



C²S CONSULTING



GNU/Linux

Basic

Part 1



FREE
TECHNOLOGY
ACADEMY

Diarmuid Ó Briain

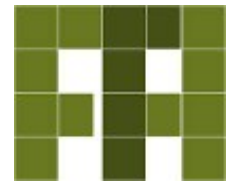
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com



This material is under a **Creative Commons BY-SA 3.0 license**
Full license: <http://creativecommons.org/licenses/by-sa/3.0/>

GNU/Linux Basic operating system



These slides are designed to follow the text book



Joaquín López Sánchez-Montañés
Sofia Belles Ramos
Roger Baig Viñas
Francesc Aulí Llinàs

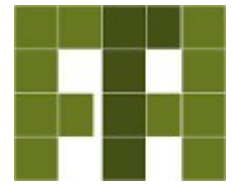
Jordi Serra i Ruiz (coordinator)
David Megías Jiménez (coordinator)
Jordi Mas (coordinator)

Diarmuid Ó Briain (Slide deck)



<http://www.ftacademy.org>

Table of Content



- **Free Software (a.k.a. Open Source)**
- **Basic concepts and commands**
- **Knoppix workshop**
- Debian installation workshop
- Basic configurations
- Daemons and runlevels
- Installation of applications
- Workshop on basic configurations
- X-Window architecture
- X-Window workshop



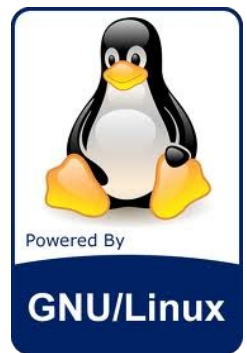
C²S CONSULTING

GNU/Linux

General Background



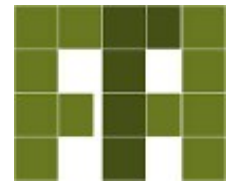
FREE
TECHNOLOGY
ACADEMY



Diarmuid Ó Briain

CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

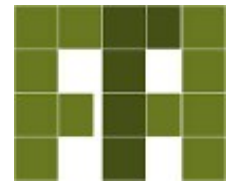


guarantees these four freedoms to the user:

- **use** for any purpose
- **study** and **adapt** to personal needs*
- **copy** and **share** with your neighbour
- **distribute** modified copies*

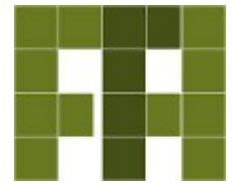
(*) access to the source code is a precondition for
this

Starting Points



- The main factors that hold back the **adoption of Free Technologies** are lack of awareness and knowledge.
- There is a **growing need** for ICT specialists, managers and decision makers knowledgeable on these subjects.
- Study programmes on Free Technology are still scarce and mainly local.
- The Universitat Oberta de Catalunya (UOC) runs a pioneering and successful **Master Programme on Free Software** since 2003.

Who's behind it



Founding partners:

- **Free Knowledge Institute** (FKI) – Netherlands
- **Universitat Oberta de Catalunya** (UOC) – Spain
- **Open Universiteit** (OUNL) - Netherlands
- **University of Agder** (UIA) – Norway

Associate Partners:

- **Free Software Foundation** (FSF)
- **Universidad Rey Juan Carlos** (URJC) – Spain

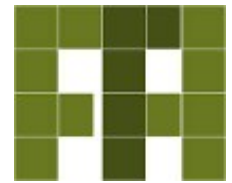
Established with funding from:

- **European Commission** (EC)
 - Lifelong Learning Programme (LLP) 2008-2010

Hosted by:

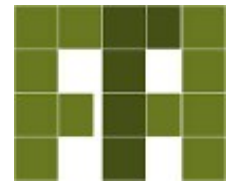
- **Free Knowledge Institute** (FKI)

What is FSF?



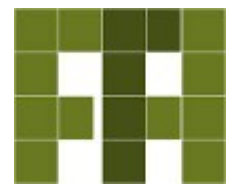
- Companies started to stop giving access to source code in 1970s.
- Richard Stallman
 - Massachusetts Institute of Technology
 - Stallman himself recalls how annoyed he was when he discovered that a company that had supplied a new printer for the laboratory where he was working did not want to give him the driver source code. All he wanted was to modify it so that the printer would automatically alert him when the paper was jammed. The company refused to provide him with the source code.
- Founded the Free Software Foundation (FSF) in October 1985 to support the free software movement, which promotes the universal freedom to create, distribute and modify computer software.





- **Freedom 0**
 - The freedom to be able to use the program for any purpose.
- **Freedom 1**
 - The freedom to study how the program works and adapt it.
 - Access to the source code is a pre-requisite for guaranteeing this freedom.
- **Freedom 2**
 - The freedom to freely distribute copies of the software.
- **Freedom 3**
 - The freedom to improve the program and to make improvements public, for the benefit of the entire community.
 - Access to the source code is also a pre-requisite for guaranteeing this freedom.

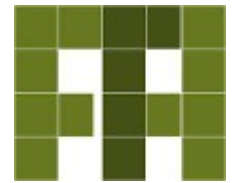
What is GNU?




- GNU recursive acronym for "GNU's Not Unix!"
- Unix-like, but differs from Unix by being free software and containing no Unix code.
- Based on the **GNU Hurd Kernel**.
- GNU General Public License (GPL)
 - Gives users freedom to share and change free software.
- GNU Lesser General Public License (LGPL)
 - Written for the GNU C Library to allow it to be linked with proprietary software.
- GNU Free Documentation License (FDL)
 - GPL for documentation.

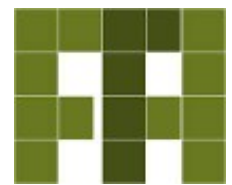


What is Linux?



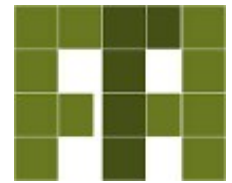
- **MINIX** 
 - Unix-like computer operating system based on a microkernel architecture created by **Andrew S. Tanenbaum** for educational purposes.
 - Micro-kernel technology for ease of study.
- **Linux**
 - **Linus Torvalds**, a student at the University of Helsinki, decided to create his own kernel for a new OS in August 1991.
 - UNIX for the Personal Computer inspired by MINIX.
 - Monolithic kernel.

Origin of Linux?



```
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Date: 25 Aug. 91 20:57:08 GMT
Organization: University of Helsinki
Hello everybody out there using minix.
I'm doing a (free) operating system (just a hobby,
won't be big and professional like gnu) for 386(486)
AT clones. This has been brewing since april, and
is starting to get ready. I'd like any feedback on
things people like/dislike in minix, as my OS res-
embles it somewhat (same physical layout of the
file-system (due to practical reasons) among other
things).
I've currently ported bash(1.08) and gcc(1.40), and
things seem to work.
This implies that I'll get something practical
within a few months, and I'd like to know what fea-
tures most people would want. Any suggestions are
welcome, but I won't promise I'll implement them :-)
```

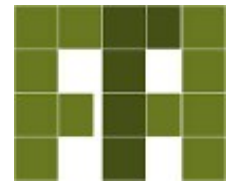
Linux Distributions



- Linux Distributions
 - Debian
 - Fedora



Linux Distributions

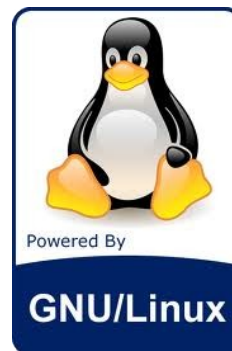


- The Linux Documentation Project
 - Manuals, HOWTOS, FAQs etc.
 - <http://www.tldp.org>
- linux.com
 - Repository of all things Linux
 - <http://www.linux.com>
- Slashdot
 - News for nurds, stuff that matters.
 - <http://slashdot.org>

The
Linux
Documentation
Project



Slashdot





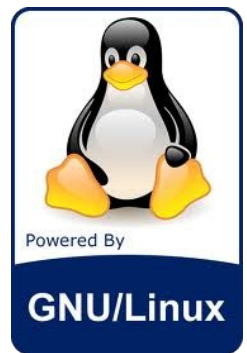
C²S CONSULTING

GNU Linux

Basic concepts and commands



FREE
TECHNOLOGY
ACADEMY



Diarmuid Ó Briain

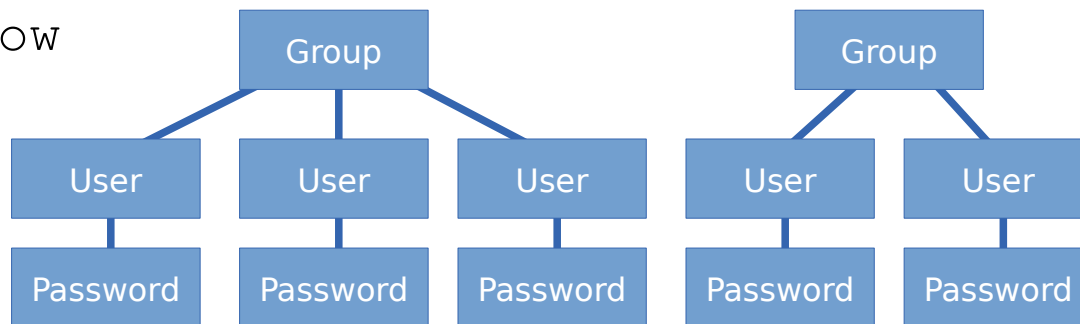
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

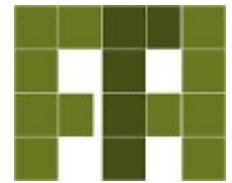
Users and groups



- Multi-user and multi-task.
 - More than one user can work on the system at the same time, carrying out one or more tasks simultaneously.
- System files:
 - information (name, directory, home etc.) of the user.
 - `/etc/passwd`
 - Information about user groups.
 - `/etc/group`
 - Users' ciphered passwords and configuration for validity, changing them etc.
 - `/etc/shadow`



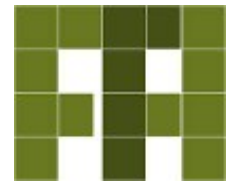
Users and groups – passwd file



- Username
- Password
- User ID
- Group ID
- Comments
- Home Directory
- Shell

```
$ sudo cat /etc/passwd | grep dobriain  
dobriain:x:1000:1000:Diarmuid O'Briain,,,:/home/dobriain:/bin/bash
```

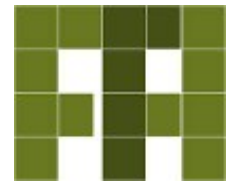
Users and groups – group file



- Group name
- Password
- Group ID
- User list

```
$ sudo cat /etc/group | grep dobriain
adm:x:4:dobriain
cdrom:x:24:dobriain
sudo:x:27:dobriain
dip:x:30:dobriain
plugdev:x:46:dobriain
lpadmin:x:112:dobriain
smbashare:x:118:dobriain
dobriain:x:1000:
```

Users and groups – shadow file



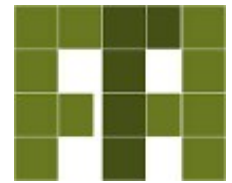
- Username.
- Ciphered password.
- Days, since 1st January 1970, until the password was last changed.
- Days to pass before the password can be changed.
- Days to pass before the password should be changed.
- Days before the password's expiry when the user will be notified to change it.
- Days that can pass following the password's expiry, before the user's account is blocked.
- Days, since 1st January 1970, since the account was blocked.
- Reserved field.

```
$ sudo cat /etc/shadow |grep dobriain  
dobriain:$6$RbCeK0dZ$xIGms0Egj8qY8qQ9Hwoo5wI3FcuGavSo5M
```

Users and groups – commands

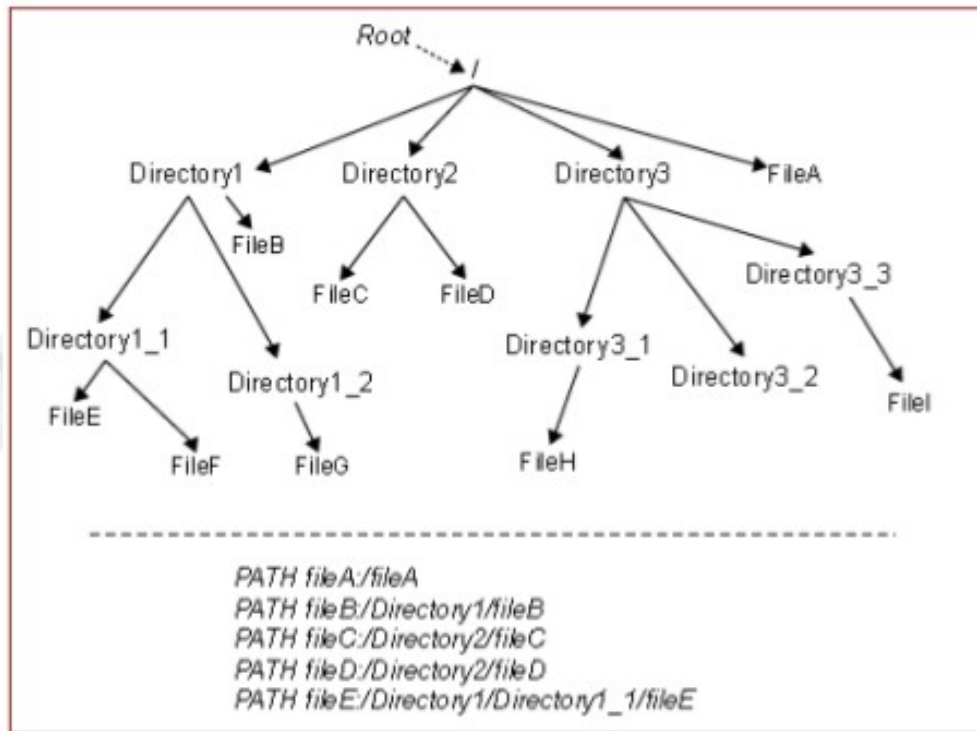
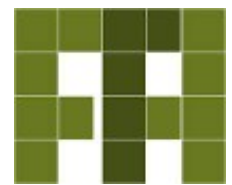


- `useradd` - create a new user or update default new user info.
- `usermod` - modify a user account.
- `chfn` - change real user name and information.
- `chsh` - change login shell.
- `userdel` - delete a user account and related files.
- `passwd` - change user password.
- `groupadd` - create a new group.
- `groupdel` - delete a group.
- `groupmod` - modify a group definition on the system.
- `gpaswd` - administer `/etc/group` and `/etc/gshadow`.



- ext2 - second extended filesystem.
 - The space in ext2 is split up into blocks. These blocks are grouped into block groups.
 - Inodes - Every file or directory is represented by an index node 'inode'.
 - The inode includes data about the size, permission, ownership, and location on disk of the file or directory.
 - Maximum file size 2 TiB with a maximum filesystem size of 32 TiB.
- ext3 - third extended filesystem.
 - ext3 adds the following features to ext2:
 - A journal (makes recovery quicker and less chance of corruption).
 - Online file system growth.
 - HTree indexing for larger directories.
- ext4 – fourth extended filesystem.
 - A follow up for ext3 and also a journaled filesystem with support for extents.

Filesystem

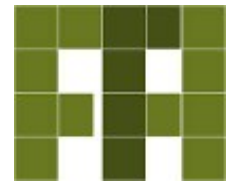


System directories



- `/` - Root directory
- `/bin/` – Essential commands
- `/boot/` – Boot loader and kernel images
- `/dev/` – Device driver access points
- `/etc/` – Host specific configuration
- `/home/` – User account directories
- `/lib/` – Libraries
- `/mnt/` – Mount point of removable media
- `/opt/` – Add-on application software packages
- `/proc/` – Kernel status routines
- `/root/` – Home directory for the root user
- `/sbin/` – Essential system commands
- `/tmp/` – Temporary Files
- `/usr/` – Secondary hierarchy
- `/var/` – Variable data

File System Abstractions

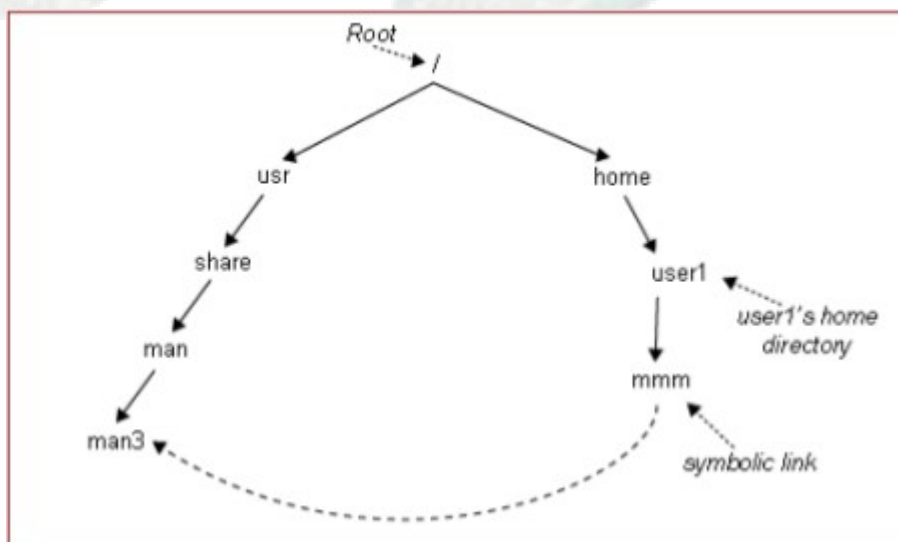


- `/proc`
 - a pseudo directory which reports the status of programs, device drivers, and kernel internal errors
- `/dev/xx0`
 - Direct access to device “0”
- `/dev/null`
 - Reports an End of File when read, and ignores input when written to
- `/dev/zeros`
 - Returns any number of characters requested as zero characters

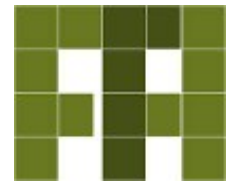
Links



- A link is a bridge to a file or directory belonging to the system.
- **Hard link** `ln /usr/share/man/man3 mmm`
 - A hard link can only be created between files or directories on the same unit due to the internal mechanism used to manage them.
- **Symbolic link** `ln -s /usr/share/man/man3 mmm`



File Permissions



SUID	SGID	Sticky	User			Group			Other		
			RD	WR	Ex	RD	WR	Ex	RD	WR	Ex

1 1 0

$$4+2+0=6$$

1 1 1

$$4+2+1=7$$

1 0 1

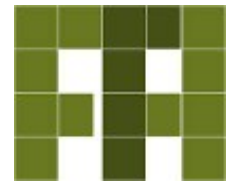
$$4+0+1=5$$

0 0 1

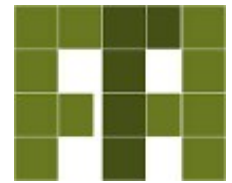
$$0+0+1=1$$

6751

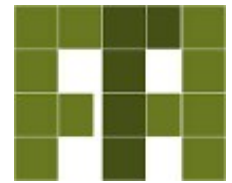
File Permissions



- SUID (Set User ID)
 - No effect on directories.
 - If set the file owner owns process no matter who executed the file.
- SGID (Set Group ID)
 - As per SUID.
 - For Directories, all new files created in the directory assume the ownership of the directory itself.
- Sticky
 - Flags the system to keep an image of the program in memory after it has completed running.
 - For directories it enhances security, regardless of file permissions, files in the directory can only be renamed or deleted by the directory owner or root.

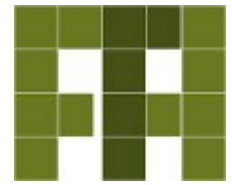


- SUID and Sticky bit special attributes associated with all files.
- Often in X Windows when you check the properties of any file (by right clicking on it and viewing its properties) has 3 special attributes besides the common read/ write/ execute rights for the owner/group/others.
- The 3 extra attributes are known as SUID, SGID and Sticky Bits.

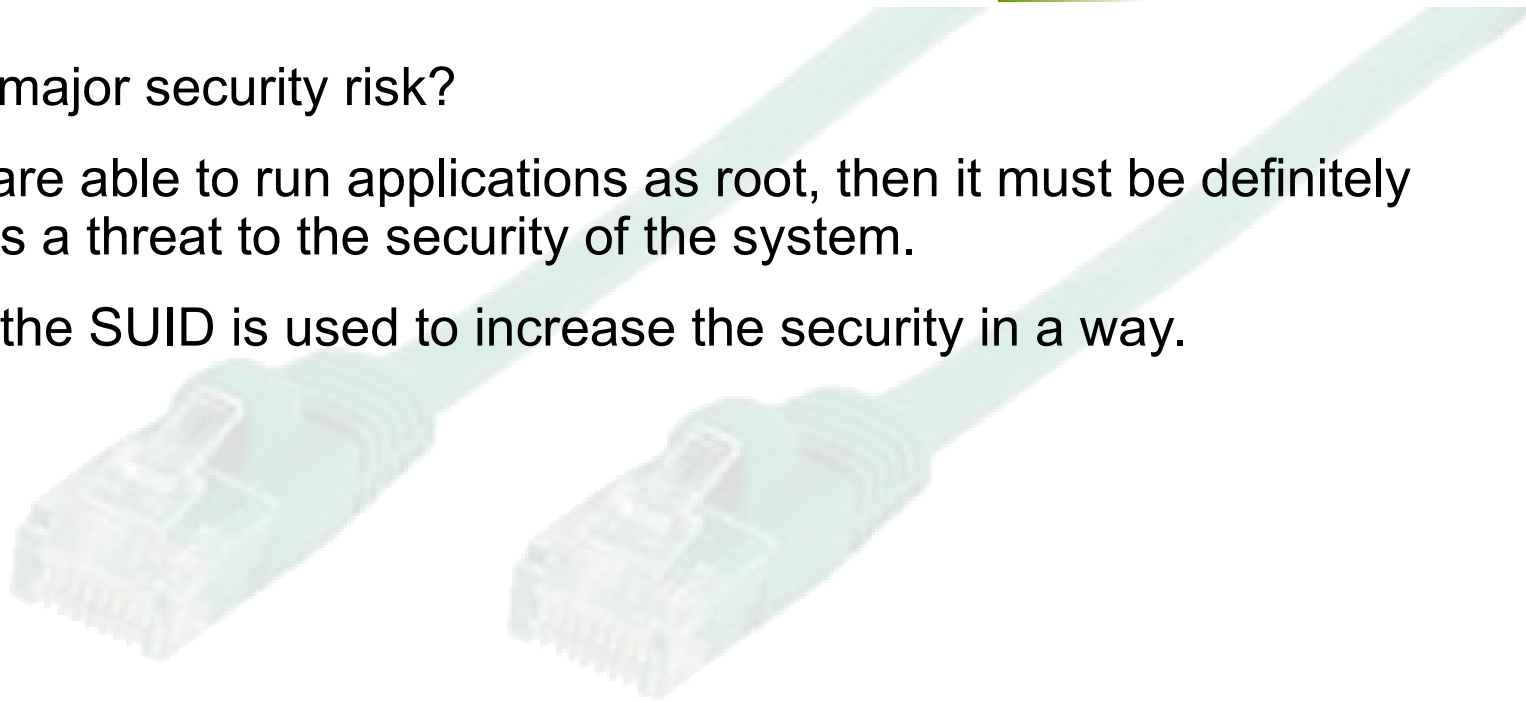


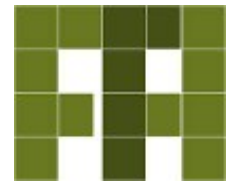
- Sometime you may face an error while trying to run any application stating that the application must be 'SUID root'.
- SUID stands for Set User ID.
- *This means that if the SUID bit is set for any application then your user ID would be set as that of the owner of application/ file rather than the current user, while running that application.*
- That means in case I have an application whose owner is ' root ' and it has its SUID bit set, then when I run this application as a normal user, that application would *still run as root*.
- Since the SUID bit tells Linux that the User ID root is set for this application and whenever this application executes it must execute as if root was executing it (since root owns this file).

SUID

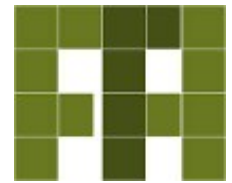


- Is this a major security risk?
- If users are able to run applications as root, then it must be definitely posing as a threat to the security of the system.
- Actually the SUID is used to increase the security in a way.





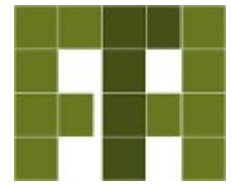
- Example application:
 - I have a few files that I modify through Linux and then before I shutdown Linux I have to transfer them to my Windows partition for further use there.
 - As a normal user I do not have write access to the Windows partitions that I have mounted. So I have to be the superuser to be able to write to that partition.
 - I have created a simple shell script that copies my files to the Windows partitions. This script was created by root user and the SUID bit was set. Access rights to this script have been given to all users.
 - Now whenever I want to copy my files I simply run this script. Even though I have logged in as a normal user, the SUID bit which is set causes this script to execute as if the root was executing it and it allows me to write to the Windows partitions.
 - Had the SUID bit not been set, I would have to type ' su ' at the prompt and get temporary superuser access to get write access to the Windows partitions.



- You may be thinking that since these applications would run as root they can also do harmful things to the OS.
- The concept behind SUID bit is that you as the superuser would be able to allow certain applications / scripts to be run by the users as if they were the superuser for the time being.
- What these application / scripts do when they execute should be completely known to you. Even though the users would be allowed to execute these programs as root they would be able to do ONLY THOSE things that these programs were designed to do initially by the superuser who created it.
- The user would not be able to modify that script in any way since he would not have write access to the script.
- He would only be having execute rights for that script.
- This way the users don't have to know the superuser password but can still use some facilities that are only available to the superuser.

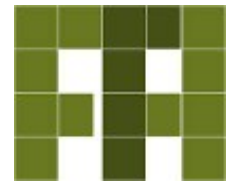


- SGID (Set Group ID) bit.
- Just like SUID, setting the SGID bit for a file sets your group ID to the file's group while the file is executing.
- It is really useful in case you have a real multi-user setup where users access each others files.
- The basic concept is the same as the SUID, the files whose SGID bit are set would be used as if they belong to that group rather than to that user alone.



- umask
 - Linux uses a three digit octal number to determine the file permission for newly created directories and files.
 - umask specifies the permissions you DO NOT want given by default to newly created files and directories.
 - Most utilities will specify a mode of 666 which allows any user to read or write to a file.
 - **Note** that the shell inherits its umask value from /etc/profile.

umask



- Establish the default permissions for new directories and files.

```
# umask 022

$ umask
0022

$ umask -S
u=rwx,g=rx,o=rx
```

- Example:
 - Here's what actually happens:

	Files	Directories
default mode	666	777
umask value	<u>-022</u>	<u>-022</u>
result	644	755

- This gives owner read, write and everyone else read permissions.

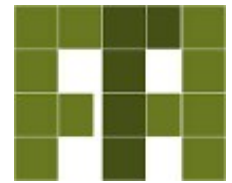
Permissions



```
$ ls -la
-rw-r--r-- 1 dobriain users 93788 Dec 14 06:08 MyFile
```

- - the dash indicates a file, a `d` would indicate a directory.
- `rw-` Read, Write, Execute permissions for the owner.
- `r--` Read, Write, Execute permissions for the group.
- `r--` Read, Write, Execute permissions for others.
- `1` # of hardlinks, for directories it shows the # of directories as well as hardlinks it contains.
- `dobriain` Owner.
- `users` Group.
- `93788` File size.
- `Dec 14 06:08` Date file was last modified.
- `MyFile` File or directory name.

Permissions



Decimal representation	Binary representation	Meaning
0	000	---
1	001	--x
2	010	-w-
3	011	-wx
4	100	r--
5	101	r-x
6	110	rw-
7	111	rwX

```
$ chmod 764 MyFile
```

```
$ ls -la
```

```
-rwxrw-r-- 1 dobriain users 93788 Dec 27 16:57 MyFile
```

Permissions

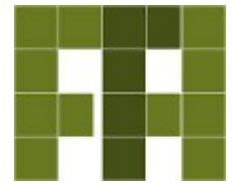


- u User.
- g Group.
- o Others.
- r Read.
- w Write.
- x Execute.
- T Sticky.
- S SUID/SGID.

```
$ chmod g-w MyFile
$ ls -la
-rwxr--r-- 1 dobriain users 93788 Dec 27 16:57 MyFile

$ chmod og+x MyFile
$ ls -la
-rwxr-xr-x 1 dobriain users 93788 Dec 27 16:57 MyFile

$ chmod +t MyFile
$ ls -la
-rwxr-xr-T 1 dobriain users 93788 Dec 27 16:57 MyFile
```



- **find** - search for files in a directory hierarchy

```
$ ls
```

```
MyFile  MyFile1  MyFile2  MyFile3
```

```
$ find ./ -name MyFile* -print
```

```
./workArea/MyFile1
```

```
./workArea/MyFile3
```

```
./workArea/MyFile2
```

```
./workArea/MyFile
```

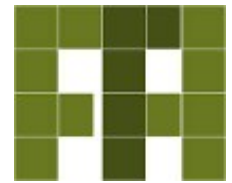
```
$ find ./ -name MyFile? -print
```

```
./playArea/MyFile1
```

```
./playArea/MyFile3
```

```
./playArea/MyFile2
```

locate



- **locate** - reads one or more databases prepared by updatedb and writes file names matching at least one of the patterns in the db.

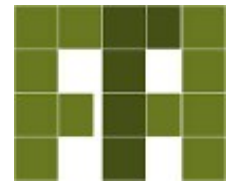
```
$ sudo updatedb
```

```
$ locate MyFile
```

```
/home/dobriain/Desktop/workArea/MyFile  
/home/dobriain/Desktop/workArea/MyFile1  
/home/dobriain/Desktop/workArea/MyFile2  
/home/dobriain/Desktop/workArea/MyFile3
```

```
$ locate --regex "MyFile\S"
```

```
/home/dobriain/Desktop/workArea/MyFile1  
/home/dobriain/Desktop/workArea/MyFile2  
/home/dobriain/Desktop/workArea/MyFile3
```

- **file** - Determines a file type.

```
$ ls
```

```
MyFile  MyFile1  MyFile2  MyFile3  MyJPG  MyPDF
```

```
$ file MyFile
```

```
MyFile: ASCII text
```

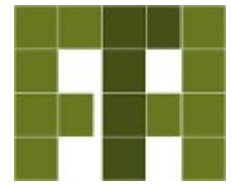
```
$ file MyJPG
```

```
MyJPG.jpg: JPEG image data, JFIF standard 1.01
```

```
$ file MyPDF
```

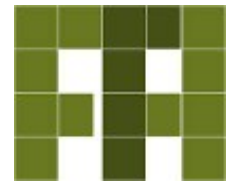
```
MyPDF.pdf: PDF document, version 1.6
```

Comparing and viewing files



- **cat** - concatenate files and print on the standard output
- **more** - file perusal filter for crt viewing
- **less** - opposite of more
- **head** - output the first part of files
- **tail** - output the last part of files
- **diff** - compare files line by line
- **cmp** - compare two files byte by byte
- **comm** - compare two sorted files line by line
- **wc** - print newline, word, and byte counts for each file
- **hexdump, hd** - ASCII, decimal, hexadecimal, octal dump
- **od** - dump files in octal and other formats

tail – monitor for changes

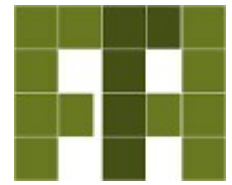


- **tail** - Output the last part of files.
 - **-f** – Follow, output appended data as the file grows.

```
$ tail -f /var/log/syslog
Dec 27 18:59:16 riomhairePAD wpa_supplicant[1256]:
eth1: WPA: Group rekeying completed with
5c:f8:a1:b3:79:c2 [GTK=CCMP]
```

- Cause a change by say plugging in a USB Stick and not the lines are displayed on the tail as the syslog file is appended.

Comparing and viewing files



```
$ wc MyFile1
```

```
39  908 5965 MyFile1
```

```
$ wc MyFile2
```

```
41  913 5998 MyFile2
```

```
$ diff MyFile1 MyFile2
```

```
0a1,2
```

```
> Free software gets an education
```

```
>
```

```
$ cmp MyFile1 MyFile2
```

```
MyFile1 MyFile2 differ: byte 1, line 1
```

```
$ hd MyFile3
```

```
00000000  54 68 65 20 71 75 69 63  6b 20 62 72 6f 77 6e 20 |The quick brown |
00000010  66 6f 78 20 6a 75 6d 70  73 20 6f 76 65 72 20 74 |fox jumps over t|
00000020  68 65 20 6c 61 7a 79 20  64 6f 67 0a                |he lazy dog.|
```

The process



- **ps** - report a snapshot of the current processes.
 - **a** - Show processes of all system terminals (tty).
 - **u** - Display user-oriented format.
 - **x** - Remove the restriction that process must have tty.

```
$ ps aux
```

USER	PID	%CPU	%MEM	VSZ	RSS	TTY	STAT	START	TIME	COMMAND
root	1	0.0	0.0	27036	2892	?	Ss	16:36	0:01	/sbin/init
root	2	0.0	0.0	0	0	?	S	16:36	0:00	[kthreadd]
root	3	0.0	0.0	0	0	?	S	16:36	0:00	[ksoftirqd/0]
root	5	0.0	0.0	0	0	?	S<	16:36	0:00	[kworker/0:0H]
avahi	1203	0.0	0.0	32352	1740	?	S	16:36	0:00	avahi-daemon: r
avahi	1208	0.0	0.0	32228	472	?	S	16:36	0:00	avahi-daemon: c
dobriain	1845	0.0	0.0	12624	320	?	Ss	16:36	0:00	/usr/bin/ssh-ag
dobriain	1848	0.0	0.0	24444	600	?	S	16:36	0:00	/usr/bin/dbus-l
dobriain	1849	0.0	0.0	31748	1944	?	Ss	16:36	0:00	//bin/dbus-daem
dobriain	1859	0.0	0.0	337396	3284	?	Sl	16:36	0:00	/usr/lib/at-spi

The process

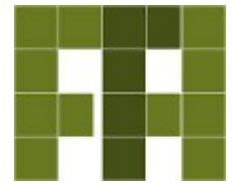


\$ top

```
top - 18:35:03 up 1:58, 2 users, load average: 0.03, 0.09, 0.16
Tasks: 208 total, 1 running, 207 sleeping, 0 stopped, 0 zombie
%Cpu(s): 1.3 us, 0.3 sy, 0.0 ni, 97.6 id, 0.8 wa, 0.0 hi, 0.0 si, 0.0 st
KiB Mem: 7738624 total, 1948064 used, 5790560 free, 144900 buffers
KiB Swap: 7942140 total, 0 used, 7942140 free, 1117184 cached
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
4419	dobriain	20	0	592m	19m	12m	S	6.4	0.3	0:05.52	gnome-terminal
1	root	20	0	27036	2892	1448	S	0.0	0.0	0:01.10	init
2	root	20	0	0	0	0	S	0.0	0.0	0:00.00	kthreadd
3	root	20	0	0	0	0	S	0.0	0.0	0:00.01	ksoftirqd/0
5	root	0	-20	0	0	0	S	0.0	0.0	0:00.00	kworker/0:0H
7	root	rt	0	0	0	0	S	0.0	0.0	0:00.05	migration/0
8	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcu_bh
9	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/0
10	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/1
11	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/2
12	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/3
13	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/4
14	root	20	0	0	0	0	S	0.0	0.0	0:00.00	rcuob/5

The process



- **kill** - send a signal to a process via process ID (PID).
 - **-1** - List signal types.
 - **-SIGTERM -15** - Terminate a process.
 - **-SIGKILL -9** - Kill a process.

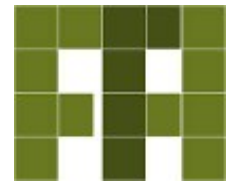
```
$ ps aux|grep top
dobriain  4545  0.0  0.0  20636  1616 pts/0    S+   18:35   0:00 top
dobriain  4622  0.0  0.0   9452   944 pts/3    S+   18:37   0:00 grep
--colour=auto top
```

```
$ kill -SIGTERM 4545
```

```
$ ps aux|grep top
dobriain  4697  0.0  0.0   9452   944 pts/3    S+   18:42   0:00 grep
--colour=auto top
```

- **killall** - Kill process by name.

Managing files and directories



```
$ ls -la
```

```
total 16
```

```
drwxr-xr-x 4 dobriain dobriain 4096 Dec 27 17:17 .  
drwxr-xr-x 3 dobriain dobriain 4096 Dec 27 16:56 ..  
drwxr-xr-x 2 dobriain dobriain 4096 Dec 27 17:17 MyDir  
-rwxr-xr-x 1 dobriain dobriain    0 Dec 27 16:57 MyFile  
drwxr-xr-x 2 root      root      4096 Dec 27 17:17 RootDir
```

```
$ rm -r MyDir
```

```
$ rm -r RootDir
```

```
rm: remove write-protected directory 'RootDir'?
```

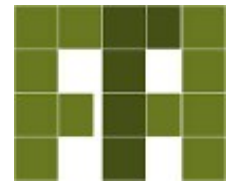
```
$ ls
```

```
MyFile  RootDir
```

```
$ sudo rm -r RootDir
```

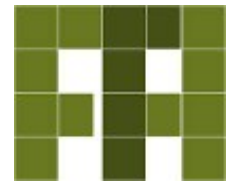
```
$ ls
```

```
MyFile
```

- **man** - an interface to the on-line reference manuals.
 - Manual sections:
 - 1) Executable programs (applications, commands etc.).
 - 2) Calls to the system provided by the shell.
 - 3) Calls to system libraries.
 - 4) Special files (generally for devices).
 - 5) Format of the configuration files.
 - 6) Games.
 - 7) Macro packages.
 - 8) System administration commands (root).
 - 9) Kernel routines.

System help



\$ **man printf**

PRINTF(1)

User Commands

PRINTF(1)

NAME

printf - format and print data

SYNOPSIS

printf FORMAT [ARGUMENT]...
printf OPTION

DESCRIPTION

Print ARGUMENT(s) according to FORMAT, or execute according to OPTION:

\$ **man 3 printf**

PRINTF(3)

Linux Programmer's Manual

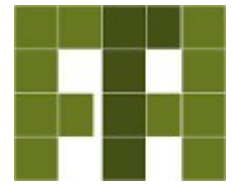
PRINTF(3)

NAME

printf, fprintf, sprintf, snprintf, vprintf, vfprintf, vsprintf,
vsnprintf - formatted output conversion

SYNOPSIS

#include <stdio.h>



- **apropos** - search the manual page names and descriptions.
 - Each manual page has a short description available within it.
 - apropos searches the descriptions for instances of keyword which is usually a regular expression, as if (-r) was used, or may contain wildcards (-w), or match the exact keyword (-e).



\$ **apropos printf**

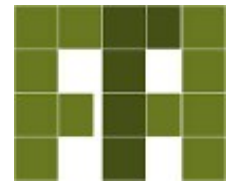
asprintf (3)	- print to allocated string
dprintf (3)	- print to a file descriptor
fprintf (3)	- formatted output conversion
fwprintf (3)	- formatted wide-character output conversion
printf (1)	- format and print data
printf (3)	- formatted output conversion
snprintf (3)	- formatted output conversion
sprintf (3)	- formatted output conversion
swprintf (3)	- formatted wide-character output conversion
vasprintf (3)	- print to allocated string
vdprintf (3)	- print to a file descriptor
vfprintf (3)	- formatted output conversion
vfwprintf (3)	- formatted wide-character output conversion
vprintf (3)	- formatted output conversion
vsnprintf (3)	- formatted output conversion
vsprintf (3)	- formatted output conversion
vswprintf (3)	- formatted wide-character output conversion
vwprintf (3)	- formatted wide-character output conversion
wprintf (3)	- formatted wide-character output conversion



- **mandb** - create or update the manual page index caches.
 - mandb is used to initialise or manually update index database caches that are usually maintained by man.
 - The caches contain information relevant to the current state of the manual page system and the information stored within them is used by the man-db utilities to enhance their speed and functionality.

```
$ mandb  
0 man subdirectories contained newer manual pages.  
0 manual pages were added.  
0 stray cats were added.  
0 old database entries were purged.
```

Packaging and compression



- **tar** - stores and extracts files from a tape or disk archive.
 - **-c, --create** create a new archive.
 - **-x, --extract, --get** extract files from an archive.
 - **-v, --verbose** verbosely list files processed.
 - **-f, --file ARCHIVE** use archive file or device ARCHIVE.
 - **-z, --gzip, --gunzip, --uncompress.**
 - **-j, --bzip2.**

```
$ tar -czvf MyFiles.tgz My*
```

```
MyFile
```

```
MyFile1
```

```
MyFile2
```

```
MyFile3
```

```
MyJPG.jpg
```

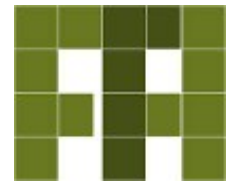
```
MyPDF.pdf
```

```
$ file MyFiles.tgz
```

```
MyFiles.tgz: gzip compressed data, from Unix, last  
modified: Tue Dec 31 07:45:37 2013
```

```
$ tar -xzvf MyFiles.tgz
```

Packaging and compression



Extension	Format
.tar	tar
.gz	gzip
.tgz	tar + gzip
.bz2	bzip2
.zip	zip
.z	compress

- If you need heavy compression and are willing to wait for it, use XZ.
- If you want to compress things a little, but don't have much time, GZIP and ZIP will work just fine.
- BZIP2 performs better than GZIP and ZIP but can take up to 6 times longer to decompress.
- COMPRESS though, seems to be pretty useless for all but the really impatient.

- **gzip, gunzip, zcat** - compress or expand files using Lempel-Ziv coding (LZ77).
- **xz, unxz** - compress or decompress .xz and .lzma files.
- **bzip2, bunzip2** - a block-sorting file compressor compresses files using the Burrows-Wheeler block sorting text compression algorithm, and Huffman coding.
 - Compression is generally considerably better than that achieved by more conventional LZ77/LZ78-based compressors.
- **compress, uncompress** -compress and expand data using adaptive Lempel-Ziv coding.

Disk operations



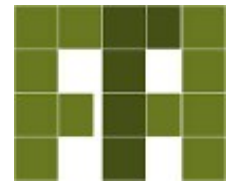
Filesystem	1k-blocks	Used	Available	Use%	Mounted on
/dev/hda1	7787712	421288	6970828	6%	/
/dev/hdb1	19541504	5742384	13799120	29%	/info
/dev/hdc	664432	664432	0	100%	/CD-ROM

- **df** - report file system disk space usage.
- **du** - estimate file space usage.
 - **-h, --human-readable**, print sizes in human readable format (e.g., 1K 234M 2G).

```
$ df -h
```

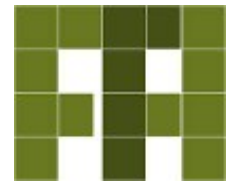
```
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda1       236M   43M  182M  19% /boot
/dev/sda2       451G   99G  330G  23% /
```


Hard Disk Devices



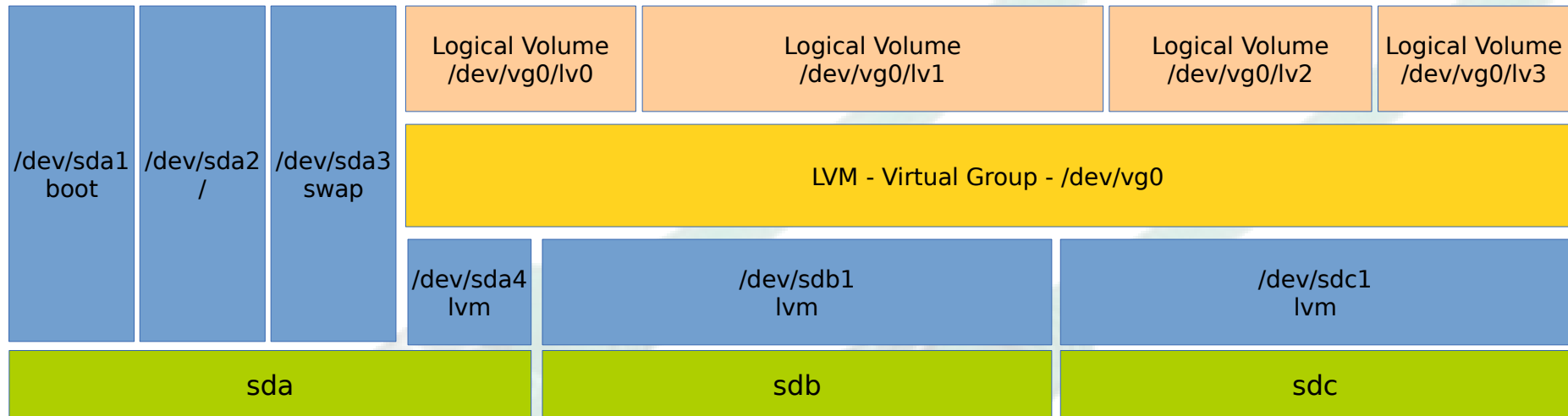
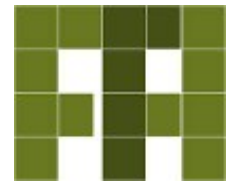
- Primary Master IDE /dev/hda
- Primary Slave IDE /dev/hdb
- Secondary Master IDE /dev/hdc
- Secondary Slave IDE /dev/hdd
- First SCSI drive /dev/sda
- Second SCSI drive /dev/sdb
- Third SCSI drive /dev/sdc

Partition Types

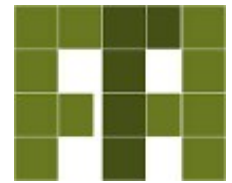


- Primary partition
 - Maximum of 4, Minimum of 1
 - /dev/hda1, /dev/hda2, /dev/hda3, /dev/hda4
- Extended Partition
 - Only 1 can exist on a physical disk
 - If exists it takes a Primary partition name
 - /dev/hda1 (Primary), /dev/hda2 (extended)
- Logical Partition
 - Logical partitions exist within an extended partition
 - Numbered from 5 to 16
 - /dev/hda5, /dev/hda6, /dev/hda7

Logical Volume Manager (LVM)

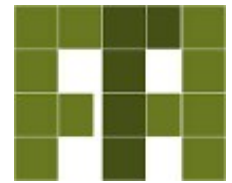


- Manages disk drives and similar mass-storage devices.
- Software layer on top of the hard disks and partitions, which creates an abstraction.



- LVM is used for the following purposes:
 - Managing large hard disk farms by allowing disks to be added and replaced without downtimes and services disruption, in combination with hot swapping.
 - On small systems LVM allows file systems to be easily resized as needed.
 - Performing consistent backups by taking snapshots of the logical volumes.
 - Creating single logical volumes of multiple physical volumes or entire hard disks, allowing for dynamic volume resizing.

Configuring LVM



Physical Volume

```
$ sudo pvcreate /dev/sda4
```

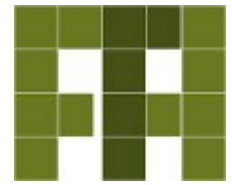
Volume Group

```
$ sudo vgcreate vg0 /dev/sda4
```

Logical Volume

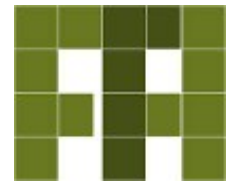
```
$ sudo lvcreate -L 20G -n lv0 vg0
```

PV Display



```
$ sudo pvdisplay
--- Physical volume ---
PV Name                /dev/sda4
VG Name                vg0
PV Size                134.89 GiB / not usable 3.00 MiB
Allocatable            yes
PE Size                4.00 MiB
Total PE               34532
Free PE                34532
Allocated PE           0
PV UUID                rrL4Q5-RWIZ-wM5a-N7QL-M3to-B1zO-0wK0yF
```

VG Display

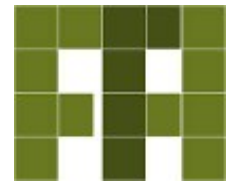


```
$ sudo vgdisplay
```

```
--- Volume group ---
```

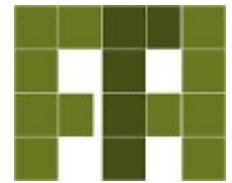
```
VG Name                vg0
System ID
Format                 lvm2
Metadata Areas         1
Metadata Sequence No   1
VG Access               read/write
VG Status               resizable
MAX LV                 0
Cur LV                 0
Open LV                 0
Max PV                  0
Cur PV                 1
Act PV                 1
VG Size                 134.89 GiB
PE Size                 4.00 MiB
Total PE                34532
Alloc PE / Size        0 / 0
Free PE / Size          34532 / 134.89 GiB
VG UUID                 9gYEvb-Lhvy-BBry-vgBU-nqVs-fIeT-e00HaP
```

LV Display



```
$ sudo lvsdisplay
--- Logical volume ---
LV Path                /dev/vg0/lv0
LV Name                 lv0
VG Name                 vg0
LV UUID                 irpKXm-pzKb-yvQA-t8Vw-gZ19-Ir9k-vRLL0k
LV Write Access         read/write
LV Creation host, time OB-Xen, 2013-10-27 20:14:19 +0000
LV Status                available
# open                  0
LV Size                 20.00 GiB
Current LE              5120
Segments                1
Allocation              inherit
Read ahead sectors      auto
- currently set to     256
Block device            254:4
```


Standard Input and Output



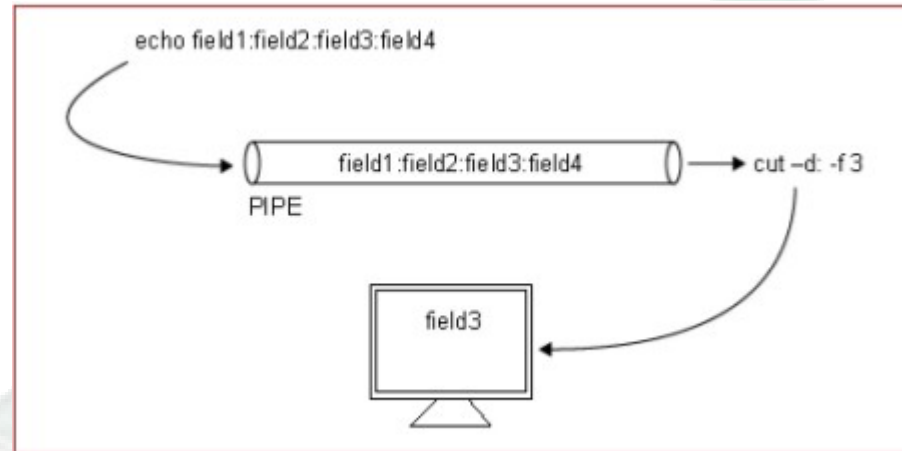
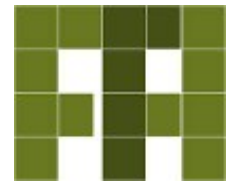
- Shell commands have three standard I/O streams
 - stdin - Standard Input (usually keyboard)
 - stdin is referenced with file descriptor 0
 - stdout - Standard Output (usually screen)
 - stdout is referenced with file descriptor 1
 - stderr - Standard Error (usually screen)
 - stderr is referenced with file descriptor 2

Rerouting



- This is a mechanism where the output or input of a command can be other than the Standard In (stdin), Standard Out (stdout), Standard Error (stderr).
 - stdin – Normally Keyboard, mouse etc..
 - stdout – Normally Monitor, printer etc..
 - stderr – Error logs and/or stdout (errors to the screen).
- It is important to realise that the shell is what performs the rerouting.
- The command never sees the rerouting.

Rerouting - pipe - (|)



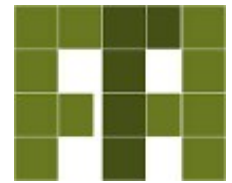
```
$ echo field1:field2:field3:field4
```

```
field1:field2:field3:field4
```

```
$ echo field1:field 2:field 3:field 4 | cut -d: -f 3
```

```
field 3
```

Rerouting - pipe - (|)

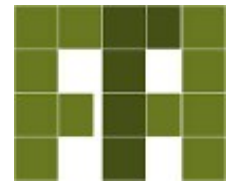


```
$ cat MyFile* | grep "conference"
```

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology **conference**, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology **conference**, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

Rerouting – redirection (>)



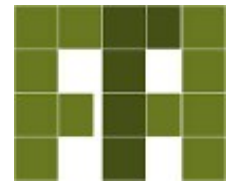
- > - redirect stdout to.

```
$ cat MyFile* | grep "conference" > MyFile-conference
```

```
$ ls
```

```
MyFile  MyFile1  MyFile2  MyFile3  MyFile-conference  
MyJPG.jpg  MyPDF.pdf
```

Rerouting – redirect (>)

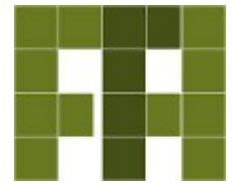


\$ **cat MyFile-conference**

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

Rerouting – append (>>)



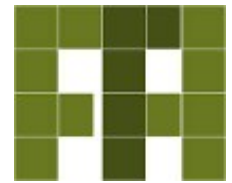
- >> - append standard output to.

```
$ echo -e "\nFree Software Academy"
```

```
Free Software Academy
```

```
$ echo -e "\nFree Software Academy" >> MyFile-conference
```

Rerouting – append (>>)



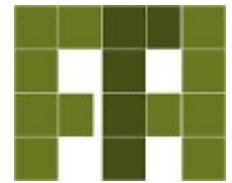
\$ **cat MyFile-conference**

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

Free Software Academy

Redirection of stderr



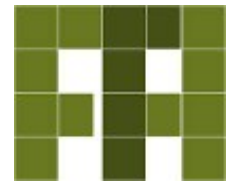
- **2>>** - append stderr to.

```
$ ls blob
ls: cannot access blob: No such file or directory

$ ls blob 2>> MyError

$ cat MyError
ls: cannot access blob: No such file or directory
```

Redirection of stderr to stdout (>&)

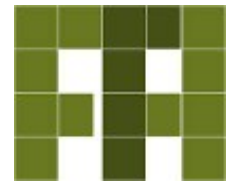


- **>&** - append stderr to.
- This instructs the shell to connect stream 2 (stderr) to the same stream as stream 1 (stdout)

```
$ ls -l ./ /homes > MyLS
ls: cannot access /homes: No such file or directory
$ cat MyLS
./:
total 1008
-rwxr-xr-x 1 dobriain dobriain      50 Dec 27 17:52 MyFile
-rw-r--r-- 1 dobriain dobriain  5965 Dec 27 18:08 MyFile1
-rw-r--r-- 1 dobriain dobriain  5998 Dec 27 18:09 MyFile2

$ ls -l ./ /homes > MyLS 2>&1
$ cat MyLS
ls: cannot access /homes: No such file or directory
./:
total 1012
-rwxr-xr-x 1 dobriain dobriain      50 Dec 27 17:52 MyFile
-rw-r--r-- 1 dobriain dobriain  5965 Dec 27 18:08 MyFile1
-rw-r--r-- 1 dobriain dobriain  5998 Dec 27 18:09 MyFile2
```

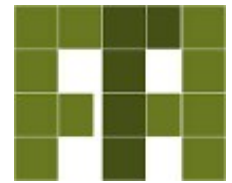
Rerouting – redirect stdin (<)



- < redirect stdin. Handy to use in scripts for backups.
- mail requires a subject and body in order to send a message. By redirecting stdin from a file instead of the keyboard, your script can mail you when done.

```
$ mail dobriain < MyError
mail
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/dobriain": 1 message 1 new
>N 1 dobriain  Wed Feb  5 13:13  18/486
& n
Message 1:
From dobriain Wed Feb  5 13:13:45 2013
Date: Wed, 5 Feb 2013 13:13:45 GMT
From: diarmuid
To: dobriain

ls: cannot access blob: No such file or directory
```



- **tee** - read from stdin and write to stdout and files.

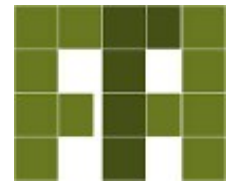
```
$ cat MyFile1 | grep conference | tee MyConference-extract
```

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

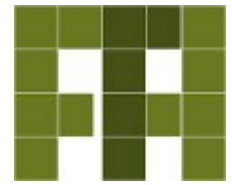
```
$ cat MyConference-extract
```

To raise awareness about their work, the project partners launched the Free Knowledge, Free Technology conference, the first edition of which took place in Barcelona in July 2008, attracting representatives of the software development and educational communities from around the world. They have also published a book titled 'Introduction to Free Software' that has been incorporated into a UOC master's course.

Shells

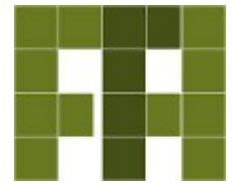


- Bourne Shell (/bin/sh)
 - UNIX shell written by Stephen Bourne at Bell Labs and released in 1977.
- UNIX C shell (/bin/csh)
 - Unix shell created by Bill Joy in the late 1970s.
- tcsh shell (/bin/tcsh)
 - Improved version of the UNIX C shell.
 - Command-completion feature.
- Korn Shell (/bin/ksh)
 - Unix Shell which was developed by David Korn in the early 1980s.
 - Korn Shell complies with POSIX.2, Shell and Utilities, Command Interpreter (IEEE Std 1003.2-1992.)
 - Major differences between Korn Shell and the traditional Bourne shell include:
 - Job control, command aliasing, and command history designed after the corresponding C shell features.
 - WYSIWYG-style line editing modes designed to make an interactive shell session behave like vi, Emacs, or XEmacs.
 - Associative arrays and built-in floating point arithmetic operations.
 - Dynamic extensibility of built-in commands.



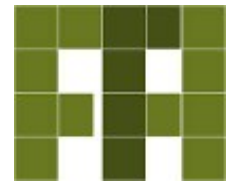
- GNU Bourne-Again Shell (/bin/bash)
 - GNU improved version of the Bourne Shell written by Brian Fox for the GNU Project as a free software replacement for the Bourne shell (sh).
 - Bourne Shell (sh) compatible command language interpreter that executes commands read from the standard input or from a file.
 - Bash also incorporates useful features from the Korn and C shells (ksh and csh).
 - Bash is a conformant implementation of the Shell and Utilities portion of the IEEE POSIX specification (IEEE Standard 1003.1).

Background (&)



- & - runs a job in the background.
- Shell available for use.

```
$ ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT &  
[1] 6901  
$
```



- jobs
 - Forgot to run in the background.
 - Stop job with Control-Z.
 - Lists suspended process and their job number. You can manipulate the process by using the `fg` and `bg` commands

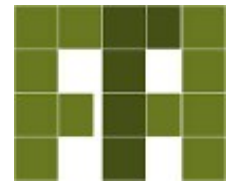
Ctrl-Z

```
$ ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT  
^Z
```

```
[1]+  Stopped ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT
```

```
$ jobs
```

```
[1]+  Stopped ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT
```

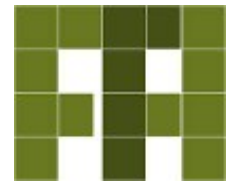



- **bg**
 - puts a job in the background.

```
$ jobs
[2]+  Stopped ls -laR ./ 2> /dev/null| grep -e "^-" > ~/Desktop/OUT

$ bg %2
[2]+  Stopped ls -laR ./ 2> /dev/null| grep -e "^-" > ~/Desktop/OUT &
```

- **bg %2** will send job 2 to background



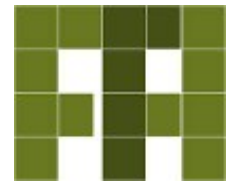
- fg
 - Brings a background job into the foreground.

```
$ jobs
jobs
[3]-  Running  ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT &

$ fg %3
[3]-  Done ls -laR ./ 2> /dev/null | grep -e "^-" > ~/Desktop/OUT
```

- fg %3 will bring job 3 to foreground

Bash command interpreter



- **Tab:** No need to write the name of a file, directory or command in full.
 - Writing the first characters and then press the tab key will write the rest.
 - If there is more than one possible match it will display the different options.
- **Ctrl+L**
 - Clears the screen (like the clear command).
- **Shift+PgUp**
 - Shows half the previous screen.
- **Shift+PgDn**
 - Shows half the next screen.
- **Ctrl+W**
 - Eliminates the last written word.
- **Ctrl+T**
 - Interchanges the order of the last characters.
- **Ctrl+U**
 - Deletes all characters in front of the cursor.
- **Ctrl+D**
 - Exits the command interpreter (equivalent to executing a logout command).

Bash history



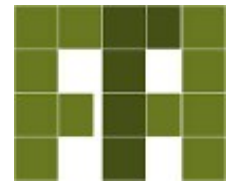
- history – Bash history library stores the
- -c – Clears history

```
$ history
 16  cd Desktop
 17  ls
 18  cat MyFile
 19  ls
 20  cat MyFile*
 21  cat MyFile1 | grep conference
 22  perl -wle 'print q/hello world/'
 23  gimp MyPDF.pdf
 24  less MyFile2
 25  history

$ !per
perl -wle 'print q/hello world/'
hello world

$ !18
The quick brown fox jumped over the lazy dog.
```

The Interactive Shell



- Shell Variables

- bash

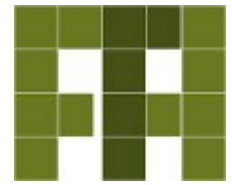
```
$ echo $PS1
```


```
Default PS1      [\u@\h \W]\$
```

```
[dobriain@gluais /usr/bin]$
```

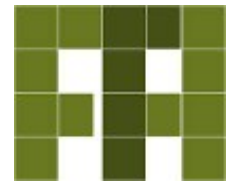
- \u username
 - \h hostname
 - \W current directory

Using the command line



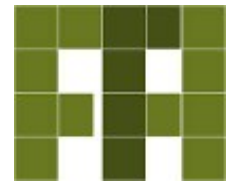
- Shell responds to  by scanning the line and performing the following:
 - Notes the spaces to identify words (arguments).
 - Interprets wildcards to generate filenames.
 - Removes any quote characters.
 - Substitutes variables.
 - Substitutes embedded commands.

Using the command line

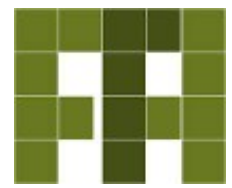


- `echo *`
 - This will NOT echo the `*` onto the screen.
 - It will provide a listing of all the files in the current directory.
 - This is because the SHELL interprets the wildcard and passes the value (all files) as an argument to the echo command.
- `echo $USER`
 - This will display the value assigned to the user variable.
- `echo "$USER"`
 - Note it also displays the value assigned to the user variable.
 - This is because double quotes protect most special characters EXCEPT double quotes allow variable and command substitution.
- `echo '$USER'`
 - Note this will display `$USER` on the screen.
 - This is because single quotes disable recognition of all special characters.

Using the command line



```
$ echo *
MyConference-extract MyError MyFile MyFile1
MyFile2 MyFile3 MyFile-conference MyJPG.jpg MyLS
MyPDF.pdf OUT
$ echo $USER
dobriain
$ echo "$USER"
dobriain
$ echo '$USER'
$USER
```

- Single Quotes '
 - Disable recognition of all special characters
- Double Quotes "
 - Protect most special characters except allow for variable and command substitution (e.g., echo "today is `date`")
- Backstroke \
 - Escapes any special meaning of the next character
- Backquote `
 - Used for command substitution (e.g., HERE=`pwd`)
- Dollar sign \$
 - Variable substitution character

Quoting examples



- What output do you expect for each of the following where:
 - The user is `dobriain`
 - The date is `Tue Dec 31 11:31:16 GMT 2013`

```
$ echo "welcome $USER today\'s date is `date`"
```

```
$ echo "welcome $USER today's date is 'date'"
```

```
$ echo "welcome $USER today's date is `date`"
```

Bash script



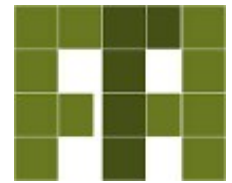
```
$ cat MyBASH-Script
#!/bin/bash

echo "My first bash script"
clear
#####
#
#   My first bash shell script to demonstrate conditional instructions #
#
#####

# Declare variables
my_variable="My first variable"
my_num1=3
my_num2=4
my_array=( G N U L i n u x )

# Main script section
echo; echo "$my_variable"; echo
ans=`expr $my_num2 - $my_num1`
echo $ans; echo
```

Bash script – Conditionals

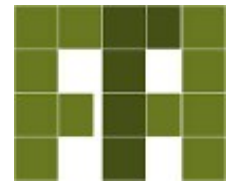


```
# IF/ELSE
if [ -f /var/log/syslog ]; then
    echo "The syslog file exists."; echo
else
    echo "The syslog file does NOT exist."; echo
fi

# CASE
case $ans in
    1) echo "1 must have been the answer of the earlier sum.";echo;;
    2) echo "2 must have been the answer of the earlier sum.";echo;;
    *) echo "Don't know the result of the earlier sum.";echo;;
Esac

# LOOP
for i in ${my_array[@]}; do
    echo "$i";
done
```

Bash script



- The script must be made executable and can then be ran.

```
$ chmod +x MyBASH-Script  
$ ./MyBASH-Script
```

```
My first variable
```

```
1
```

```
The syslog file exists.
```

```
1 must have been the answer of the earlier sum.
```

```
$
```



C²S CONSULTING

Knoppix Workshop



FREE
TECHNOLOGY
ACADEMY

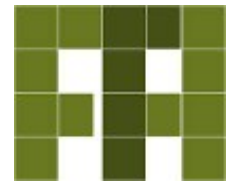


Diarmuid Ó Briain

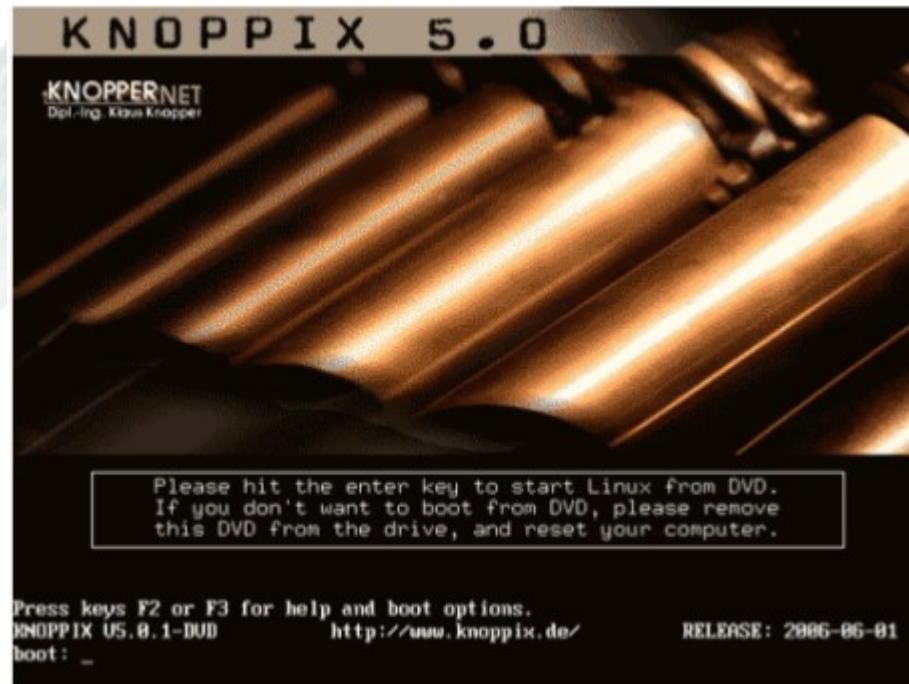
CEng, FIEI, FIET, CISSP

diarmuid@obriain.com

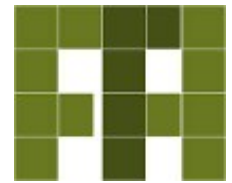
Knoppix Workshop



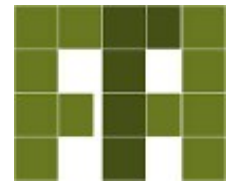
- Follow the Knoppix Workshop from the FTA GNU/Linux Basic handbook.



Knoppix Workshop summary



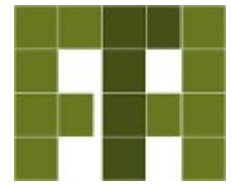
- Home directory - /home/dobriain
- Mount and /etc/fstab
- Why `rm -rf *` is so dangerous
- `useradd`, `passwd`, `groupadd`, `chown`, `chgrp`
- `su`



- **su** - change user ID or become superuser.

```
$ ls -la
total 36
drwxr-xr-x 2 dobriain dobriain 4096 Dec 31 13:12 .
drwxr-xr-x 3 dobriain dobriain 4096 Dec 31 13:12 ..
-rwxr-xr-x 1 dobriain dobriain  46 Dec 31 13:12 MyFile
-rw-r--r-- 1 dobriain dobriain 5965 Dec 31 13:12 MyFile1
-rw-r--r-- 1 dobriain dobriain 5998 Dec 31 13:12 MyFile2
-rw-r--r-- 1 dobriain dobriain  44 Dec 31 13:12 MyFile3
-rw----- 1 root      root      19 Dec 31 13:12 MySU-File

$ su
Password: <superuser password>
# cat MySU-File
This is a su file.
#
```



- **sudo** - execute a command as another user.

As the superuser install the sudo program.

```
# apt-get install sudo
```

Add the users who have access to sudo to the sudoers group.

```
# vi /etc/group
```

```
~~~~
```

```
sudo:x:27:dobriain
```

```
~~~~
```

```
:wq!
```

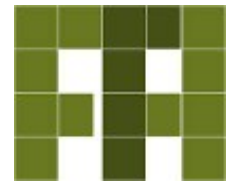
```
# exit
```

```
$ sudo cat MySU-File
```

```
[sudo] password for dobriain:
```

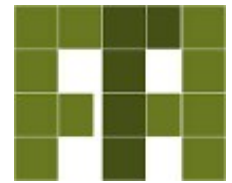
```
This is a su file.
```

Knoppix USB Stick



- Plugin USB stick.
- Use dmesg to determine the USB device, in this case /dev/sdb.

```
$ dmesg
[23009.350985] usb 1-1.2: Product: USB DISK
[23009.350988] usb 1-1.2: Manufacturer: SMI Corporation
[23009.391597] usb-storage 1-1.2:1.0: USB Mass Storage device
detected
[23009.391727] scsi6 : usb-storage 1-1.2:1.0
[23009.391886] usbcore: registered new interface driver usb-storage
[23011.006289] scsi 6:0:0:0: Direct-Access          SMI          USB DISK
                1100 PQ: 0 ANSI: 0 CCS
[23011.006786] sd 6:0:0:0: Attached scsi generic sg2 type 0
[23011.007394] sd 6:0:0:0: [sdb] 7987200 512-byte logical blocks:
(4.08 GB/3.80 GiB)
```



- Create two partitions on USB Stick, one for OS and one for storage, both FAT32 partitions.

```
$ sudo gparted
```

The screenshot shows the GParted application window titled `/dev/sdb - GParted`. The window has a menu bar (GParted, Edit, View, Device, Partition, Help) and a toolbar with icons for creating, deleting, moving, copying, pasting, and undo/redo. A dropdown menu shows `/dev/sdb (3.81 GiB)`. The main area displays two partitions, `/dev/sdb1` and `/dev/sdb2`, each 1.90 GiB in size, both formatted as FAT32. Below this is a table with the following data:

Partition	File System	Label	Size	Used	Unused	Flags
<code>/dev/sdb1</code>	fat32	Knoppix	1.90 GiB	3.82 MiB	1.90 GiB	boot
<code>/dev/sdb2</code>	fat32	Storage	1.90 GiB	3.82 MiB	1.90 GiB	

At the bottom of the window, it says "0 operations pending".

Knoppix USB Stick – fdisk



- Alternatively create partations with fdisk.

```
$ sudo fdisk /dev/sdb
```

```
Command (m for help): p
```

```
Disk /dev/sdb: 4089 MB, 4089446400 bytes
```

```
126 heads, 62 sectors/track, 1022 cylinders, total 7987200 sectors
```

```
Units = sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes / 512 bytes
```

```
I/O size (minimum/optimal): 512 bytes / 512 bytes
```

```
Disk identifier: 0x000e56d9
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sdb1	*	2048	3993599	1995776	b	W95 FAT32
/dev/sdb2		3993600	7987199	1996800	b	W95 FAT32

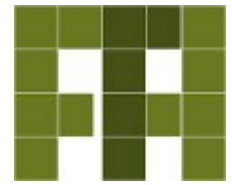
```
$ sudo mkfs -t vfat /dev/sdb1
```

```
mkfs.vfat 3.0.16 (01 Mar 2013)
```

```
$ sudo mkfs -t vfat /dev/sdb2
```

```
mkfs.vfat 3.0.16 (01 Mar 2013)
```

Knoppix USB Stick – mkfs



- Make VFAT Filesystem on each partition.
- Install Master Boot Record in /dev/sdb1.
- Download Knoppix from a <http://www.knoppix.org> download mirror.
- Install unetbootin.

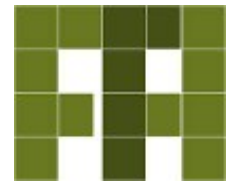
```
$ sudo install-mbr -e1 /dev/sdb
```

```
$ ls
```

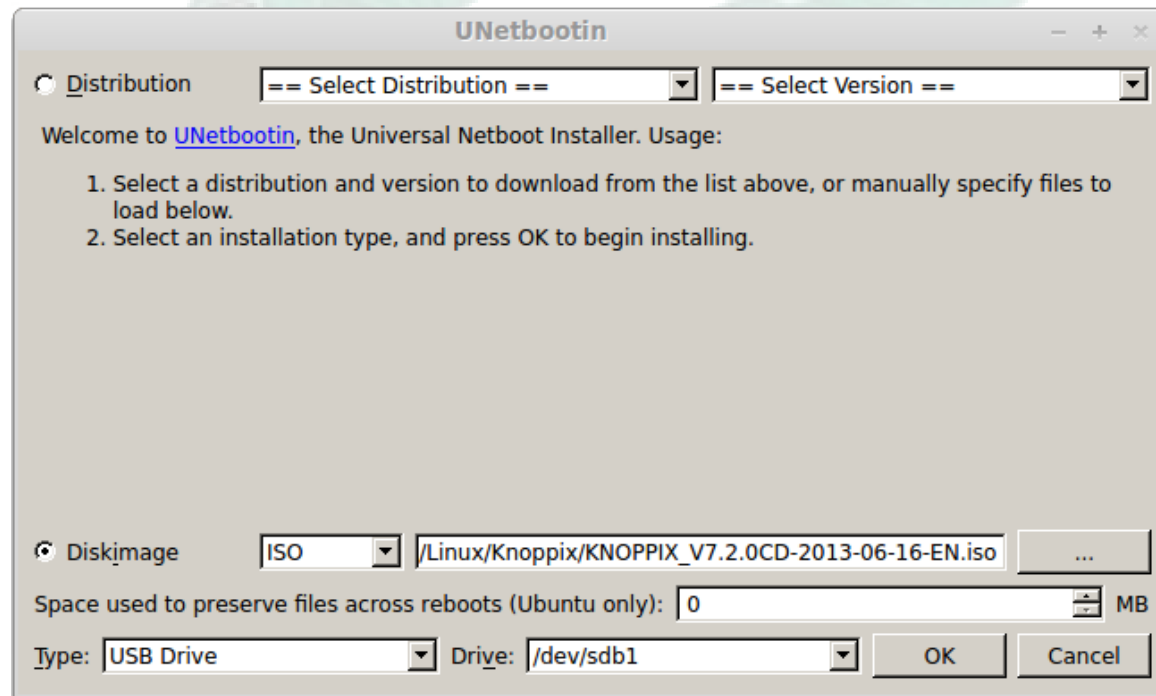
```
KNOPPIX_V7.2.0CD-2013-06-16-EN.iso
```

```
$ sudo apt-get install unetbootin
```

UNetbootin



- Select Diskimage, ISO and the Knoppix ISO image.
- Select USB Drive and /dev/sdb1 (The Linux partition on stick).

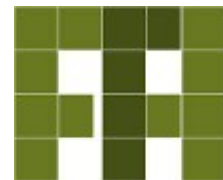


- 

KNOPPIX

KNOPPIX 7.

KNOPPIX 7.2



Thank you