chaos Filesystem Hierarchy Standard

$Id: filesystem-hierarchy.sgml,v 1.2 2000/07/02 23:01:38 plundis Exp $

This is the chaos Filesystem Hierarchy Standard. In here, we try to document the filsystem hierarchy used by the chaos Operating System. See the section for more information about this.

PrologueThis file contains the FHS for the chaos operating system. It should be noted that this is not the only way to structure things in a chaos system, but it is the preferred and programs that want to call themselves ’Designed for chaos’ must obey to these standards. More specifically, a program that meets our criterias must also have no hard-coded paths at all; all references to those directories must use our defined paths, which are defined in the include file ’chaos.h’ in the standard chaos library. (More about this in the chaos Programming Reference Manual)This standard does not try to be like everything else. The regular way of doing things does not necessarily have to be the best. In many cases, you’ll see this in that we have chosen a totally different approach than everything else. Remember that chaos is not just another boring UNIX clone; this is something totally different. Something new.If you feel something is wrong or missing, please contact us at one of the mail addresses listed at our . Flames go directly to /devices/character/null. :-)The FHSDirectory /configThis is where all the configuration files are stored. It does not contain any files, only directories. System configuration files are stored in /config/chaos. A program designed for chaos should put its system wide configuration files in ’/config/PROGRAM’, where PROGRAM is the name of the program. (I.e., if program foo wants to store configuration about bar, it should create the directory ’/config/foo’ and put the information in ’/config/foo/bar’) General preferences for a program should be put in a file called "preferences" in the program’s configuration directory. As a general rule, never put the program’s name in the name of the configuration file.Directory /config/chaosThis is where all the system configuration files are stored. An example of such a file is ’config/chaos/servers’ that contains a list of all the servers that should be started at boot time. (Servers started from the Multiboot bootloader (usually GRUB) does not have to be listed in this file)Directory /librariesThis is where all libraries (only shared; static are placed elsewhere) are stored. A library should generally have the name ’<name>-x.y.z’, so the foo library of version 1.2.3 would be called ’foo-1.2.3’. (There’s no need to indicate in the filename that this is a library since it is implicated by the path name)Directory /modulesHere, loadable modules used by some programs and libraries will be found (in appropriate subdirectories, of course). This is used to support, for example, data types (the ability for a program to read/write to data formats without knowing how they are stored).Directory /clientsThis is where all the binaries of the programs you have installed. For example, the binary for the program ’cluido’ is located at ’/clients/cluido/cluido’.Directory /systemThis is where operating system files are stored (except for configuration files). There should be no files in this directory; only subdirectories.Directory /system/serversHere, all the servers installed are stored. They are usually compressed with GNU zip. Most servers doesn’t have a configuration file, so the binary is just about it. (except for the documentation)Directory /system/kernelThis is where the kernel image should be stored. A normal gzipped storm kernel image is called /system/kernel/storm-xx.yy.zz.gz, for example.Directory /dataHere, the programs store can store arbitrary data. For example, the program "Quake" could store its data in the directory ’/data/quake’. Another example is ’/data/programming/c/headers’, where C header files is stored. (Yes, we know this is very different from what most of you are used to. This is on purpose. A non-programmer shouldn’t need to have any header files installed.)Directory /tempThis is where temporary files should be stored, for example information that a program doesn’t need for the moment. This directory is cleared at system startup.Directory /usersHere, the users’ home directories are stored. The home directory for user "foo" is located at ’/users/foo’.Virtual DirectoriesDirectory /devicesWhen a server is launched, it can create an entry in this directory if, and only if, it serves a block or character device. This entry can then be used for accessing the device the server manages (for example, the server ’keyboard’ creates the entry ’/devices/keyboard’ here, where it will put an entry for each keyboard found. Thus, ’/devices/keyboard/0’ can be used to access the first keyboard, and so on.Directory /servicesServers that don’t do hardware but more abstract things may want to create entries here. For example, the IP server might create ’/services/network/protocol/ipv4’. This virtual file can then be IOCTL:d (TODO: how should this work?) to get the PID of this process, so that a program that wants to do IP knows which PID to send the message toDirectory /kernelfsThis is the kernel virtual file system. In here, you can find some information about the status of the system; for example, process information (CPU usage, memory load, etc).