

Simple Glossary

Jeremy Sanders

October 2, 2011

These definitions are not exact as I've tried to simplify the concepts. They may be incorrect in some circumstances. They are definitely *not* complete. Please see e.g. <http://www.webopedia.com/> for more comprehensive definitions. Please send suggestions (and definitions!) to me.

1. *ADS* - Astronomical Data Service - NASA's database to find all published papers and recent preprints. Go to http://ukads.nottingham.ac.uk/abstract/_service.html
2. *Algorithm* - A set of instructions on how to solve a problem - usually programmed into the computer in the form program(!)
3. *Application* - Another word for a program, typically a program used by a normal user.
4. *awk* - One of the original scripting languages - useful for short operations.
5. *astro-ph* - The astrophysics section of arXiv.org. Probably the one you should read.
6. *Backup* - Storing another copy of data to make sure it doesn't accidentally get lost. See tar.
7. *Bash* - The GNU Bourne-Again SHell - a good shell for programming.
8. *Bit* - A binary digit - Either a 1 or a 0. n bits can hold 2^n different numbers.
9. *Bitmap* - An image consisting of a set of pixels or dots, each capable of different values or colours. Observation astronomical data often consists of bitmaps stored in a FITS file. Unlike Vector Graphics, bitmaps are often "jaggy" when scaled.
10. *Byte* - 8 bits of information - typically holds value from 0 to 255 ($2^8 - 1$). One byte can store one character in English (but some programs use Unicode to store text, which may require more than one byte per character).
11. *C* - The C programming language (successor to BCPL) is widely used outside science, but is useful for scientific programming. Most of UNIX is written in C.
12. *C++* - C++ is a development of the C programming language, but with Object-Oriented and Generic Programming features (I won't define those for you). The language is widely used outside science, but not so widely in science. It is famed for its steep learning curve (which doesn't have to be so steep) and its power.
13. *Client* - A computer or a program which is being given information by another computer or program (the Server).
14. *Command* - Some text to tell the computer to do something - may refer to a command in a programming language, an executable, something interpreted by a shell, or a instruction to another program (e.g. IRAF).
15. *Compiler* - A program which takes text in a particular programming language, and produces either an Executable or an Object File.

16. *Configuration file* - A file containing commands to modify the behaviour of a program. See Dot Files.
17. *CPU* - Central Processing Unit - the part of the computer which executes a program. There are different sorts of CPUs (e.g. Sparc, Alpha, i386). Typically programs written to run on one CPU do not run on another sort (plenty of exceptions here, though).
18. *csh* - The C Shell - A C-like shell. Don't use this.
19. *Data* - Data consists of a series of bytes on most machines. Data in a file can either be stored as it is in the computer's memory (raw or unencoded), or stored in text format (formatted).
20. *Data reduction* - Taking unprocessed astronomical data and getting it in a more understandable form (e.g. calibrating it).
21. *Device* - Something attached to your computer - can be internal to the machine like a disk, or external like a mouse, or a logical device-like entity. Devices include disks, tape drives, mice, keyboards, terminals. In UNIX devices often have an associated device file in the /dev directory which represents the device and can be used to operate it. Special devices include /dev/null which throws everything away, and /dev/zero which provides infinite numbers of zeros for you enjoyment.
22. *Directory* - A name for a set of files and directories. Directories can contain other directories (This can make a "tree" of directories). Directories can also refer to different devices to their parent - hard disks, floppy disks, network drives. All directories in UNIX have the root directory ("/") back in their family tree.
23. *Disk* - See Hard Disk or Floppy Disk (also others). A disk is usually a permanent (hopefully!) storage device.
24. *Dot File* - A file with a name starting with a dot. These files are hidden from a normal ls (try ls -a instead). Dot files in your home directory are typically configuration files. The other common dot files are "." and ".." which exist in every directory, meaning the current directory and the parent directory.
25. *Editor* - A program for editing (usually) a text file. Examples are emacs, vi, ed and nedit. People are very opinionated about which is the best editor! (Search on the internet for Editor-Wars).
26. *emacs* - A very popular and powerful editor. You can do anything in emacs but it may not be easy to find out how to do it. The editor can be extended using Lisp code to do any sort of text manipulation you like.
27. *Executable* - File stored on a disk which can be run on the computer. Executables are marked with the executable attribute (See the chmod command). Executables are stored in a particular format (e.g. ELF), and consist of headers plus machine code.
28. *Extension (of a filename)* - Conventional ending for a file name (usually dot-something) to tell you what sort of file it is (e.g. .a - library, .c - C source code, .h - C source header code, .html - Web page, .pl - Perl script, .sh - sh shell script, .tex - \LaTeX or \TeX input file...)
29. *File* - Where data are permanently stored on "normal" computers. On UNIX consists of a sequence of bytes up to a certain length. Files may contain normal ASCII text or a random sequence of bytes which can be interpreted by a program. Files are stored in Directories on Disks.
30. *Finger* - See who's on a computer, e.g. finger @cass00.
31. *FITS file* - Standard astronomical file format. Usually used for observational data but also suitable for theoretical work. FITS files consist of a series of "extensions" and a set of headers.

32. *Floating point* - How real numbers are stored on a computer. Numbers are stored as a sign (s , +1 or -1), a mantissa (m , $0 \leq m \leq 1$), and an exponent (e) in a similar form to $s \times m \times X^e$, e.g. 512.43 in base 10 would be $(+1) \times 0.51243 \times 10^3$. The number of bytes used for each part is larger for doubles than single floats, meaning operations are more accurate. Floating point works well but you can get big problems if you aren't careful (e.g. don't subtract large numbers from each other, don't compare floating point numbers to be an exact number). There are long documents on floating point numbers (see the IEEE standard). *See for very useful information if you're doing numerical work* <http://docs.sun.com/db?p=/doc/800-7895>
33. *Floppy* - A small transferable disk holding about 1.4 megabytes.
34. *Fork* - Splitting a process into two - a Parent and a Child. This is the way other processes are run on UNIX.
35. *FORTTRAN* - FORMula TRANslation - The original scientific programming language. FORTRAN 77 (an earlier dialect) lacks modern features (such as dynamic memory allocation) but is in wide-spread use. Fortran 90/95 is a much better version, but is less widely used.
36. *Home directory* - A personal directory belonging to a particular user. Home directories are normally given the name /home/username.
37. *FTP* - File Transfer Protocol - a method of transferring files between different computers on the internet. It's to be avoided if you need to type a password to get the file as the file goes over the network in Plain Text, which can be "sniffed". Use scp or sftp instead if possible.
38. *Gigabyte* - 1024 Megabytes, 2^{30} bytes. Still a fair chunk of information.
39. *GIF* - A bitmap image format. Use a PNG instead if you can.
40. *GPL* - GNU Public Licence - A licence applied to a program to specify it can be distributed and modified to/by anyone, but if a modified version is distributed, the source must be distributed too (this is the general idea). See <http://www.gnu.org/>
41. *GNU* - GNU is Not Unix - A set of programs written by the Free Software Foundation to provide a Free (as in libre) UNIX framework. GNU is a large component of a Linux distribution. GNU software include emacs and gcc. See <http://www.gnu.org/>. GNU software is licenced under the GPL.
42. *grep* - The grep command lets you find lines in file(s) with particular text or regular expressions.
43. *Grid* - A form of Parallel computing, allowing jobs to be submitted to a set of machines.
44. *GSL* - GNU Scientific Library - Free library for scientific and numerical work, including integration and equation solving.
45. *Hard Disk* - A large non-portable disk attached to (or in) a computer capable of holding a large quantity of data.
46. *HTML* - HyperText Markup Language - The language by which Web Pages are described. HTML is simple text with added codes to identify elements.
47. *HTTP* - HyperText Transfer Protocol - The protocol by which Web Pages are transmitted. Other things can use this protocol too.
48. *HTTPS* - A secure version of HTTP which encrypts data.
49. *IDL* - A proprietary programming environment with strong numerical and graphing features. The down sides are that you need a licence to run this on all machines, and the syntax is particularly poor. It is also quite slow for some programming cases.

50. *info* - A system like man but providing more information about GNU programs, like emacs. Run info to try it.
51. *Internet* - The global network of computers running TCP/IP, which the IoA is connected to.
52. *IP address* - Internet Protocol address - each computer on the internet (with exceptions) has a unique 4-byte number identifying it, which is called an IP address. A name like www.bbc.co.uk is converted to an IP address by a DNS (Directory Name Service) server. This is useful as you don't need to remember lots of numbers like 131.111.23.12
53. *IRAF* - The Image Reduction and Analysis Facility - A standard astronomical package for data reduction.
54. *JANET* - The Joint Academic NETwork - The network linking universities in the UK together. It is linked to the Internet.
55. *Job* - Using a program with particular inputs. e.g. submitting a job.
56. *JPEG, JPG* - The JPEG format is a lossy (in that it loses data from the original, but can look very good) bitmap image format. It is widely used on the internet. The advantage of this format is that it doesn't take much space up.
57. *K, Kilobyte* - A kilobyte, or 1024 bytes (2^{10}). Anyone claiming this is 1000 bytes is wrong.
58. *Kernel* - See UNIX
59. *Killing* - Stopping a process permanently. Processes can be killed "nicely" with kill PID, or hastily with kill -9 PID.
60. *Library* - A set of Object Files combined together into one file. Libraries are either static or dynamic. Static libraries are linked into an executable when it is made (.a extension). Dynamic libraries are linked when the executable is run. (.so extension). Using static libraries is simpler than using dynamic ones, but they need more space in the executable. Libraries are sometimes standard (come with UNIX) like the C-runtime library, or optional (like gtk). Lots more could be said here.
61. *Lisp* - A computer language often used for text processing. Used as the internal language in emacs.
62. *Machine* - A commonly used word meaning a computer.
63. *Maple* - An proprietary mathematics program. Maple can do numerical work and symbolic work (like doing integrals). Might be useful if you're a paper and pencil theorist, but cannot be relied upon to do all your work. See Mathematica.
64. *man* - The UNIX manual system. Very useful. Type man man.
65. *Mathematica* - The original proprietary mathematics system (see Maple). Maple and Mathematica claim to do similar things, but with varying success.
66. *Megabyte* - 1024 kilobytes, or 1048576 bytes (2^{20}).
67. *Memory* - See RAM
68. *Mount* - The act of making a disk or network drive available for use to the users of a system. Undoing this is called unmount.
69. *Mount point* - A disk or network drive is "mounted" into the file system at a mount point, which looks like a directory, e.g. /mnt/cdrom.
70. *N-Body* - A way of simulating something by considering it as separate bodies, linked by gravity.
71. *NAG* - Numerical Algorithms Group - A proprietary computer library for numerical work.

72. *Network* - A system for allowing computers to communicate using the transmission of packets of information. They do this using a Protocol like TCP/IP. Networks can be linked together (e.g. JANET, Internet)
73. *Newsgroup* - A newsgroup is a discussion forum. There are many newsgroups, for many subjects. Newsreading programs include mozilla, trn and pine.
74. *NFS* - Network Filing System - A system for accessing files over the network as if they were on your local machine. Often used for home directories.
75. *Object file* - Compiled source code which can be Linked together with other Object Files or Libraries to produce an executable or library. Object Files typically have an .o extension.
76. *Parallel* - Running code on more than one machine or CPU in a machine at once.
77. *Perl* - A very powerful scripting language but has arcane syntax. Code can be very hard to read once written.
78. *PID* - Process ID - Each process has an ID number (PID) (normally 0 to $2^{15} - 1$) which can be used to kill it or stop it.
79. *Ping* - See whether a computer is on the network (e.g. ping cass00).
80. *Pipe* - A connection between the output of one program and the input of another. e.g. ls | more, takes the output of "ls" (a list of files), and displays it one screen at a time with the more program.
81. *PNG* - Portable Network Graphics - An lossless bitmap image format. Should be used instead of GIF which has patent problems.
82. *Process* - When a program is started it becomes a process. You can run more than one copy of a program and you get more than one process. Processes are started from a shell (or another process) and can be killed using the kill or top shell commands.
83. *Program* - Refers to a set of instructions which can be "run" on a computer. Programs are stored in a file (an Executable) and are started from a shell (or indirectly from other processes). The word program can also refer to the source code of the program.
84. *Programming language* - A standard which specifies how (sort of) human readable text is run on a computer. Programming languages vary quite a lot in style (e.g. Procedural - made up of blocks run from each other, Object-Oriented - program made of a set of communicating isolated objects, Functional - everything is a function!, ...), and make some things easier than others. There are low-level languages which give you lots of control over the computer and things run quickly, or higher level where you can get the computer to do lots with little work but have less total control, however these categories aren't fixed. Examples include C, FORTRAN, Perl, Python, BASIC, Pascal, FORTH...).
85. *Protocol* - An accepted standard for transmission of data, like TCP, HTTP, FTP.
86. *Python* - A scripting language. Designed for simplicity and clarity. Less widely used than Perl, however, but is a good choice for large programs.
87. *Quota* - The maximum amount of space you are allowed to use on a disk. The quota can have a soft-limit (which may be exceeded for some time), and a hard-limit (which may not be exceeded).
88. *RAM* - Random Access Memory (typically just called memory) - fast memory in a computer which holds currently running programs and data being processed. When the computer is switched off anything in memory is lost.
89. *Recursive* - See Recursive.
90. *Redirect* - Send the output of one thing to another (e.g. redirect to a file).

91. *Regular expression* - A useful way of getting a program to match particular bits of text. Often used in scripting languages to process data files.
92. *Scanner* - A device for transferring text or images from paper to a computer.
93. *Script* - A set of shell commands in an executable file.
94. *Scripting languages* - A computer language designed for ease of writing. Scripting languages are often used to process text files or run external programs. Examples include Perl, Python, tcl and awk.
95. *SCP* - Secure Copy - Sends files over SSH
96. *SCSI* - A way of talking to devices on a computer like disks, tape drives or scanners.
97. *sed* - The Stream EDitor - A program for simple manipulation of the contents of files.
98. *Server* - A machine or program which is giving data to another machine or program (the Client).
99. *SFTP* - Secure FTP - Like SCP, but more like FTP.
100. *sh* - The Bourne Shell - very basic but standard shell.
101. *Shell* - A program which takes instructions (either by someone typing them in or from a shell script file), and follows them, including running programs. Shells include, sh (The Bourne shell), csh (The C shell), bash (The Bourne-Again shell - bad pun!), tcsh (An extended C shell). Shells vary in their features and ease of use. Some shells are horrible (e.g. csh, tcsh) :-).
102. *Socket* - A communication mechanism between programs. Data can be sent over the network, making a Network Socket. This is the primary mechanism of the Internet.
103. *Source code* - Human readable (hopefully!) program text. Source code can either be compiled and then stored as an executable on the computer, or interpreted by an "interpreter" to run directly (e.g. Python, shell, BASIC), though some things are between compiler and interpreter.
104. *SPH* - Smooth Particle Hydrodynamics - A way of simulating gas.
105. *SSH* - Secure SHell - A way to access another machine. Data is sent encrypted between the machines making it hard to grab information like passwords.
106. *Submit* - To send something to another system - e.g. the grid engine or a printer.
107. *Tape* - A tape is used to back up data. DO THIS! A tape looks like an audio cassette. Tapes come in different sizes and different standards.
108. *tar* - A UNIX command used to combine (and optionally compress) files together.
109. *tcl* - Tool Command Language - An older scripting language. Less powerful and harder to use than modern languages like Perl or Python.
110. *TCP/IP* - Transmission Control Protocol / Internet Protocol - The Protocols the Internet uses to transmit data.
111. *tcsh* - The exTended C-Shell - the default astronomy shell. Not very good for programming.
112. *Telnet* - An old protocol for connecting to a different machine. Don't use it as it sends passwords in plain text. Use SSH instead.
113. *Terminal* - A terminal is a device (or application) which allows you to type commands into a Shell. See XTerm.

114. *UNIX* - The environment (Operating System) your computer is running, providing the framework for your programs. There are lots of different sorts of UNIX which have lots of subtle differences (e.g. Linux, Solaris, Tru64, HPUX...). UNIX is made up of a Kernel which provides common interfaces to the hardware, the Runtime Library which bridges applications (or programs you write) and the Kernel, and the Shell which lets you run programs. UNIX is a pretty amorphous concept and also consists of the huge suit of programs you can run, e.g. sed, ls, rm, ping...
115. *Vector graphics* - An image consisting of geometrical shapes, like lines, circles and text. Vector graphics make suitable figures for papers and scale well. See Bitmap also.
116. *Web* - The World Wide Web - often known as the "internet", though it is only part.
117. *Web comic* - How to spend your PhD doing very little.
118. *Web page* - A page intended to be viewed with a web browser.
119. *Window Manager* - The program which controls the windows (the boxes around the programs) on your screen. Common examples are fvwm, twm, sawfish.
120. *Windows* - Do not confuse this Microsoft system with the superior X-Windows System.
121. *windows* - The rectangles on your screen programs run in.
122. *XTerm* - The X-Terminal application allows you to type commands into your shell and run programs. This is the most useful X application. Many versions of this exist (e.g. kterm, gnome-terminal, rxvt).
123. *X-Windows (or X)* - The X-Windows System - The system UNIX machines typically use to display graphical programs. X-Windows uses a Window Manager to control the windows on the desktop. X-Windows can also run a program over the network from a remote machine (very useful!). This can be done with SSH.