Unit II

Booting Up RPi- Operating System and Linux Commands

WHAT IS LINUX ?

- An operating system.

- Software on a computer that enables applications and the computer operator to access the devices on the computer to perform desired functions.

Where is Linux?

- Linux, which began its existence as a server OS and Has become useful as a desktop OS, It can also be used on all of these devices. "From wristwatches to supercomputers," is the popular description of Linux' capabilities.

BEFORE LINUX

- In 80's, Microsoft's DOS was the dominated OS for PC
- Apple MAC was better, but expensive
- UNIX was much better, but much, much more expensive. Only for minicomputer for commercial applications
- People was looking for a UNIX based system, which is cheaper and can run on PC
- Both DOS, MAC and UNIX were proprietary, i.e., the source code of their kernel is protected
- No modification is possible without paying high license fees

FOUNDER OF LINUX

- In sept 1991, while attending the University of Helsinki , Torvalds became curious about operating systems and frustrated by the licensing of MINIX, which limited it to educational use only. He began to work on his own operating system which eventually became the Linux kernel.


MR: Linus Torvalds

LINUX TODAY

- Linux has been used for many computing platforms
  - PC, PDA, Supercomputer,...
- Not only character user interface but graphical user interface is available
- Commercial vendors moved in Linux itself to provide freely distributed code. They make their money by compiling up various software and gathering them in a distributable format
  - Red Hat, Slackware, etc

GROWING AND GROWING...

- In order to encourage wide dissemination of his OS, Linus made the source code open to public. At the end of 1992 there were about a hundred Linux developers. Next year there were 1000. And the numbers multiplied every year.
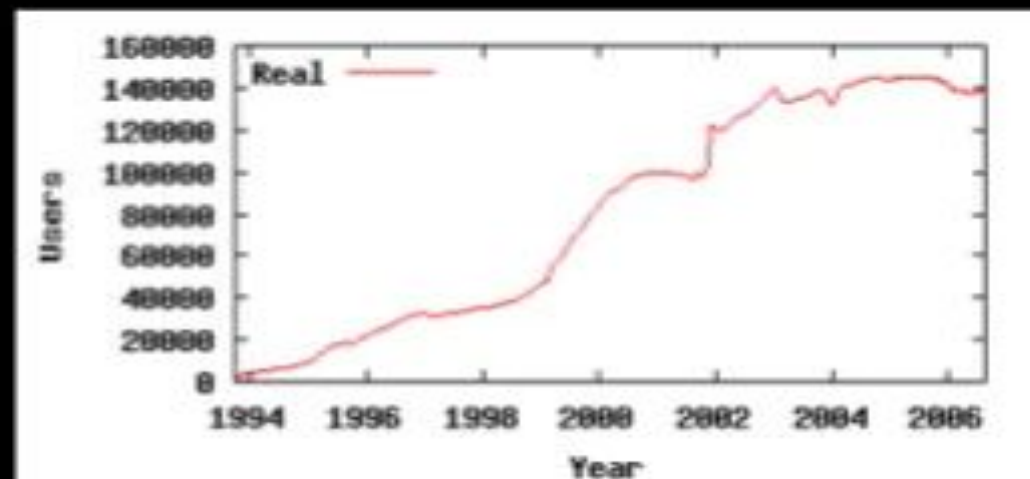
138712
users
registered
155679
machines
registered

Linux
Registered User # 425052

LINUX – A FREE SOFTWARE

• Free software, as defined by the FSF (Free Software Foundation), is a "matter of liberty, not price." To qualify as free software by FSF standards, you must be able to:

- Run the program for any purpose you want to, rather than be restricted in what you can use it for.
- View the program's source code.

Linux advantages

- No licensing fees
- Most server software is stable and reliable
- Requires minimal hardware resources
- Open source allows for greater security

Linux disadvantages

- Supporting it is expensive
- Troubleshooting is difficult
- Configuration is complicated and time-consuming
- Desktop software is unstable and buggy

TechRepublic

LINUX SYSTEM ARCHETECTURE
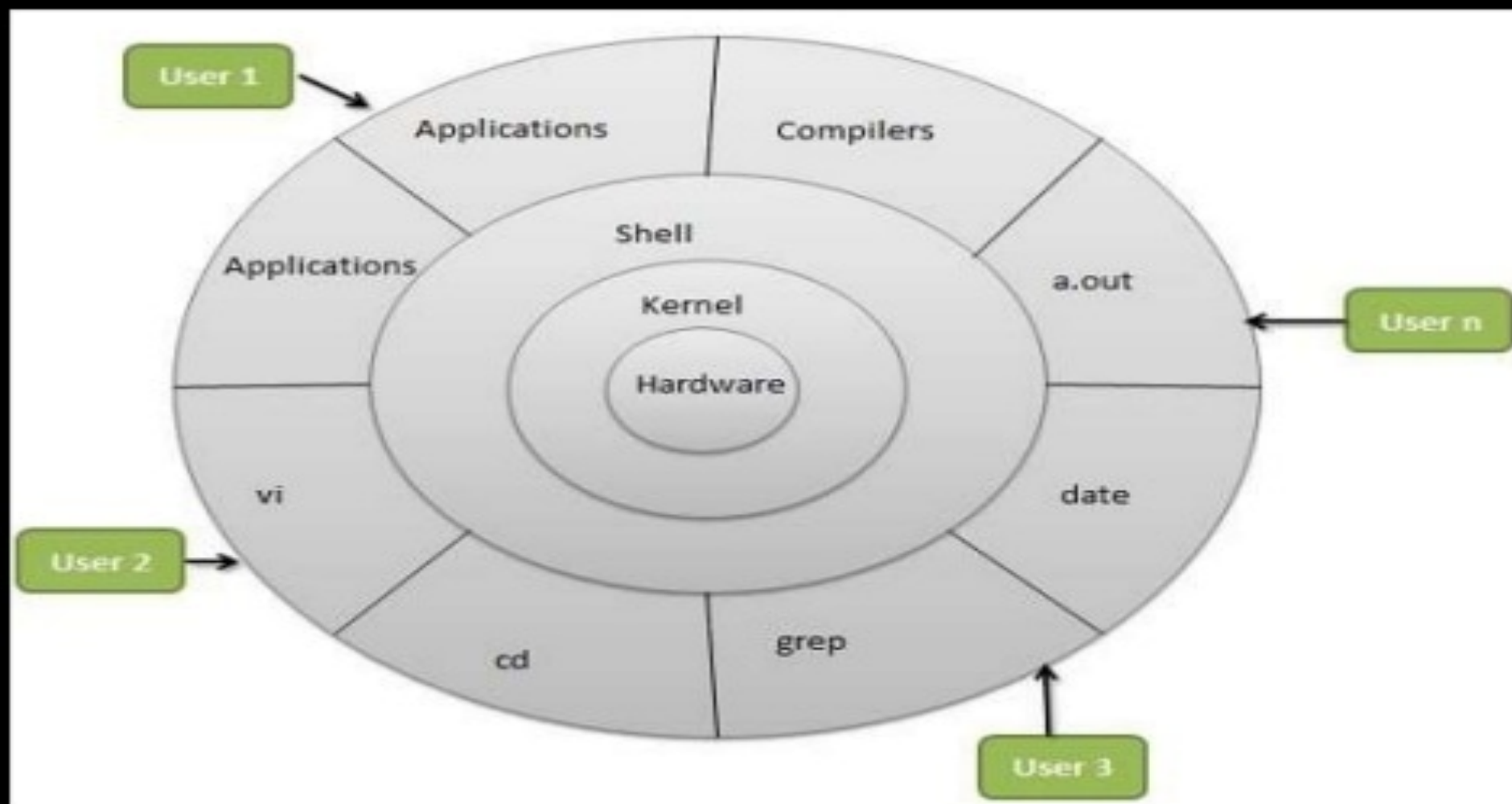
Hardware layer:-

consists of several peripheral devices like CPU, HDD, and RAM.

Shell:-

- It is an interface among the kernel and user. It can afford the services of kernel.

- It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes.

- These operating systems are categorized into two different types, which are the **graphical shells** and **command-line shells**.

- The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface.

- Thus, both of these shells implement operations.

- However, the graphical user interface shells work slower as compared to the command-line interface shells.

Kernel:-

- The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS.

- This operating system contains distinct types of modules and cooperates with underlying hardware directly.

- The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system.
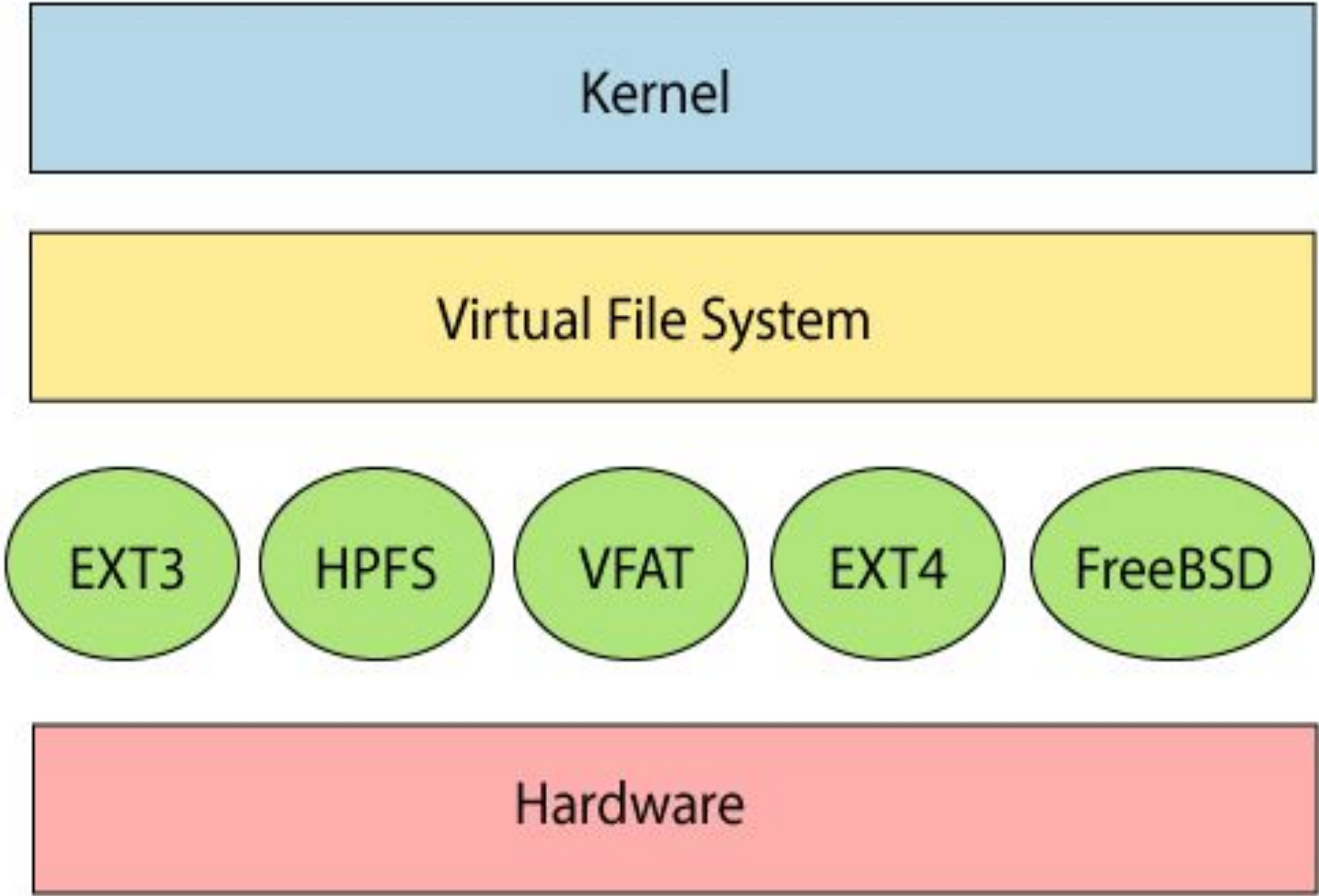
System Libraries:-

- These libraries can be specified as some special functions.

- These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

System Utility Programs:- It is responsible for doing specialized level and individual activities.

Linux File System

- A file system is a logical collection of files on a partition or disk.

- Everything in Linux/Unix is considered to be a file, including physical devices such as DVD-ROMs, USB devices, and floppy drives.

- LFS is generally a built-in layer of a Linux operating system used to handle the data management of the storage.

- It helps to arrange the file on the disk storage. It manages the file name, file size, creation date, and much more information about a file.

- Linux file system contains the following sections:

1. The root directory (/)
2. A specific data storage format (EXT3, EXT4, BTRFS, XFS and so on)
3. A partition or logical volume having a particular file system.

EXT3

- *An enhanced version of the ext2 file system.*

- *After installation, it is sometimes necessary to create a new ext3 file system.*

- *For example, if you add a new disk drive to the system, you may want to partition the drive and use the ext3 file system.*

- *The steps for creating an ext3 file system are as follows:*

- *Format the partition with the ext3 file system using mkfs.*

- *Label the partition using e2label.*

provide the following advantages:
- **Availability**
- **Data Integrity**
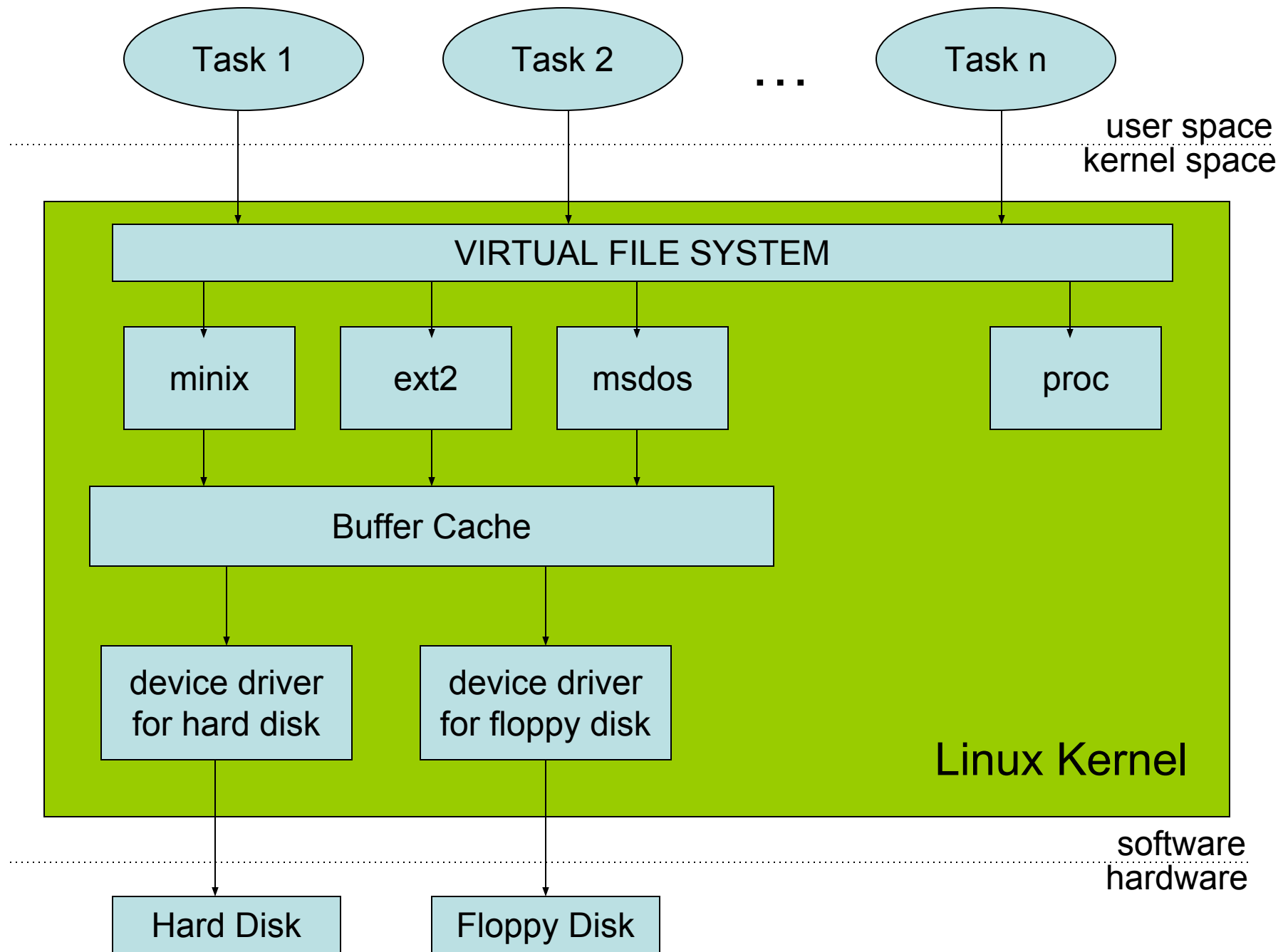- **Speed**
- **Easy Transition**

EXT4

- **Ext4** file system is the faster file system among all the Ext file systems.

- It is a very compatible option for the SSD (solid-state drive) disks, and it is the default file system in Linux distribution.

- Primarily improves performance, reliability, and capacity.

- To improve reliability, metadata and journal checksums were added.

- Data allocation was changed from fixed blocks to extents.

- EXT4 reduces fragmentation by scattering newly created files across the disk so that they are not bunched up in one location at the beginning of the disk, as many early PC filesystems did.

- The best method for upgrading to EXT4 from EXT3 is to back up all the data on the target filesystem partition, use the **mkfs** command to write an empty EXT4 filesystem to the

File Allocation Table (FAT)

- All of the Linux filesystem drivers support all three FAT types, namely [FAT12](), [FAT16]() and [FAT32]().

- VFAT is an extension of the FAT file system and was introduced with Windows 95.

- VFAT maintains backward compatibility with FAT but relaxes the rules.

- For example, VFAT filenames can contain up to 255 characters, spaces, and multiple periods.

- Although VFAT preserves the case of filenames.

The Virtual File System idea

- Multiple file systems need to coexist
- But filesystems share a core of common concepts and high-level operations
- So create a filesystem abstraction
- Applications interact with this VFS
- Kernel translates abstract-to-actual

Task 1　　　Task 2　　　...　　　Task n

user space
kernel space

VIRTUAL FILE SYSTEM

minix　　　ext2　　　msdos　　　proc

Buffer Cache

device driver for hard disk　　　device driver for floppy disk

Linux Kernel

software
hardware

Hard Disk　　　Floppy Disk

Filesystem Comparison

|	Minix	Ext	Xia	Ext2
Maximal FS size	64MB	2GB	2GB	4TB
Maximal filesize	64MB	2GB	64MB	2GB
Maximal filename	14/30 chars	255 chars	248 chars	255 chars
3 timestamps	no	no	yes	yes
Extensible?	no	no	no	yes
Can vary block size?	no	no	no	yes
Code is maintained?	yes	no	?	yes

Common concepts

- Files are represented by inodes
- Directories are files too (with dentries)
- Devices accessed by I/O on 'special' files
- UNIX filesystems can implement 'links'

Inodes

- A structure that contains file's description:
  - Type
  - Access rights
  - Owners
  - Timestamps
  - Size
  - Pointers to data blocks

- Kernel keeps the inode in memory (open)

Inode diagram

inode

File info

Direct blocks

Indirect blocks

Double
Indirect
Blocks
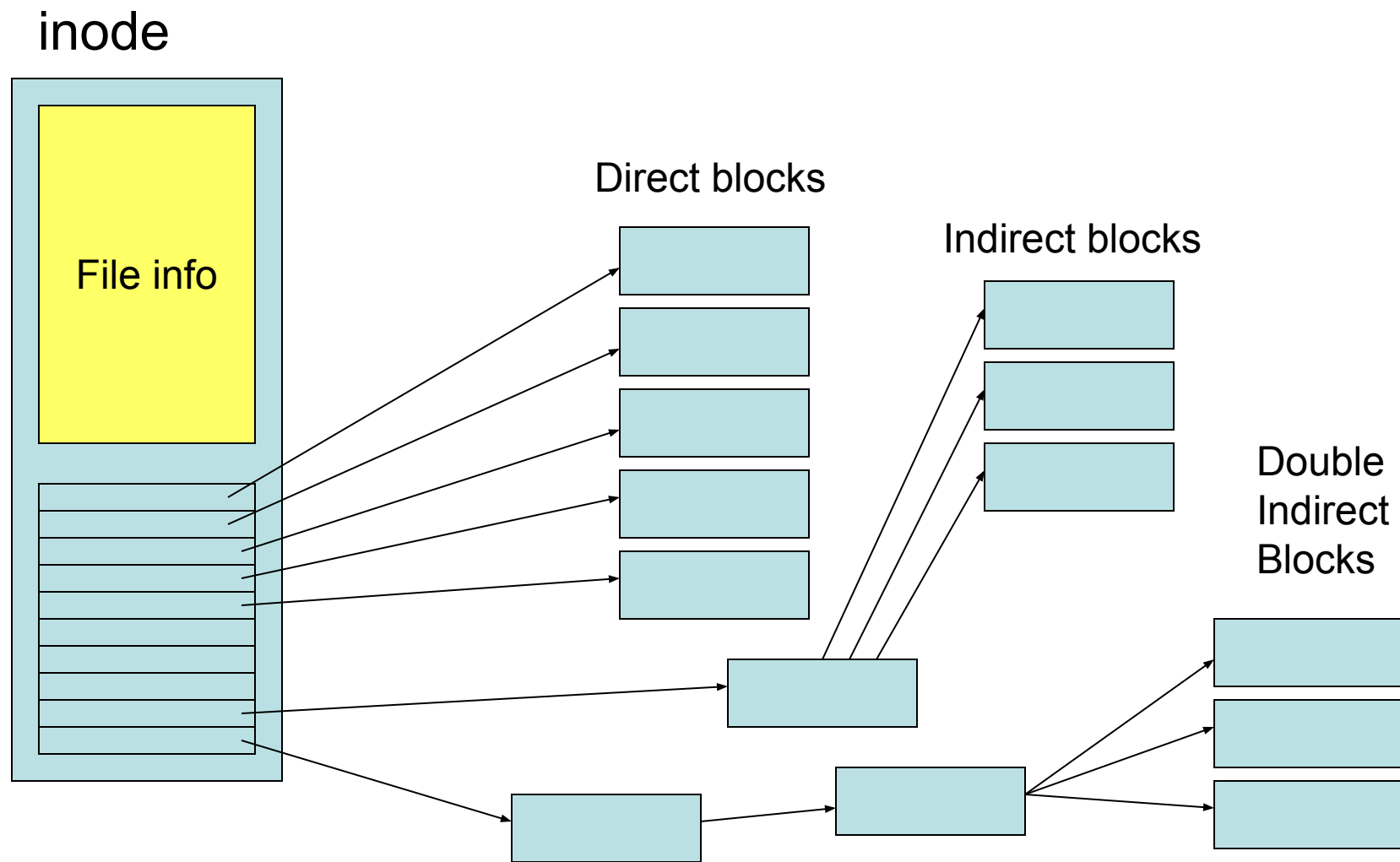
Directories

- These are structured in a tree hierarchy
- Each can contain both files and directories
- A directory is just a particular type of file
- Special user-functions for directory access
- Each dentry contains filename + inode-no
- Kernel searches the directory tree, and translates a pathname to an inode-number

Directory diagram

Inode Table

Directory

| i1	name1
i2	name2
i3	name3
i4	name4
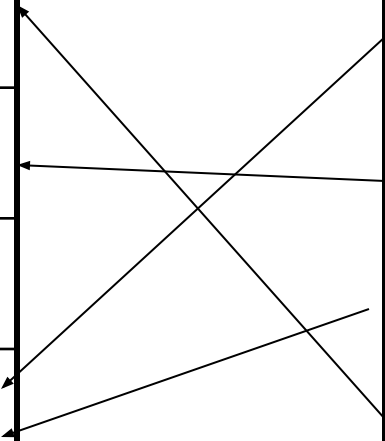
Links

- Multiple names can point to same inode
- The inode keeps track of how many links
- If a file gets deleted, the inode's link-count gets decremented by the kernel
- File is deallocated if link-count reaches 0
- This type of linkage is called a 'hard' link
- Hard links may exist only within a single FS
- Hard links cannot point to directories (cycles)
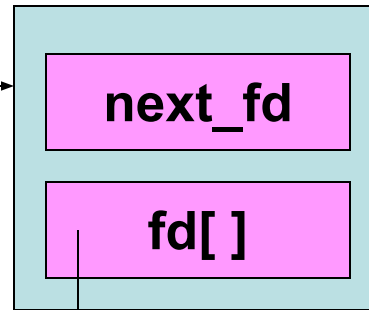
Symbolic Links

- Another type of file linkage ('soft' links)
- Special file, consisting of just a filename
- Kernel uses name-substitution in search
- Soft links allow cross-filesystem linkage
- But they do consume more disk storage
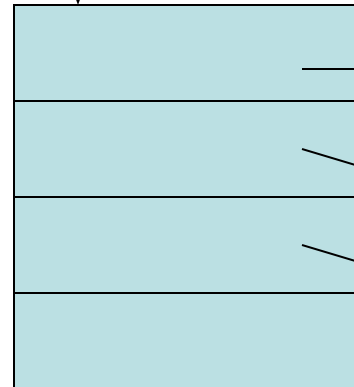
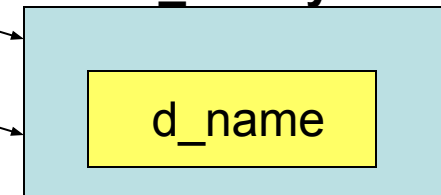# *Each task opens its own files*

struct task_struct

files

struct files_struct

next_fd

fd[ ]
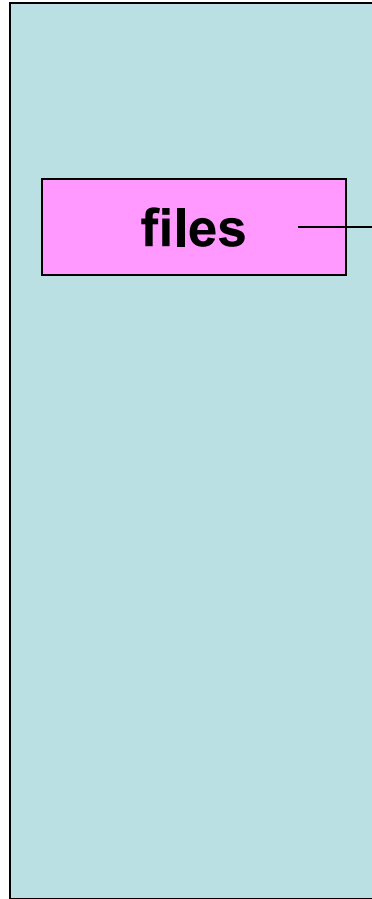
struct file

struct_dentry

d_name

struct_dentry

d_name

LFS Features

- Specifying paths
- Partition, Directories, and Drives
- Case Sensitivity
- File Extensions
- Hidden files

Leafpad Editor

- **Leafpad is a too simple text editor and its ideal replacement for popular Notepad application.**

- **Based on GTK+ (Gimp Tool Kit)**

- **Minimal dependencies**

- **Leafpad has all the basic text options available, e.g. Print, Search, replace, Line Numbers, Word Wrap, Auto Indent and Font.**

- **It also has capability to detect codeset, unlimited undo-redo and drag-n-drop capabilities**
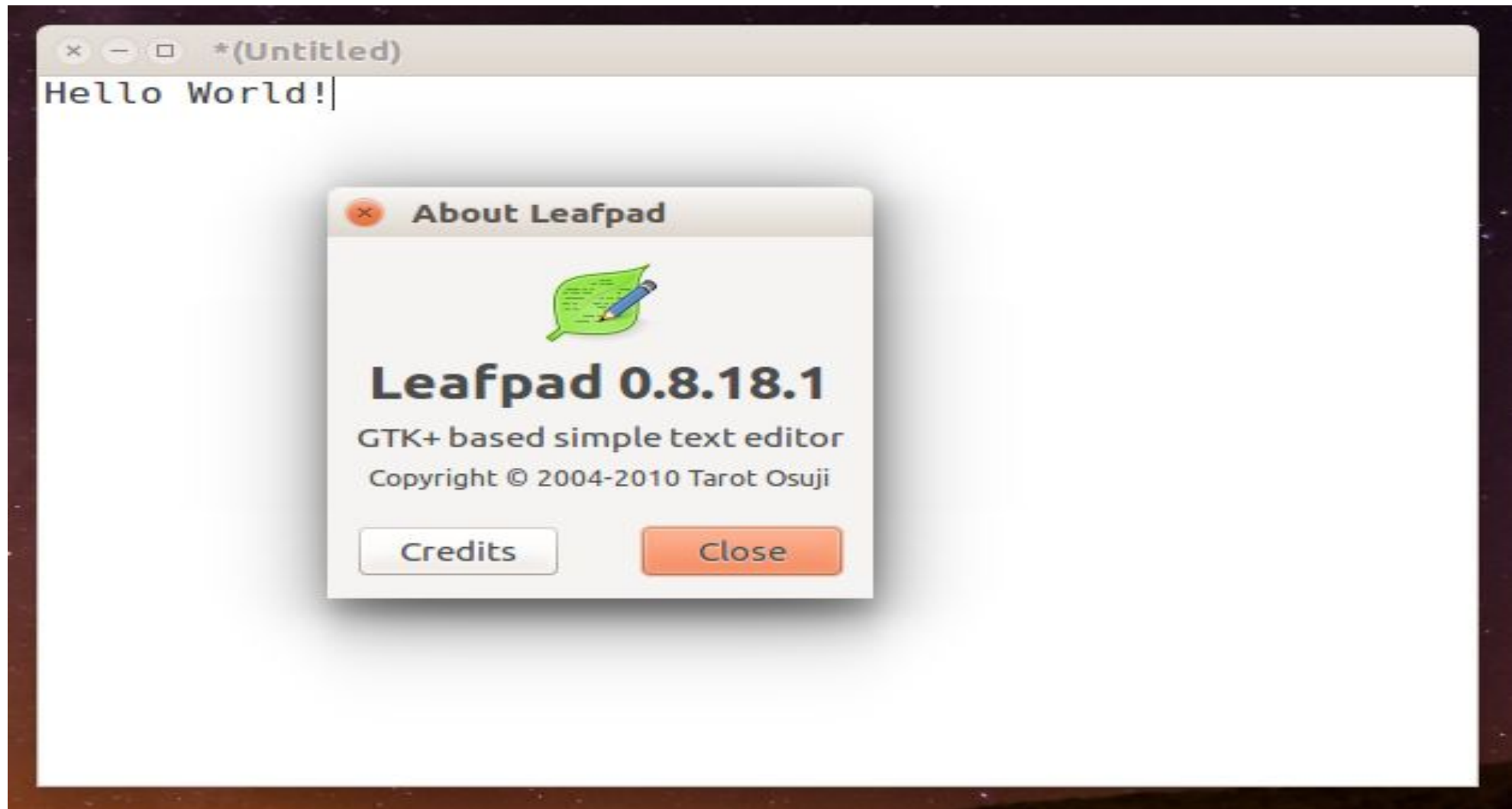
Installation & Running Leafpad

- Leafpad comes as a default text editor for Lightweight distributions and desktop environments.

- It comes as default application for LXDE, XUBUNTU and LBUNTU.

- However, if you are using vanilla Ubuntu, you can install it via running the below command from terminal.

sudo apt-get install leafpad
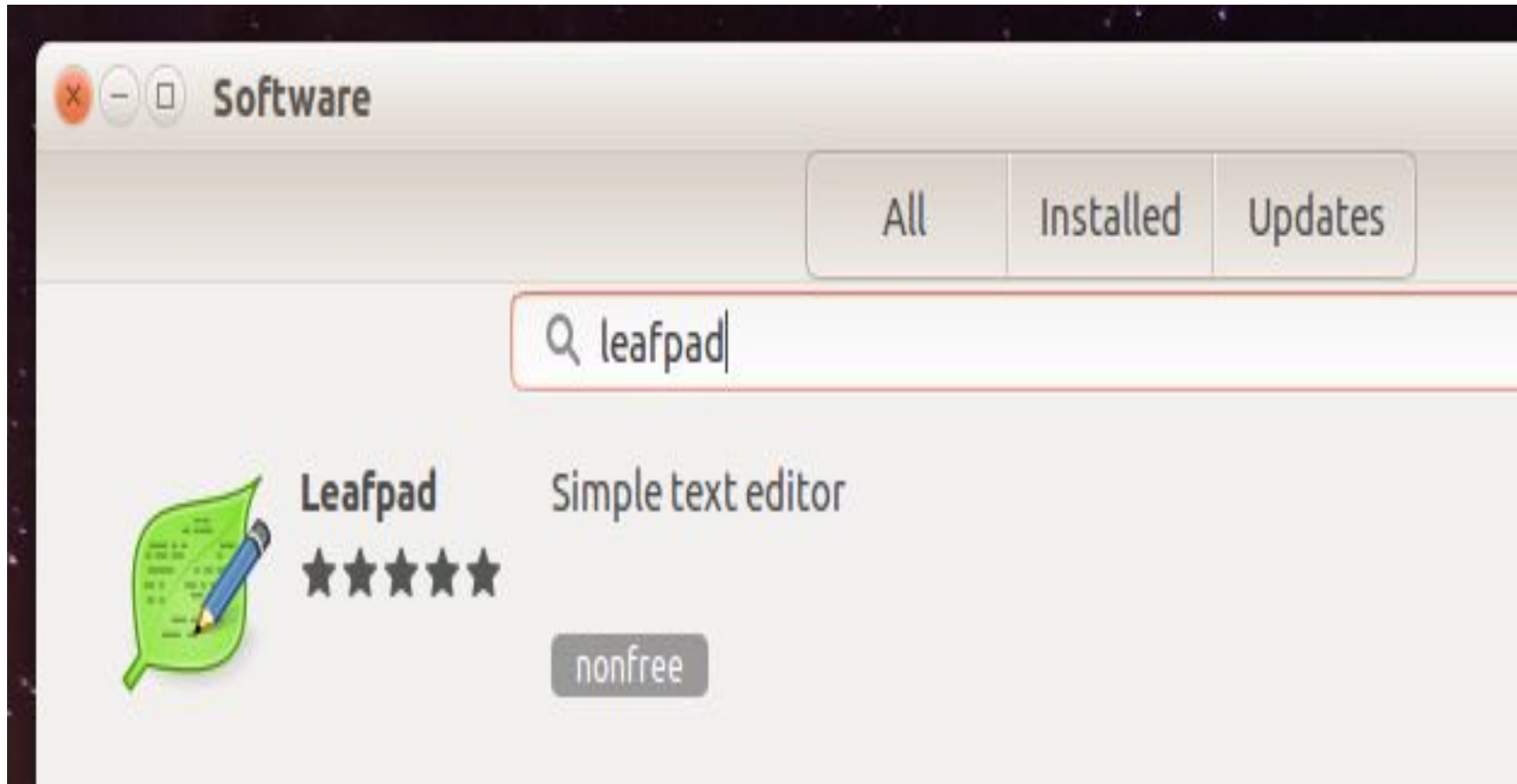
Install Leafpad Using GNOME Software

Leafpad Running in Ubuntu

File   Edit   Search   Options   Help

New	Ctrl+N
Open...	Ctrl+O
Save	Ctrl+S
Save As...	Shift+Ctrl+S
Print...	Ctrl+P
Quit	Ctrl+Q

```
y>sans-serif</family>
r>
        <family>Lucida</family>
er>


y>monospace</family>
r>
        <family>LucidaTypewriter</family>
        </prefer>
    </alias>
    <match target="font">
        <edit name="autohint" mode="assign">
            <bool>true</bool>
        </edit>
        <edit name="rgba" mode="assign">
            <const>rgb</const>
        </edit>
    </match>
</fontconfig>
```

Useful Raspberry Pi Commands

- $ apt-get update
- $ apt-get upgrade
- $ apt-get dist-upgrade
- $ rpi-update
- $ apt-get install <package>
- $ apt-get install phpmyadmin
- $ apt-get remove <package>
- $ apt-get remove phpmyadmin
- $ update-rc.d ssh enable
- $ update-rc.d -f ssh remove
- ps aux

Useful Raspberry Pi Commands

- $ mkdir new_directory
- $ mv /home/pi/documents/file.txt /home/pi/landing/
- $ mv oldfile.txt newfile.txt
- $ rm samplefile.txt
- $ rmdir sample_directory
- $ touch example.txt
- $ ifconfig
- $ ifconfig wlan0 172.168.1.0 (IP)

Important Linux commands

- ls (ls −a / ls −l)
- cd
- pwd
- mkdir
- rmdir
- rm
- cp
- mv
- touch
- cat
- head (-n /)
- tail
- chmod
- chown
- ssh
- scp

How to add new users and add users to groups

- *If multiple people are using the same raspberry pi you may want to add new users so each person has their own user account.*

- *Adding a new user is simple enough, the following method was done whilst logged in as the default user 'pi' which has the root or 'sudo' privileges needed to add and modify user accounts.*

- *For this example we will create a new user called 'newuser', simply replace this with whatever name you want. To add the user 'newuser' use the following command:*

- *sudo adduser newuser*

```
pi@raspberrypi ~ $ sudo adduser newuser
Adding user `newuser' ...
Adding new group `newuser' (1002) ...
Adding new user `newuser' (1001) with group `newuser' ...
Creating home directory `/home/newuser' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for newuser
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
```

- If you ever need to change the password of the 'newuser' you can do so with the command:

sudo passwd newuser

- In order for the user to have the same privileges as the default account 'pi' we'll need to add the it to several user groups. To find out what groups the user 'pi' belongs to type the following command:

groups pi

```
pi@raspberrypi ~ $ groups pi
pi : pi adm dialout cdrom sudo audio video plugdev games users netdev input indiecity
```

- pi belongs to several groups such as 'audio' and 'video'

```
pi@raspberrypi ~ $ groups newuser
newuser : newuser
```

- So to add 'newuser' to the group 'dialout' type:
- *sudo adduser newuser dialout*

```
pi@raspberrypi ~ $ sudo adduser newuser dialout
Adding user `newuser' to group `dialout' ...
Adding user newuser to group dialout
Done.
```

```
pi@raspberrypi ~ $ groups newuser
newuser : newuser adm dialout cdrom sudo audio video plugdev games users netdev input indiecity
```

Deleting a user and removing a user from a group

- If you want to remove a user from a group you can use the following command:

sudo deluser

- So to remove user 'newuser' from the group 'adm' you would type:

sudo deluser newuser adm

- A similar command can be used to delete an entire user along with it's home directory:

sudo deluser

Creating a new user with proper privileges

Let's begin by creating a user first (in this case `my_user`),

```
1    $ sudo adduser my_user
```

After that, you will have to set a password and provide some basic info for the new user.

Now we need to find out which groups the `pi` user belongs. For that run the following command,

```
1    $ sudo groups pi
```

You get an output like this,

```
1    pi : pi adm dialout cdrom sudo audio video plugdev games users input netdev spi i2c gpi
```

The next step is to add the new user (`my_user`) to all the above groups,

```
1    $ sudo usermod -aG adm,dialout,cdrom,sudo,audio,video,plugdev,games,users,input,netdev,
```

We should try to see whether we can log in as the user and run a `sudo` command,

```
1    $ sudo login -f my_user
2    $ sudo ls /tmp
```

Deleting the pi user

The last step is to remove the `pi` user from the system. For that, we login as the new user and run the following command,

```
1   $ sudo userdel pi
```

If you get an `userdel: user pi is currently used by process 2030` error message, either you have to kill the process or can force delete the user,

```
1   $ sudo userdel -f pi
```

You may get the exact error again, but the user should be removed already. You can confirm it by trying to log in as the `pi` user.

```
1   $ sudo login -f pi
```

It should give an `Authentication failure` error.

Finally, let's remove the `pi` home directory (Backup all files that needed beforehand),

```
1   $ sudo rm -rf /home/pi
```