

dd Komanda

dd kopira podatke sa jednog mesta na drugo. Ponekad [cat](#) moze uraditi istu stvar (sa [redirection](#)), ali dd ima opcije za prevodjenje podataka, selektivno kopira samo deo data stream-a, i buffer-uje svoja ucitavanja I upisivanja. dd moze da [copy a CD to an ISO file](#), kopira jednu particiju na drugu, ili da vrati imidz fajl na disk. Koristeci `seek` i `count` opcije, individualni sektor diska moze biti ekstraktovan bez potrebe za cekanjem da se ucita ceo ostatak diska.

dd se isto tako moze koristiti za ekstraktovanje podataka iz, ili ubacivanje podataka u, proizvoljne pozicije u fajlu. Ovo moze biti korisno kao nacin za rad sa binarnim fajlovima iz komandne linije.

Sadržaj

- [1 Primeri](#)
 - [1.1 Kreiranje backup-a hard drajva](#)
 - [1.2 Kreiranje backup image hard drajva](#)
 - [1.3 Kopiranje floppy-ja](#)
 - [1.4 Backup-ujte Master Boot Record \(MBR\).](#)
 - [1.5 Zaobilazenje ogranicanja velicine fajla koristeci split](#)
 - [1.6 Kreiranje praznih disk imidza](#)

Primeri

Kreiranje backup-a hard drajva

Glavne opcije koje ce vas interesovati su `if=` (input file) i `of=` (output file). Podrazumevano, dd cita iz [stdin](#) I pise u [stdout](#). Ovo je jedan primer koriscenja dd komande:

```
# dd if=/dev/md0 of=/dev/hda conv=noerror,sync bs=4k
```

Ova komanda se cesto koristi za kreiranje backup-a drajva (/dev/md0) direktno na drugi hard drajv (/dev/hda). Ovo radi jedino u slucaju da hard drajv ima dovoljno prostora za smestanje fajl sistema izvornog drajva. Prednost ovoga je da nemorate da montirate hard drajv da bi ste napravili backup I jedini referens za hda je u /dev I u komandi koja je obicno u skripti u cron-u.

Opcija "`bs=4k`" se koristi za odredjivanje velicine bloka koji ce se koristiti pri kopiranju. Podrazumevano za dd komandu je 512 bajtova: koriscenje ovako male velicine bloka moze rezultirati veoma sporom kopiranjem. Ipak, losa strana koriscenja vece velicine bloka je da kada se dogodi greska, ostatak bloka se puni sa nultim-bajtovima. Tako da ako povecate velicinu vaseg bloka kada kopirate neispravan uredjaj, izgubivete vise podataka ali cete provesti manje vremena pokusavajući da citate lose sektore. Aati kao sto su [dd_rescue](#) i [dd_rhelp](#) mogu biti bolje resenje u ovakvim slucajevima, kombinujuci brzinu vece velicine bloka za regione bez gresaka sa sa bolje isparcanim kopijama bloka sa regionima koje sadrze greske.

Kreiranje backup image hard drajva

```
# dd if=/dev/hda | gzip > /mnt/hdb1/system_drive_backup.img.gz
```

Ovde dd pravi imidz prvog hard drajva, I prosledjuje ga kroz [gzip](#) program za kompresovanje. Kompresovani imidz se onda stavlja kao fajl na posebnom drajvu. Da bi ste vratili proces:

```
# gzip -dc /mnt/hdb1/system_drive_backup.img.gz | dd of=/dev/had
```

Sada, gzip dekompresuje (-d nastavak) fajl, saljuci rezultate do stdout (-c nastavak), koje se prosledjuju do dd, I onda upisuju u /dev/hda.

Kopiranje floppy-ja

```
# dd if=/dev/fd0 of=/tmp/floppy.img bs=10240
```

Ovo ce kopirati sadrzaj floppy-ja u fajl. Onda, da bi ste stavili imidz na novi floppy, zamenite "if" i "of" parametre.

```
# dd if=/tmp/floppy.img of=/dev/fd0 bs=10240
```

Backup-ujte Master Boot Record ([MBR](#)).

Ovo trebate uraditi pre nego sto editujete vasu particionu tabelu tako da je mozete vratiti u slucaju da nesto zabrljate.

```
# dd if=/dev/hda of=/root/hda.boot.mbr bs=512 count=1
```

Ako se stvari zakomplikuju, mozete boot-ovati [Knoppix](#), mount-ovati particiju koja sadrzi /root (hda1 u ovom primeru) I da vratite MBR sa komandom:

```
# dd if=/mnt/hda1/root/hda.boot.mbr of=/dev/hda bs=512 count=1
```

Zaobilazenje ogranicanja velicine fajla koristeći split

Kada pravite imidze, veoma je lako naleteti na razlicita ogranicenja velicine fajla. Jedan nacin za zaobilazak datog ogranicenja velicine fajla je da se koristi [split](#) komanda.

```
# dd if=/dev/hda1 | gzip -c | split -b 2000m - /mnt/hdc1/backup.img.gz.
```

1. Ovaj primer koristi dd da napravi imidz prve particije na prvom hard drajvu.
2. Rezultati prolaze kroz [gzip](#) za kompresiju
 - o Opcija -c se koristi da prosledi izlazne podatke do [stdout](#).
3. Kompresovani imidz se onda prosledjuje do [split](#) alatke
 - o Opcija -b 2000m govori split alatki koliko veliki treba da budu individualni fajlovi. Mozete koristiti k i m za kilbajte I megabajte (ova opcija podrazumevano koristi bajtove).
 - o Opcija -kaze split alatki da cita iz [stdin](#). U suprotnom, split ce uzeti /mnt/hdc1... kao fajl koji se treba podeliti.
 - o /mnt/hdc1...je prefiks za kreirane fajlove. Split ce kreirati fajlove sa imenima backup.img.gz.aa, backup.img.gz.ab, etc.

Da bi ste vratili multi-file backup, uradite sledece:

```
# cat /mnt/hdc1/backup.img.gz.* | gzip -dc | dd of=/dev/hda1
```

1. Cat sastavlja sadržaj kompresovanih i podeljenih fajlova imidza do [stdout](#), redom.
2. Rezultati se prosledjuju kroz gzip za dekompresiju.
3. I onda se upisuju na prvu particiju hard drajva sa dd.

Kreiranje praznih disk imidza

Da bi ste kreirali prazni disk imidz, koji ce se na primer koristiti kao disk za emulator, mogu se uzeti podaci iz /dev/zero. Da bi ste kreirali imidz od 10mb:

```
$ dd if=/dev/zero of=myimage bs=1024 count=10240
```

Pametnija alternativa je:

```
$ dd of=myimage bs=1024 count=0 seek=10240
```

Ovde ne upisujemo nista, cak ni nule, samo smo zatrazili 10mb u fajlu i zatvorili ga. Rezultat je rastavljeni fajl koji je potpuno koji zadrzi 10mb nula, ali to ne zauzima nikakav prostor na disku. [ls -l](#) ce prijaviti 10mb, dok [du](#) i [df](#) ce prijaviti 0. Kada se fajl upise u njega, bilo kao emulator disk ili loopback uredjaj, Linux ce dodeliti prostor na disku za podatke. [ls](#) ce i dalje prikazivati 10mb, dok ce se [du](#) pribliziti 10mb.

Za swap imidze, gde je vaznije da se sacuvaju podaci nego da se sacuva prostor na disku, ne-rasclanjeni fajl je bolji.

Izvor: <http://wiki.linuxquestions.org/wiki/Dd>

~Dalibor Gudzic@soxxx