The 'dd', 'ddrescue', 'dcfldd' and 'dc3dd' Commands

dd Copy a file, converting and formatting according to the operands.

ddrescue - Data recovery tool. Copies data from one file or block device to another, trying hard to rescue data in case of read errors. It does not write zeros to the output when it finds bad sectors in the input, and does not truncate the output file unless asked to. So, every time you run it on the same output file, it tries to fill in the gaps without wiping out the data already rescued.

dcfldd is an enhanced version of dd with features useful for forensics and security including:

- On-the-fly hashing of the transmitted data.
- Progress bar of how much data has already been sent.
- Wiping of disks with known patterns.
- Verification that the image is identical to the original drive, bit-for-bit.
- Simultaneous output to more than one file/disk is possible.
- The output can be split into multiple files.
- Logs and data can be piped into external applications.

The program only produces raw image files.

dc3dd and dcfldd programs are based on slightly different code bases and have different feature sets. **dcfldd is a fork** of GNU dd, whereas **dc3dd is a patch** to the current version of dd. This means that dc3dd will be updated every time GNU dd is updated, whereas dcfldd has its own release schedule. Certain features added to GNU dd after dcfldd forked, such as direct input/output mode, are not found in dcfldd.

On the other hand, dcfldd supports more hashing algorithms than dc3dd, allows the user greater control over how hashes are displayed, supports wiping output files with random patterns, and is supported on the Cygwin platform.
dc3dd has added features for computer forensics:

- possibility to write errors to a file
- group errors in the error log

The basic command is structured as follows:
dd if=<s*ource*> of=<*target*> bs=<byte size>(usually some power of 2, not less than 512 bytes(ie, 512, 1024, 2048, 4096, 8192, 16384, but can be any number.) skip= seek= conv=<conversion>.
Source is the data being read. (defaults to keyboard)
Target is where the data gets written. (defaults to screen)

***If you mess up, and accidentally reverse the source and target, you can destroy a lot of data. dd is the data destroyer!!

dd Usage
dd.....

Progress information using dd (slow)
dd if=/dev/zero | pv | dd of=/dev/null count=10MB
progress information in dcfldd, and dc3dd is built in.

Checksum a file: dd if=/dev/cdrom | md5sum

Copy an ISO

From Linux there's a very easy way to create a bootable memory stick from an ISO image -- and this should work for *any* OS. Assuming the memory stick is `/dev/sdb` and the image is `/home/username/Downloads/system.iso`.
sudo if=/dev/sdb of=/home/username/Downloads/system.iso bs=4096

Copy one hard disk partition to another hard disk:

dd if=/dev/sda2 of=/dev/sdb2 conv=notrunc,noerror

Copy one hard disk partition to another hard disk:

dd if=/dev/sda2 of=/dev/sdb2 bs=4096 conv=notrunc,noerror
If sdb2 doesn't exist, dd will start at the beginning of the disk, and create it.

Cloning an entire hard disk:

sudo dd if=/dev/sdb of=/dev/sdc bs=512 conv=notrunc,noerror
or
sudo dcfldd if=/dev/sdb of=/dev/sdc bs=512 conv=notrunc,noerror
In this example, sdb is the source, sdc is the target.
***** Do not reverse the intended source and target**.
notrunc means do not truncate.
noerror means to keep going if there is an error. Normally dd stops at any error and reports.

Testing a hard drive

One way to test if a hard drive is working reliably is to use it as the source drive for the dd command; the output file can be NULL. If the drive is not working it will give dd errors. Target drives need to be really messed up before they give an error in dd.

Wipe a hard drive of all data (boot from a live cd to do this)

dd if=/dev/zero of=/dev/sda conv=notrunc

To view your virtual memory : sudo dd if=/proc/kcore | hexdump -C | less

What filesystems are installed: sudo dd if=/proc/filesystems | hexdump -C | less

All loaded modules: sudo dd if=/proc/kallsyms | hexdump -C | less

Interrupt Table: sudo dd if=/proc/interrupts | hexdump -C | less

How many seconds has the system been up:

dd if=/proc/uptime | hexdump -C | less

Partitions and sizes in kb: sudo dd if=/proc/partitions | hexdump -C | less

Memory statistics: sudo dd if=/proc/meminfo | hexdump -C | less

Convert a file to all uppercase: sudo dd if=filename of=filename conv=ucase

Write random data over a file before deleting it:

First do an '/bin/ls -l' to find the filesize. In this case it is 3769
ls -l afile
-rw------- ... 3769 Nov 2 13:41 <filename>
dd if=/dev/urandom of=afile bs=3769 count=1 conv=notrunc
This will write random characters over the entire file.