



Capitolo 1

Introduzione

In questi appunti, che non escludono l'utilizzo della wiki¹ cosa che è, anzi, consigliata per capire le scelte fatte riguardo le procedure di installazione e configurazione del sistema e dei pacchetti, non prenderò in considerazione scenari di utilizzo che prevedano il Dual Boot, ad esempio: Windows + GNU/Linux.

La procedura descritta di seguito comporterà la totale cancellazione del contenuto del disco rigido.

I due scenari di utilizzo saranno quindi:

A. Pc con Bios² e disco partizionato con MBR³.

B. Pc con UEFI⁴ e disco partizionato con GPT⁵.

Nel primo caso farò riferimento allo schema di figura 1.1, che chiamerò "Schema A";

¹<https://wiki.archlinux.org>

²<https://en.wikipedia.org/wiki/BIOS>

³https://wiki.archlinux.org/index.php/Master_Boot_Record

⁴https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface

⁵https://wiki.archlinux.org/index.php/GUID_Partition_Table

Schema di partizionamento usato

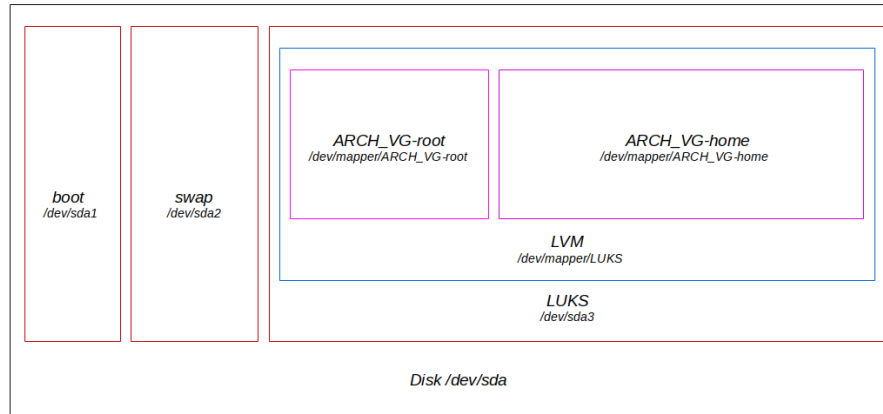


Figura 1.1: «Schema A: Bios + MBR»

e supponendo di volere partizionare un disco da 500 GB il risultato sarà quello descritto nella tabella 1.1

Device	Dimensioni	Flags	Tipo	FS
/dev/sda1	512 MB	boot	linux	ext4
/dev/sda2	4 GB	none	linux-swap	raw
/dev/sda3	476 GB	none	luks	raw
/dev/mapper/ARCH_VG-root	50 GB	none	lvm	ext4
/dev/mapper/ARCH_VG-home	426 GB	none	lvm	ext4

Tabella 1.1: Partizioni schema A

Nel secondo caso farò riferimento allo schema di figura 1.2, che chiamerò "Schema B":

Schema di partizionamento GPT su sistema UEFI

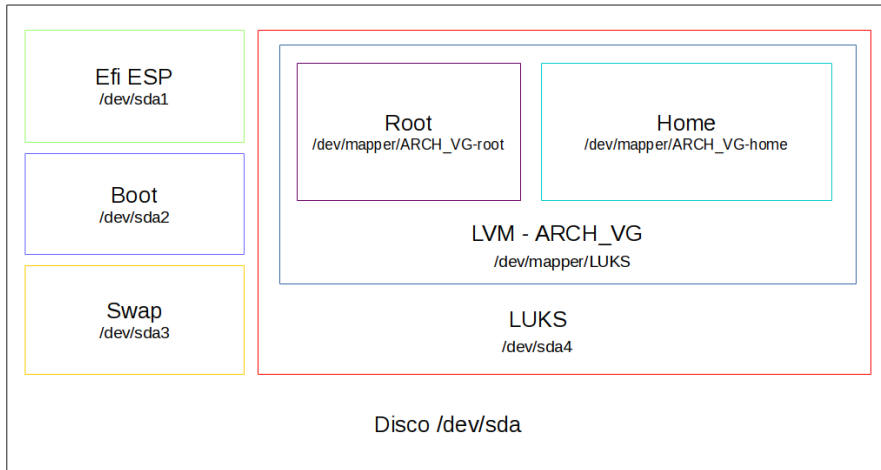


Figura 1.2: «Schema B: Uefi + GPT»

e supponendo di volere partizionare un disco da 500 GB il risultato sarà quello descritto nella tabella 1.2

Device	Dimensioni	Flags	Tipo	FS
/dev/sda1	512 MB	boot,esp	efi	fat32
/dev/sda2	512 MB	none	linux	ext4
/dev/sda3	4 GB	none	linux-swap	raw
/dev/sda4	476 GB	none	luks	raw
/dev/mapper/ARCH_VG-root	50 GB	none	lvm	ext4
/dev/mapper/ARCH_VG-home	426 GB	none	lvm	ext4

Tabella 1.2: Partizioni schema B

Capitolo 2

Operazioni preliminari

2.1 Impostare la tastiera italiana

```
1 loadkeys it
```

Listing 2.1: loadkeys

2.2 Partizionamento

Facendo riferimento allo schema di figura 1.1 ed alla tabella 1.1 Per prima cosa creo una nuova tabella delle partizioni:

```
1 parted /dev/sda mktable msdos
```

Listing 2.2: creazione tabella delle partizioni: schema A

ed ora le partizioni:

```
1 parted -a optimal /dev/sda mkpart primary ext4 1MiB 513MiB set 1
   boot on mkpart primary linux-swap 514MiB 4609MiB mkpart primary
   4610MiB 100%
```

Listing 2.3: creazione partizioni: schema A

Se si fa riferimento allo schema di figura 1.2 ed alla tabella 1.2 creo, allora, una nuova tabella delle partizioni in questo modo:

```
1 parted /dev/sda mktable gpt
```

Listing 2.4: creazione tabella delle partizioni: schema B

e le partizioni:

```
1 parted -a optimal /dev/sda mkpart primary fat32 1MiB 513MiB name 1
   efi set 1 boot on set 1 esp on makpart primary ext4 514MiB 1025
   MiB name 2 boot mkpart primary linux-swap 1026MiB 5122MiB name
   3 swap mkpart primary 5123MiB 100%
```

Listing 2.5: creazione partizioni: schema B

In caso di utilizzo di GPT su sistema Bios qualora si voglia usare GRUB come bootmanager è necessario:

- creare all'inizio del disco una partizione Bios_boot senza alcun filesystem
- contrassegnare tale partizione come *bios boot* oppure come *bios_grub*

Se si usa *gdisk* allora bisogna impostare il tipo della partizione a **EF02**. Mentre se si usa *parted* bisogna impostare il flag **bios_grub**.

Microsoft indica come dimensione minima consigliata per il filesystem fat32: 512MB.

2.3 Volume cifrato

Adesso bisogna creare il volume cifrato che ospiterà i volumi lvm che verranno montati come / e /home.

Per lo schema A:

```
1 cryptsetup -v -y -c aes-xts-plain64 -s 512 -h sha512 -i 7000 --use-  
random luksFormat /dev/sda3
```

Listing 2.6: creazione volume cifrato: schema A

per lo schema B:

```
1 cryptsetup -v -y -c aes-xts-plain64 -s 512 -h sha512 -i 7000 --use-  
random luksFormat /dev/sda4
```

Listing 2.7: creazione volume cifrato: schema B

Apriamo il volume creato:

```
1 cryptsetup luksOpen /dev/sda3 LUKS
```

Listing 2.8: apertura volume cifrato: schema A

per lo schema B:

```
1 cryptsetup luksOpen /dev/sda4 LUKS
```

Listing 2.9: apertura volume cifrato: schema B

LUKS alla fine dei comandi sta ad indicare il nome che si vuole dare al volume cifrato. In questo caso avremo mappato il volume cifrato su */dev/mapper/LUKS*.

Avremmo potuto utilizzare un qualsiasi altro nome a nostro piacere. L'importante è ricordarlo visto che sarà necessario in seguito, al momento della configurazione di Grub.

2.4 LVM

Creiamo i volumi logici che ospiteranno la *root /* e la directory */home*. Questa procedura vale tanto per lo schema A quanto per quello B. Inizializziamo */dev/mapper/LUKS* come volume fisico.

```
1 pvcreate /dev/mapper/LUKS
```

Listing 2.10: creazione volumi logici - pvcreate

Creiamo il gruppo *ARCH_VG*, potremmo usare un altro identificatore. L'importante è tenerlo a mente visto che ci servirà in seguito.

```
1 vgcreate ARCH_VG /dev/mapper/LUKS
```

Listing 2.11: creazione volumi logici - vgcreate

Infine creiamo i due volumi logici */dev/mapper/ARCH_VG-root*, da 50 GB, e */dev/mapper/ARCH_VG-home* assegnandogli tutto lo spazio rimasto.

```
1 lvcreate -L 50G ARCH_VG -n root
2 lvcreate -l 100\%FREE ARCH_VG -n home
```

Listing 2.12: creazione volumi logici - lvcreate

2.5 Formattazione

Formattiamo le partizioni ed i volumi appena creati:

```
1 mkfs -L Boot -m 1 -t ext4 /dev/sda1
2 mkfs -L Root -m 5 -t ext4 /dev/mapper/ARCH_VG-root
3 mkfs -L Home -m 1 -t ext4 /dev/mapper/ARCH_VG-home
```

Listing 2.13: formattazione: schema A

per lo schema B:

```
1 mkfs.vfat -F32 /dev/sda1
2 mkfs -L Boot -m 1 -t ext4 /dev/sda2
3 mkfs -L Root -m 5 -t ext4 /dev/mapper/ARCH_VG-root
4 mkfs -L Home -m 1 -t ext4 /dev/mapper/ARCH_VG-home
```

Listing 2.14: formattazione: schema B

Per adesso lasciamo da parte la partizione *swap* che verrà anch'essa cifrata ma in seguito.

2.6 Connessione alla rete Internet

Questo passo è fondamentale dal momento che dovremo scaricare i pacchetti che via via si renderanno necessari. Nel caso si abbia una scheda Wifi supportata dal Kernel linux, digitare:

```
1 wifi -menu
```

Listing 2.15: conf rete con wifi-menu

Basta seguire la procedura guidata. Oppure in caso di rete cablata basta configurare l'indirizzo ip manualmente o tramite dhcp.

2.7 Montaggio delle partizioni

Nel caso dello schema A:

```
1 mount /dev/mapper/ARCH_VG-root /mnt
2 mkdir -p /mnt/boot
3 mount /dev/sda1 /mnt/boot
4 mkdir -p /mnt/home
5 mount /dev/mapper/ARCH_VG-home /mnt/home
```

Listing 2.16: montaggio partizioni: schema A

per lo schema B:

```
1 mount /dev/mapper/ARCH_VG-root /mnt
2 mkdir -p /mnt/boot
3 mount /dev/sda2 /mnt/boot
4 mkdir -p /mnt/boot/efi
5 mount /dev/sda1 /mnt/boot/efi
6 mkdir -p /mnt/home
7 mount /dev/mapper/ARCH_VG-home /mnt/home
```

Listing 2.17: montaggio partizioni: schema B

2.8 Installazione sistema base

2.8.1 Impostazione dei mirrors

Editare il file `/etc/pacman.d/mirrorlist` e selezionare i mirrors più vicini. Questo file verrà copiato da *pacstrap* nel nuovo sistema.

2.8.2 pacstrap

Installiamo il sistema base e gli strumenti di sviluppo essenziali (*gcc*, *make*, etc.)

```
1 pacstrap /mnt base base-devel
```

Listing 2.18: pacstrap

Capitolo 3

Configurazione sistema base

3.1 Genfstab

```
1 genfstab -U -p /mnt >> /mnt/etc/fstab
```

Listing 3.1: genfstab

L'opzione `-U` fa sì che in `fstab` vengano usate le UUID ma si sarebbero potute usare le labels, create con `mke2fs -L "etichetta"`, usando l'opzione `-L`.

3.2 Chroot

```
1 arch-chroot /mnt /bin/bash
```

Listing 3.2: arch-chroot

3.3 Hostname

Imposto l'`hostname`

```
1 echo <hostname> > /etc/hostname
```

Listing 3.3: set hostname

3.4 Fuso orario

Imposto il fuso orario e l'orologio di sistema

```
1 ln -s /usr/share/zoneinfo/Europe/Rome /etc/localtime
2 hwclock --systohc --utc
```

Listing 3.4: set localtime

3.5 Locales

Imposto la lingua

```
1 sed -i 's/^#it_IT.UTF-8/it_IT.UTF-8/' /etc/locale.gen
2 echo LANG=it_IT.UTF-8 > /etc/locale.conf
3 echo LC_COLLATE=C >> /etc/locale.conf
4 locale-gen
```

Listing 3.5: locale.gen

3.5.1 Tastiera Italiana

Imposto la tastiera

```
1 echo KEYMAP=it > /etc/vconsole.conf
2 echo FONT=ter-116n >> /etc/vconsole.conf
3 echo FONT_MAP=8859-15 >> /etc/vconsole.conf
```

Listing 3.6: tastiera

Avere settato `FONT=ter-116n` richiede l'installazione del pacchetto *terminus-font*.

3.5.2 Tastiera Italiana

Imposto la password di root

```
1 passwd
```

Listing 3.7: password di root

3.6 Installazione gruppo pacchetti 1

Installo alcuni pacchetti utili per terminare l'installazione del sistema di base.

```
1 pacman -S dialog wpa_supplicant rfkill iw
2 pacman -S dhclient net-tools vim haveged terminus-font links
```

Listing 3.8: gruppo pacchetti 1

I pacchetti *dialog wpa_supplicant rfkill iw* sono necessari solo se possediamo e vogliamo usare, al successivo riavvio, la scheda di rete wifi. In caso contrario possiamo tralasciarli, dal momento che il sistema base possiede i pacchetti per configurare la rete cablata. Per comodità installo anche i pacchetti *dhclient, net-tools* e *links*.

Mentre *haveged* aumenta l'entropia disponibile nel sistema e ciò è utile nella fase di inizializzazione delle chiavi di crittografia necessarie per la verifica dei pacchetti.

Se, inoltre, il pc ha una cpu Intel è duopo installare *intel-ucode* che aggiorna a runtime il microcodice della cpu;

```
1 pacman -S intel-ucode
```

Listing 3.9: pacman intel-ucode

se, invece, è presente una cpu AMD non è necessario alcun pacchetto aggiuntivo.

3.7 Installazione e configurazione Boot Manager

Scarico Grub installando con il pacchetto *grub*

```
1 pacman -S grub
```

Listing 3.10: pacman grub

Installo Grub sul hard disk

```
1 grub-install /dev/sda
```

Listing 3.11: grub-install

Da notare che bisogna indicare il disco *sda* e non le partizioni *sda1 sda2* etc

3.7.1 Configurazione Efi Boot Manager

Nel caso in cui il sistema sia UEFI, installo anche il pacchetto *efibootmgr*

```
1 pacman -S efibootmgr
```

Listing 3.12: pacman efibootmgr

Il comando *efibootmgr* serve a gestire le impostazioni relative al Boot Manager UEFI integrate nel firmware della scheda madre.

Considerando i mount points delle partizioni secondo lo schema 2.7, si ha che la partizione **ESP** */dev/sda1* è montata in */boot/efi*.

In caso di dubbi con il comando

```
1 findmnt /boot/efi
```

Listing 3.13: findmnt

si potrà trovare l'esatto punto di mount della partizione efi.

Nel momento in cui si è impartito il comando 3.7 nella directory */boot/efi* è stata creata una subdirectory *ESP* all'interno della quale, a sua volta, né è

stata creata un'altra chiamata *arch*. All'interno di quest'ultima è stato creato il file *grubx64.efi*. Sarà proprio a questo file che il firmware cederà il controllo, non appena terminate le fasi¹ di avvio.

Quindi il path completo del file *grubx64.efi* sarà: */boot/efi/ESP/arch/grubx64.efi*. Il comando da impartire per creare una entry nel menu di boot di UEFI è:

```
1 efibootmgr -c -g -d /dev/sda -p 1 -w -L "<nome entry> -l '\ESP\arch  
  \grubx64.efi'
```

Listing 3.14: efibootmgr setup

Importanti sono le opzioni *-d*, tramite la quale si specifica l'**intero device non la partizione**; l'opzione *-p* con la quale, invece, si indica il numero della partizione sul device: in questo caso la partizione è */dev/sda1* quindi **1**; l'opzione *-L*, tramite cui si specifica una label a piacere con cui chiamare il sistema all'interno del menu di avvio di UEFI ed infine l'opzione *-l*, che indicherà al firmware dove si trova il file eseguibile a cui cedere il controllo per continuare l'avvio del sistema operativo.

Una cosa a cui prestare attenzione è che il percorso, relativo alla directory */boot/efi*, del file *grubx64.efi* è stato specificato utilizzando i backslashes `\` come in Windows.

Da notare che tra l'altro il comando *findmnt* 3.7.1 mostrerà le opzioni con cui è montato il filesystem fat32 della partizione efi. Se fra queste opzioni è presente *utf-8* ciò vuol dire che i nomi di files e directories ed i relativi percorsi sono *case-sensitive*. Quindi il percorso `\ESP\arch\grubx64.efi` sarà diverso da `|esp|arch|grubx64.efi`.

Un utile consiglio è di specificare i percorsi come se il filesystem fosse *case-sensitive*, anche nel caso non lo è.

Rimuovere una entry

Sono possibili due strade:

- utilizzare *efibootmgr*
- utilizzare il comando *bcfg* integrato nella *Efishell v2*

Entrambe le possibilità richiedono due passaggi:

- identificare il numero della entry
- rimuovere la entry

¹A questo proposito si veda https://wiki.archlinux.org/index.php/Unified_Extensible_Firmware_Interface

L'elenco delle entry mostrerà, insieme a quelle create dall'utente, anche le entries predefinite e relative all'avvio da cd, usb, rete, etc.

Bisogna prestare massima attenzione: i comandi che seguono possono compromettere l'usabilità del sistema oltre che la possibilità di avviare il sistema operativo.

Con *efibootmgr* si procede in questo modo. Prima si identifica la entry elencandole tutte:

```
1 efibootmgr -v
```

Listing 3.15: efibootmgr list entries

Una volta individuato il numero relativo alla entry che si desidera cancellare, si impartisce:

```
1 efibootmgr -b # -B
```

Listing 3.16: efibootmgr delete entry

per cancellare la entry numero #.

Invece volendo usare *bcfg* si procede così:

```
1 bcfg boot dump -b
```

Listing 3.17: bcfg list entries

ed una volta ottenuta la lista, con il comando precedente, si cancella in questo modo:

```
1 bcfg boot rm #
```

Listing 3.18: bcfg delete entry

per cancellare la entry numero #. Infine rimuovere, con i normali comandi di gestione dei files del sistema operativo, gli eventuali file ancora presenti nella directory */boot/efi* e relativi alle entries appena cancellate.

3.8 Far riconoscere al sistema la partizione cifrata

Per prima cosa bisogna uscire dall'ambiente *chroot* con il comando *exit* per ritornare temporaneamente all'ambiente live. Questo per individuare gli *UUID* della partizione cifrata con *luks*. Se lo schema usato è quello indicato nella tabella 1.1 allora l'uuid sarà dato da:

```
1 lsblk -o +UUID | grep sda3 | awk '{print $7}' >> /mnt/etc/default/grub
```

Listing 3.19: luks uuid - schema A

Per comodità si è accodato l'uuid al file `/mnt/etc/default/grub` che dovremo configurare fra poco. Prima di rientrare nell'ambiente `chroot` individuino l'identificatore univoco della partizione di swap con il comando:

```
1 ls -lF /dev/disk/by-id/* | grep sda2 | awk '{print $9}' >> /mnt/etc/crypttab
```

Listing 3.20: swap by-id - schema A

e lo accodiamo al file `/mnt/etc/crypttab` Rientriamo nell'ambiente `chroot` come mostrato in 3.2.

3.8.1 Modifica di `/etc/default/grub`

```
1 vim /etc/default/grub
```

Listing 3.21: modifica `/etc/default/grub`

Individuare la riga:

```
1 GRUB_CMDLINE_DEFAULT=""
```

e modificarla in:

```
1 GRUB_CMDLINE_DEFAULT="cryptdevice=UUID=<uuid-of-luks>:LUKS"
```

`<uuid-of-luks>` è l'output del comando 3.8 e che troviamo alla fine del file, `LUKS` è il nome che abbiamo scelto in 2.3 quando abbiamo aperto il volume cifrato. Infine aggiungere la riga:

```
1 LINUX_ROOT_DEVICE="/dev/mapper/ARCH_VG-root"
```

salvare le modifiche ed uscire dall'editor.

3.8.2 Modifica di `/etc/mkinitcpio.conf`

```
1 vim /etc/mkinitcpio.conf
```

Listing 3.22: modifica `/etc/mkinitcpio.conf`

Aggiungere, nella riga che inizia con

```
1 HOOKS="... autodetect filesystems ..."
```

gli hooks² per la gestione delle partizioni cifrate e dei volumi lvm, in questo modo:

```
1 HOOKS="... encrypt lvm2 keyboard keymap... autodetect filesystems... fsck shutdown"
```

²Si veda <https://wiki.archlinux.org/index.php/mkinitcpio>

Gli hooks *encrypt lvm2* devono comparire prima di quello relativo alla rilevazione del filesystem e cioè prima di *filesystems*.

Salvare le modifiche al file, uscire dall'editor.

3.8.3 Configurazione di `/etc/crypttab`

Le modifiche che seguono faranno sì che la partizione di swap venga sì anch'essa cifrata, ma con una password casuale che varia ad ogni riavvio, rendendo impossibile, quindi, operazioni come la sospensione o l'ibernazione, che implicano, appunto, la possibilità di salvare lo stato del sistema operativo sulla swap e il suo successivo recupero per un pronto avvio del sistema operativo stesso. Creiamo, dunque, una entry per la swap cifrata nel file `/etc/crypttab`, apriamo il file:

```
1 vim /etc/crypttab
```

Listing 3.23: modifica `/etc/crypttab`

ed aggiungiamo la riga:

```
1 SWAP /dev/disk/by-id/<swap-part-id> /dev/urandom swap,cipher=aes-  
cbc-essiv:sha256,size=256
```

dove `<swap-part-id>` è l'identificatore univoco della partizione di swap individuato con il comando 3.8, salviamo le modifiche ed usciamo ed apriamo il file `/etc/fstab`

```
1 vim /etc/fstab
```

Listing 3.24: modifica `/etc/fstab`

in cui aggiungiamo a la riga:

```
1 /dev/mapper/SWAP swap swap defaults 0 0
```

In questo caso non si può usare l'UUID della partizione di swap dal momento che esso cambia od ogni riavvio a causa del fatto che la swap viene sostanzialmente ricifrata ad ogni riavvio.

3.8.4 Rigenerazione dell'`initramfs` e di `grub.cfg`

Rigeneriamo l'`initramfs` con il comando:

```
1 mkinitcpio -p linux
```

Listing 3.25: rigenera `initramfs`

e lo stesso facciamo con il file di configurazione di `grub` con il comando:

```
1 grub-mkconfig -o /boot/grub/grub.cfg
```

Listing 3.26: grub-mkconfig

A questo punto possiamo uscire dall'ambiente *chroot* con il comando

```
1 exit
```

Listing 3.27: exit

3.9 Smontaggio partizioni

Smontiamo le partizioni montate. Per lo schema A:

```
1 umount /mnt/home
2 umount /mnt/boot
3 umount /mnt
```

Listing 3.28: unmount - schema A

per lo schema B bisogna smontare anche la partizione *efi*:

```
1 umount /mnt/home
2 umount /mnt/boot/efi
3 umount /mnt/boot
4 umount /mnt
```

Listing 3.29: unmount - schema B

3.10 Riavvio

Possiamo, a questo punto, riavviare con:

```
1 reboot
```

Listing 3.30: reboot