

# dd

The `dd` program is a surprisingly powerful one. It can be used for everything from copying a disk partition to converting ASCII files to EBCDIC.



# dd conversions

```
ascii    # from EBCDIC to ASCII
ebcdic   # from ASCII to EBCDIC
ibm      # from ASCII to alternated EBCDIC
lcase    # change upper case to lower case
ucase    # change lower case to upper case
swab     # swap every pair of input bytes
```



# dd copying

Copying raw block-structured devices is quite easy:

```
dd if=/dev/hda1 of=/dev/hda2
```



## dd other tricks

You can also remove bytes from the beginning or the end of a file:

```
dd bs=1 skip=4000      # skip over the first 4000 characters
```

```
dd count=10000 bs=1   # copy only the first 10000 characters
```



# csplit

`csplit` (context split) lets you split a file by specifying a pattern for each split point.

```
csplit /PATTERN/ /PATTERN/|COUNT
```



# csplit

For instance, say you want to split the `/etc/termcap` file into 1200 separate definitions.

You can easily do this with the single line:

```
csplit /etc/termcap '/^[a-z]/' '{*}' # the second item is a repeat count
```



# csplit

You can then get 1300+ files, such as

```
[langley@sophie tmp]$ head -1000 xx*
```

```
==> xx01 <==
```

```
dumb|80-column dumb tty:\
```

```
  :am:\
```

```
  :co#80:\
```

```
  :bl=^G:cr=^M:do=^J:sf=^J:
```

```
==> xx02 <==
```

```
unknown|unknown terminal type:\
```

```
  :gn:tc=dumb:
```

```
==> xx03 <==
```

```
lpr|printer|line printer:\
```

```
  :bs:hc:os:\
```

```
  :co#132:li#66:\
```

```
  :bl=^G:cr=^M:do=^J:ff=^L:le=^H:sf=^J:
```



```
==> xx04 <==
glasstty|classic glass tty interpreting ASCII control characters:\
      :am:bs:\
      :co#80:\
      :bl=^G:cl=^L:cr=^M:do=^J:kd=^J:kl=^H:le=^H:nw=^M^J:ta=^I:
```

```
==> xx05 <==
vanilla:\
      :bs:\
      :bl=^G:cr=^M:do=^J:sf=^J:
```

```
==> xx06 <==
ansi+local1:\
      :do=\E[B:le=\E[D:nd=\E[C:up=\E[A:
```

```
==> xx07 <==
ansi+local:\
      :DO=\E[%dB:LE=\E[%dD:RI=\E[%dC:UP=\E[%dA:tc=ansi+local1:
```

```
==> xx08 <==
ansi+tabs:\
      :bt=\E[Z:ct=\E[2g:st=\EH:ta=^I:
```





```
==> xx09 <==
```

```
ansi+inittabs:\
    :it#8:tc=ansi+tabs:
```

```
==> xx10 <==
```

```
ansi+erase:\
    :cd=\E[J:ce=\E[K:cl=\E[H\E[J:
```

```
==> xx100 <==
```

```
arm100|arm100-am|Arm(RiscPC) ncurses compatible (for 640x480):\
    :am:ms:ut:xn:xo:\
    :co#80:it#8:li#30:\
    :@8=\E[M:DO=\E[%dB:K1=\E[q:K2=\E[r:K3=\E[s:K4=\E[p:K5=\E[n:\
    :LE=\E[%dD:RA=\E[?7l:RI=\E[%dC:SA=\E[?7h:UP=\E[%dA:\
    :ac=` `aaffggjjkklmmnnooppqrrssttuuvvwxyz{|}~::~\
    :ae=^O:as=^N:bl=^G:cb=\E[lK:cd=\E[J:ce=\E[K:cl=\E[H\E[J:\
    :cm=\E[%i%d;%dH:cr=^M:cs=\E[%i%d;%dr:ct=\E[3g:do=^J:\
    :eA=\E(B\E)0:ho=\E[H:k0=\E[y:k1=\E[P:k2=\E[Q:k3=\E[R:\
    :k4=\E[S:k5=\E[t:k6=\E[u:k7=\E[v:k8=\E[l:k9=\E[w:k;=\E[x:\
    :kb=^H:kd=\E[B:ke=\E[?1l\E>:kl=\E[D:kr=\E[C:ks=\E[?1h\E=:\
    :ku=\E[A:le=^H:mb=\E[5m:md=\E[lm:me=\E[m\017:mk=\E[8m:\
    :mr=\E[7m:nd=\E[C:rc=\E8:\
    :rs=\E>\E[?3l\E[?4l\E[?5l\E[?7h\E[?8h:\
    :..sa=\E[0%?%p1%p6%|%t;1%;%?%p2%t;4%;%?%p1%p3%|%t;7%;%?%p4%t;5%;%?%p7%t;8%;m%?%p
    :sc=\E7:se=\E[m:sf=^J:so=\E[7m:sr=\EM:st=\EH:ta=^I:ue=\E[m:\
```



```
:up=\E[A:us=\E[4m:tc=ecma+sgr:tc=klone+color:
```

```
==> xx1000 <==
```

```
ncr260wy60wpp|NCR 2900_260 wyse 60 wide mode:\
```

```
:co#132:\
```

```
:cm=\Ea%i%dR%dC:\
```

```
:is=\Ee6\E~4\E+\Ed/\Ee1\Ed*\Er\EO\E'1\E';\E'@\E~!\E"\Ee4\Ex@\E'9\Ee7:\
```

```
:rs=\Ee6\E~4\E+\Ed/\Ee1\Ed*\Er\EO\E'1\E';\E'@\E~!\E"\Ee4\Ex@\E'9\Ee7:\
```

```
:tc=ncr260wy60pp:
```



# csplit

Alternatively, you can also just specify arbitrary line numbers:

```
% csplit /etc/termcap 4 10 110  
110  
107  
5023  
734959
```



# Portable anymaps

Way back, there was a package called “PBM”, the Portable BitMap package. It allowed you to convert files of many different graphic types to other types, and it allowed you to manipulate these files from the command line.

For instance, when I did the window dumps for some of the lectures, I used this package something along these lines:

```
sleep 10 ; xwd > /tmp/xwd.1
```



```
xwdtopnm < /tmp/xwd.1 | pnmtopng > /tmp/rdesktop01.png
```



# The conversions

# PNM conversions

```
giftopnm      # GIF to pnm
rasttopnm     # Sun rasterfile to pnm
tifftopnm     # tiff to pnm
xwdtopnm      # X window dump format to pnm
```

```
pnmtotiff     # pnm to tiff
pnmtoxwd      # pnm to xwd
pnmtorast     # pnm to Sun rasterfile
pnmtops       # convert to postscript
```

# PPM conversions

```
gouldtoppm    # Gould scanner file to ppm
ilbmtoppm     # Amiga format to ppm
```



```
ppmtogif    # gif to ppm  
pgmtoppm   # convert pgm to PPM (convert grayscale to color)
```



# Manipulations

```
ppmdither # dither a file (reduce the number of colors used)
ppmdepth # change the number of planes in an image
ppmquant # reduce the number of colors used in a file
ppmquantall # run ppmquant over many files so they share common colormap
ppmforge # create fractal forgeries of clouds, stars, and planets
pnmcrop # crop borders from an image
pnmcut # extract arbitrary rectangle from an image
pnmarith # add, subtract, multiply, abs(diff) two images
pnm enlarge # enlarge an image by integer factor
pnmscale # arbitrary resize an image
pbmreduce # reduce image by integer factor
pnmsmooth # smooth a picture (useful after resizing)
pnmfile # describe file's image characteristics
pnmflip # flip an image
pnm gamma
```





# ppmforge fun



Image generated with ppmforge

```
ppmforge -stars 100 -night -width 200 -height 200 | pnmtopng > /tmp/xyz.png
```

