ed **sys.tar.gz** in base alla versione installata.

Per la sincronizzazione.

1- impostiamo l'ambiente:

```
# export CVS_RSH=/usr/bin/ssh  
# export CVSROOT=anoncvs@anoncvs.nl.openbsd.org:/cvs
```

2 – otteniamo i sorgenti:

```
# /usr/bin/cvs checkout -rOPENBSD_4_2 src
```

Se abbiamo prelevato i file tramite ftp o http:

decomprimere i file in /usr/src/

```
# pwd  
/usr/src  
# tar xvfz src.tar.gz && rm src.tar.gz  
# tar xvfz sys.tra.gz && rm sys.tar.gz
```

Editiamo il file che definisce le funzionalità di sistema, indipendenti dall'architettura:

```
# cd /usr/src/sys/conf
# cp GENERIC merdules
# vi merdules
```

Commentando o de-commentando le opzioni che interessano con il # .
Salviamo e andiamo ad editare il file che definisce i device driver
da caricare per la nostra architettura:

```
# cd /usr/src/sys/arch/i386/conf
# cp GENERIC merdules
# vi merdules
```

Stessa cosa anche qua con i commenti, facendo attenzione al 13° rigo,
dove dovremmo impostare GENERIC con il nuovo nome dato al kernel
se è stato fatto.

Esempio:

```
include      "../..../conf/GENERIC"
```

in questo caso verrà cambiato in:

```
include      "../..../conf/merdules"
```

se volessimo il supporto il supporto bluetooth, dovremmo de-commentare le righe:

```
# Bluetooth controllers
sbt*   at sdmmc?   # SDIO Bluetooth
ubt*   at uhub?   # USB Bluetooth
```

un'altro esempio è andare nella sezione:

```
# Wireless network cards
```

e attivare o disattivare i driver che vogliamo utilizzare.

Finite le giuste impostazioni, salviamo e chiudiamo il file e passiamo alla compilazione:

```
# config merdules
# cd ../compile/merdules
# make clean && make depend && make
```

```
# cp /bsd /bsd.old
# cp /usr/src/sys/arch/i386/compile/merdules/bsd /bsd
```

oppure al posto di questi ultimi due comandi lanciare:

```
# make install
```

Fate un reboot ed avrete il sistema con il nuovo kernel.
Se avete salvato il vecchio kernel (bsd.old) nella root al boot di OpenBSD potete scegliere di usare il vecchio kernel lanciando bsd.old al posto di bsd.

Se vogliamo aggiornare i sorgenti, dopo aver impostato l'ambiente:

```
# /usr/bin/cvs update -P -rOPENBSD_4_2 src
```

- **Ricompilare il sistema**

Impostare l'ambiente:

```
# export CVS_RSH=/usr/bin/ssh
# export CVSROOT=anoncvs@anoncvs.nl.openbsd.org:/cvs
```

Aggiornare i sorgenti:

```
# cd /usr/src
# /usr/bin/cvs update -P -rOPENBSD_4_2 src
```

Ricompilare il sistema:

```
# rm -rf /usr/obj/*
# cd /usr/src
# make obj
# cd /usr/src/etc && env DESTDIR=/ make distrib-dirs
# cd /usr/src
# make build
```

- **Ports**

Scaricare il file ports.tar.gz dai server openbsd.org o tramite sincronizzazione.
Attenzione a scaricare i ports in base alla versione installata.

Per la sincronizzazione:

1 – preparazione ambiente:

```
# export CVS_RSH=/usr/bin/ssh
# export CVSRROOT=anoncvs@anoncvs.nl.openbsd.org:/cvs
# cd /usr
```

2 – ottenere i ports:

```
# /usr/bin/cvs checkout -rOPENBSD_4_2 ports
```

Se abbiamo scaricato il file dai server openbsd.org:
Decomprimere il file in /usr

```
# cd /usr
# tar xvfz ports.tar.gz && rm ports.tar.gz
```

Compilazione:

prendo in esempio unzip:

```
# cd /usr/ports/archivers/unzip
# make install clean
```

Cercare un dentro i ports:

```
$ cd /usr/ports
$ make search name=gaim
oppure
$ make search key=gaim
```

alcuni ports hanno delle funzionalità aggiuntive.

Tramite la variabile **FLAVORS** è possibile specificare quale funzione si vuole aggiungere alla compilazione.

Per visualizzare le funzionalità aggiuntive:

```
# cd /usr/ports/net/snort
# make show=FLAVORS
```

se vogliamo la lista dei subpackages:

```
# make show=MULTI_PACKAGES
```

se vogliamo installare la funzionalità **mysql** su snort usando la variabile FLAVORS:

```
# cd /usr/ports/net/snort
# FLAVOR="mysql" make install clean
```

se vogliamo aggiornare i ports:

```
# /usr/bin/cvs update -P -rOPENBSD_4_2 ports
```

- **Pacchetti precompilati**

OpenBSD mette a disposizione dei repository dove si possono trovare pacchetti 'ufficiali' precompilati. In altri server è possibile trovare diversi repository con pacchetti 'non ufficiali' dove OpenBSD non accerta la sicurezza.

I comandi comunemente usati per i packages sono:

pkg_add	installa il pacchetto .tgz
pkg_delete	cancella dal sistema il pacchetto .tgz
pkg_info	informazioni riguardo i pacchetti
pkg_create	per la creazione dei pacchetti

Per installare un pacchetto già scaricato nel sistema:

```
# pkg_add -v screen-4.0.3p0.tgz
parsing screen-4.0.3p0
installed /etc/screenrc from /usr/local/share/examples/screen/screenrc | 71%
screen-4.0.3p0: complete
```

L'opzione -v sta ad indicare il modo verbose che serve per avere informazioni stampate a video durante la sua installazione.

pkg_add ha altre opzioni (per avere tutta la lista delle opzioni usate pkg_add -h o usate il man: man pkg_add) uno dei quali è il sistema -i che sta ad indicare in modo interattivo. Questa opzione è utile quando nello stesso repository abbiamo lo stesso pacchetto con dei subpackages.

Prendendo in esempio un'installazione di screen via rete tramite i repository OpenBSD in modo interattivo:

preparo il sistema:

```
sh-# export PKG_PATH="ftp://ftp.openbsd.org/pub/OpenBSD/4.1/packages/i386"
```

installo il pacchetto:

```
sh-# pkg_add -i screen
```

```
Ambiguous: screen could be screen-4.0.3p0 screen-4.0.3p0-shm screen-4.0.3p0-static
```

```
Choose one package
```

```
0: <None>
```

```
1: screen-4.0.3p0
```

```
2: screen-4.0.3p0-shm
```

```
3: screen-4.0.3p0-static
```

```
Your choice: 1
```

```
screen-4.0.3p0: complete
```

Con l'opzione '-i' il sistema stampa a video una lista di pacchetti, alla richiesta:

Your choice:

dobbiamo dare il numero indicativo al pacchetto che vogliamo installare.

Un'altro modo per installare pacchetti on the fly tramite i repository è questo:

```
pkg_add ftp://ftp.openbsd.org/pub/OpenBSD/4.1/packages/machine -a/screen-4.0.3p0.tgz
```

Con `pkg_info` è possibile avere informazioni riguardo un pacchetto specifico se installato o presente nelle nostre directory:

```
$ pkg_info screen-4.0.3p0
```

se diamo solo il comando `pkg_info` senza passare altri argomenti il sistema stampa a video la lista dei pacchetti installati nel sistema.

- **Alcune configurazioni**

Network

per visualizzare la lista dell'interface :

```
# ifconfig -a
```

configurazione on the fly:

statico

```
# ifconfig r10 192.168.0.2 net mask 255.255.255.0 up
# echo "nameserver 212.216.112.112" > /etc/resolv.conf
# route add default 192.168.0.1
```

dinamico

```
# dhclient r10
```

wifi statico

```
# ifconfig wi0 192.168.0.1 netmask 255.255.255 nwid tuo_ssid nwkey 0x1234567890 up
(0x1234567890 è una chiave hex a 64 bit)
# echo "nameserver 212.216.112.112" > /etc/resolv.conf
# route add default 192.168.0.1
```

wifi dinamico

```
# ifconfig wi0 nwid tuo_ssid nwkey 1234567891011 up
(in questo caso è stata usata una chiave ascii – notare che non è stato incluso 0x)
#dhclient wi0
```

configurazione network nel sistema:

statico

```
# echo "inet r10 192.168.0.2 255.255.255.0 NONE" /etc/hostname.r10
# echo "192.168.0.1" > /etc/mygate
# echo "nameserver 212.216.112.112" > /etc/resolv.conf
```

dinamico

```
# echo "dhcp" > /etc/hostname.r10
```

wifi statico

```
# echo "inet wi0 192.168.0.1 255.255.255 nwid tuo_ssid nwkey 0x1234567890" >
/etc/hostname.wi0
# echo "192.168.0.1" > /etc/mygate
# echo "nameserver 212.216.112.112" > /etc/resolv.conf
```

wifi dinamico

```
# echo "dhcp nwid tuo_ssid nwkey 0x123450" > /etc/hostaname.wi0
```

Varie

Cancellare il file /etc/motd

Editare /etc/rc.conf e attivare il packet filter nella riga:

```
pf=NO
```

impostare a:

```
pf=YES
```

editare il file /etc/pf.conf (file di configurazione di packet filter) e attivarlo:

```
# pfctl -e
```

per disattivarlo

```
# pfctl -d
```

Editare il file /etc/rc.conf e attivare o disattivare le funzioni che interessano.

Editare /etc/rc.shutdown se volete che la vostra macchina si spenga al shutdown.

Editare /etc/wsconsctl.conf in base alle proprie esigenze.

Editare /etc/ssh/sshd_conf, consiglio di lasciare solo il protocollo 2 e di non permettere il login da root:

```
PermitRootLogin no
```

Editare /etc/sysctl.conf e attivare, disattivare le funzioni di sysctl che interessano, consigliato abilitare il routing IP attivando la riga:

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

1 sta per attivato – 0 sta per disattivato

E' possibile attivarlo on the fly tramite shell, in esempio l' abilitazione dello swapencrypt:

```
# sysctl -w vm.swapencrypt.enable=1
```

Per avere la lista della opzioni:

```
# sysctl -a
```

Opzioni al boot:

-s avvio in modalità single user

-a chiede quale device usare come root

-d debugging

Impostare la mappatura della tastiera italiana:
on the fly:

```
# wsconsctl -w encoding=it
```

oppure editare il file /etc/wsconsctl alla riga:

```
keyboard.encoding=it
```

Cambiare ed impostare la shell di default:

```
$ chsh -s zsh
```

visualizzare la slice:

```
# fdisk wd0
```

visualizzare la partizione:

```
# disklabel wd0
```

evilsocket.net

OpenBSD: sending script kiddies in /dev/null

Link di riferimento:

<http://www.evilsocket.net>

<http://www.openbsd.org>

<http://www.openbeer.it>