



Jawaharlal Nehru Technological University Kakinada
University College of Engineering Vizianagaram

Department of Computer Science and Engineering



Scire

Know More

25

years of

LINUX



hadoop



cassandra



node



MySQL



mongoDB



Java

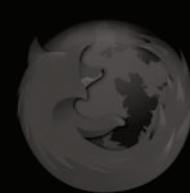


CentOS

HAPPY BIRTHDAY



ubuntu



Audacity



redis



git



python



WORDPRESS



FEB
2017



Magento

Chief Patron

Prof.V.S.S.Kumar
Hon'ble VC
JNTUK

Patron

Dr.Ch.Sai Babu
Registrar - Incharge
JNTUK

Co-Patron

Prof V.Sreenivasulu
Principal
JNTUK-UCEV

Honorary Editor

Prof.R.Rajeswara Rao
HOD - CSE

Chief Editor

Mr.T.Siva Rama Krishna
Asst.Prof - CSE

Creative Head

Mr.P.Arjun Kesava
IV B.Tech., CSE

Editorial Board

Mr.T.Naga Ravi Teja
IV B.Tech., CSE

Mr.K.Sumanth
III B.Tech., CSE

Ms.G.Bhavya Sri Harika
III B.Tech., CSE

Advisory Board

Prof.D.Rajya Lakshmi
Prof.A.S.N.Chakravarthy
Mr.N.Venkatesh
Mrs.P.Arunkumari
Mr.S.Radha Krishna
Mrs.S.Surekha
Mr.R.D.D.V.Siva Ram



Principal's Message

As the Head of the Institution, its my duty to ensure that students are excelling in various fields and presenting their ultimate talents. The Department of CSE made a beautifully crafted and outstanding work on <SCIRE> magazine. I feel departmental magazine is a way of building relations among students, its a feast to viewers and inculcates the habits of reading and writing, which are essential parts of communication. With this second edition of <SCIRE>, I wish them good luck for the future in advance. It's my pleasure to be a part of his visual wonder. On the whole <SCIRE> is one of a kind.

Wishing you all the best...!

Prof. V. Sreenivasulu

Principal

HOD's Message

We are very proud to release our second edition of <SCIRE> on the eve of our technical fest – CRESENSE. This magazine is the fruitful work of our faculty and students who thrive for excellence in all areas to give out an unparalleled experience. This magazine is our check-point, on a glance gives out a crystal clear knowledge about how far we have come together as a family. Its displays the talents of our students and how supportive our faculty are in curricular and co-curricular activities. I personally, feel very privileged to announce, we are going to make more of these magazines in the forth coming future. And to be frank, I can't wait to unleash the next edition.

Miracles do happen and <SCIRE> is one of them.

Prof. R. Rajeswara Rao

HOD CSE

Constelle Association Members

President

Mr.P.Dhanraj
IV B.Tech., CSE

Vice President

Ms.D.Sameera
IV B.Tech., CSE

Secretary

Mr.Sk.Faruk
III B.Tech., CSE

Joint Secretary

Ms.G.Bhavya Sri Harika
III B.Tech., CSE

Treasurer

Mr.K.Sumanth
III B.Tech., CSE

Executive Committee

Mr.S.Dhawal
III B.Tech., CSE

Mr.K.Vasanth
II B.Tech., CSE

Ms.K.Teena Reshmanjali
II B.Tech., CSE

CONTENTS



GitHub

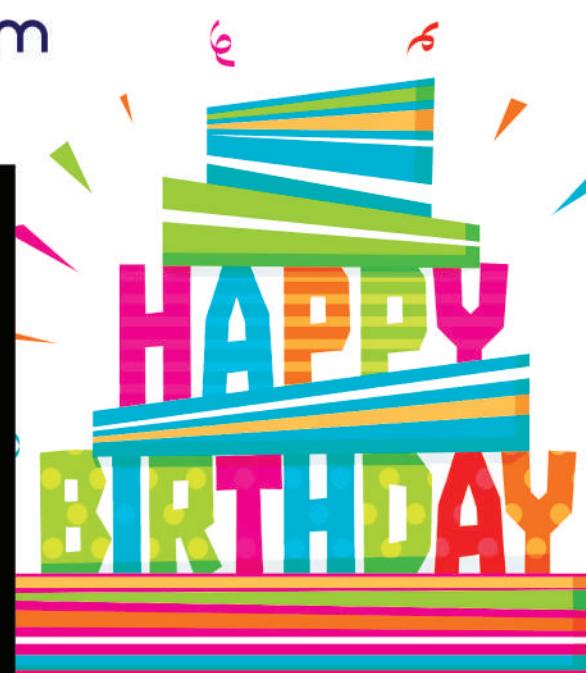


- 1 Department Breakthroughs
- 2 Student Achievements
- 3 'What and Why' on Linux
- 5 'How and Who' on Linux
- 7 The Hidden Face
- 8 Others vs Linux
- 9 Story of Linux
- 11 Composition vs Inheritance
- 13 Saakshar Organisation
- 15 Top 25 Facts on Linux
- 17 GitHub
- 18 Life - A Program
- 19 CSE Timeline

"When I read commentary about suggestions for where C should go, I often think back and give thanks that it wasn't developed under the advice of a worldwide crowd"

- Dennis Ritchie

[1941 - 2011]





DEPARTMENT BREAKTHROUGHS



A Two Day National Workshop on Fog Computing and Internet of Things was conducted on 28th and 29th March 2016. This event was co-sponsored by Government of Andhra Pradesh, Microsoft, Particle (USA) and Computer Society of India.

UPGRADATION

On campus Internet bandwidth was upgraded from 20 Mbps to 65 Mbps with an addition of 45 Mbps Railtel connection, inaugurated by Prof. V. Sreenivasulu, Principal.



INTERNSHIP

15 B.Tech. Students and 05 M.Tech. Students have completed/undergoing internships at prestigious organizations like Teradata, Vizag Steel Plant and NIT Jaipur.

Internet of Things Research facility was started in the department with Raspberry Pi 3, Arduino and Particle Photon boards.



CAMPUS RECRUITMENT

Till date 25 final year Students are recruited into various IT Firms, and still counting.

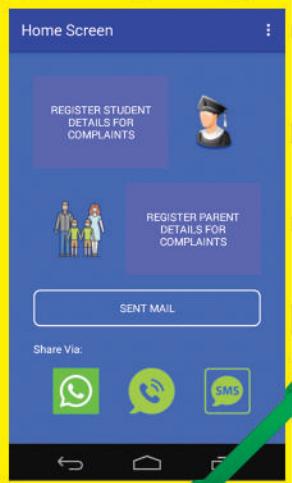
Ms. Navya recruited into **Teradata**, Hyderabad with a package of Rs: 7.5 lakhs per annum

Mr. Jayadhar recruited into **Uurmi Systems**, Hyderabad with a package of Rs: 6.0 lakhs per annum

UCEV WATCH APP

Android

Mr. K. L. M. Kumar of IV B.Tech. developed a mobile application to prevent ragging in the campus. This app enables students to anonymously report ragging incidents to Wardens and Principal, along with their location through GPS tracking.



STARS OF THE YEAR



A
C
H
I
E
V
E
M
E
N
T
S

- ▶ Yegreddi Sankar Rao was selected for **ATHLETICS** in **ALL INDIA INTER UNIVERSITY LEVEL**.
- ▶ Kondeti Naveen won **BRONZE** in **ZONAL LEVEL ATHLETICS** held at **JNTU-KAKINADA**.
- ▶ Megavath Bheem Sen Nayak was part of the **KABADDI** team which bagged **1st Place** in **CENTRAL ZONAL LEVEL**.
- ▶ Yegreddi Sankar Rao won **BRONZE** in **ZONAL LEVEL ATHLETICS** held at **JNTU -KAKINADA**.
- ▶ G.V.Bhavya Sri Harika won **FIRST** place in **INTER COLLEGIATE CHESS CHAMPIONSHIP** held at **JNTU KAKINADA**.
- ▶ P. Vara Lakshmi won **FIRST** place in **INTER COLLEGIATE CHESS CHAMPIONSHIP** held at **JNTU-KAKINADA**.
- ▶ Y. Vivek won **FIRST** place in **SEMINAR ON IOT** held on the eve of **ENGINEERS DAY** in **JNTUK UCEV**.



A
C
H
I
E
V
E
M
E
N
T
S

- ▶ V.Harika was the honorary achiever of **INTERNSHALA VTC YOUNG ACHIEVER SCHOLARSHIP 2016**.
- ▶ E. Navya earned **3rd PLACE IN UTKRANTHI 2K16 - IIT MADRAS (ALL INDIA CODING CHAMPIONSHIP)**
- ▶ P. Ravi Kiran won **1st PLACE IN ZEITGEIST 2K16 – JNTU KAKINADA (ONLINE PROGRAMMING CONTEST- HACKERRANK)**.
- ▶ S.G. Anirudh, T.N. Ravi Teja, G.Mohan Krishna and D. Santosh Gandhi received **CERTIFICATE OF ACHIEVEMENT IN ACM-ICPC MULTI-SITE REGIONAL CONTEST**.

‘What & Why’ on LINUX

WHAT

What is Linux?

Just like Windows XP, Windows 7, Windows 8, and Mac OS X, Linux is an operating system. An operating system is software that manages all of the hardware resources associated with your desktop or laptop. To put it simply – the operating system manages the communication between your software and your hardware. Without the operating system (often referred to as the “OS”), the software wouldn’t function.

The OS is comprised of a number of pieces:

The Bootloader: The software that manages the boot process of your computer. For most users, this will simply be a splash screen that pops up and eventually goes away to boot into the operating system.

The kernel: This is the one piece of the whole that is actually called “Linux”. The kernel is the core of the system and manages the CPU, memory, and peripheral devices. The kernel is the “lowest” level of the OS.

Daemons: These are background services (printing, sound, scheduling, etc) that either start up during boot, or after you log into the desktop.

The Shell: You’ve probably heard mention of the Linux command line. This is the shell – a command process that allows you to control the computer via commands typed into a text interface. This is what, at one time, scared people away from Linux the most (assuming they had to learn a seemingly archaic command line structure to make Linux work). This is no longer the case. With modern desktop Linux, there is no need to ever touch the command line.

Graphical Server: This is the sub-system that displays the graphics on your monitor. It is commonly referred to as the X server or just “X”.

Desktop Environment: This is the piece of the puzzle that the users actually interact with. There are many desktop environments to choose from (Unity, GNOME, Cinnamon, Enlightenment, KDE, XFCE, etc). Each desktop environment includes built-in applications (such as file managers, configuration tools, web browsers, games etc).

Applications: Desktop environments do not offer the full array of apps. Just like Windows and Mac, Linux offers thousands upon thousands of high-quality software titles that can be easily found and installed. Most modern Linux distributions (more on this in a moment) include App Store-like tools that centralize and simplify application installation. For example: Ubuntu Linux has the Ubuntu Software Center which allows you to quickly search among the thousands of apps and install them from one centralized location.

What is the difference between Unix and Linux?

You may have heard of Unix, which is an operating system developed in the 1970s at Bell Labs by Ken Thompson, Dennis Ritchie, and others. Unix and Linux are similar in many ways, and in fact, Linux was originally created to be similar to Unix. Both have similar tools for interfacing with the systems, programming tools, filesystem layouts, and other key components. However, Unix is not free. Over the years, a number of different operating systems have been created that attempted to be “unix-like” or “unix-compatible,” but Linux has been the most successful, far surpassing its predecessors in popularity.

Why

Why use Linux?

This is the one question that most people ask. Why bother learning a completely different computing environment, when the operating system that ships with most desktops, laptops, and servers works just fine? Before getting answer for that question, one should pose another question. Does that operating system you're currently using really work "just fine"? Or are you constantly battling viruses, malware, slow downs, crashes, costly repairs, and licensing fees?

If you struggle with the above, and want to free yourself from the constant fear of losing data or having to take your computer in for the "yearly clean up," Linux might be the perfect platform for you. Linux has evolved into one of the most reliable computer ecosystems on the planet. Combine that reliability with zero cost of entry and you have the perfect solution for a desktop platform.

• ZERO COST OF ENTRY:

That's right, zero cost of entry...as in free. You can install Linux on as many computers as you like without paying a cent for software or server licensing.

Let's take a look at the cost of a Linux server, in comparison to Windows Server 2012. The price of the Windows Server 2012 software alone can run up to \$1,200.00 USD. That doesn't include CALs, and licenses for other software you may need to run (such as a database, a web server, mail server, etc). With the Linux server...it's all free and easy to install. In fact, installing a full blown web server (that includes a database server), is just a few clicks or commands away (take a look at "Easy LAMP Server Installation" to get an idea how simple it can be).

If you're a system administrator, working with Linux is a dream come true. No more daily babysitting servers. In fact, Linux is as close to "set it and forget it" as you will ever find. And, on the off chance, one service on the server requires restarting, re-configuring, upgrading, etc...most likely the rest of the server won't be affected.

• TROUBLE FREE ENVIRONMENT:

Be it the desktop or a server, if zero cost isn't enough to win you over – what about having an operating system that will work, trouble free, for as long as you use it? Most of the users who have personally used Linux for nearly twenty years (as a desktop and server platform), have not reported any issue with malware, viruses, or random computer slow-downs. It's that stable. And server reboots? Only if the kernel is updated. It is not out of the ordinary for a Linux server to go years without being rebooted. That's stability and dependability.

Linux is also distributed under an open source license. Open source follows the following key philosophies:

- The freedom to run the program, for any purpose.
- The freedom to study how the program works, and change it to make it do what you wish.
- The freedom to redistribute copies so you can help your neighbor.
- The freedom to distribute copies of your modified versions to others.

The above are crucial to understanding the community that comes together to create the Linux platform. It is, without a doubt, an operating system that is "by the people, for the people". These philosophies are also one of the main reasons a large percentage of people use Linux. It's about freedom and freedom of choice.

‘How & Who’ on LINUX

HOW

How was Linux created?

Linux was created in 1991 by Linus Torvalds, a then-student at the University of Helsinki. Torvalds built Linux as a free and open source alternative to Minix, another Unix clone that was predominantly used in academic settings. He originally intended to name it “Freak,” but the administrator of the server Torvalds used to distribute the original code named his directory “Linux” after a combination of Torvalds’ first name and the word Unix, and the name stuck.

How does Linux differ from other operating systems?

In many ways Linux is similar to other operating systems you may have used before, such as Windows, OS X, or iOS. Like other operating systems, Linux has a graphical interface, and types of software you are accustomed to using on other operating systems, such as word processing applications, have Linux equivalents. In many cases, the software’s creator may have made a Linux version of the same program you use on other systems. If you can use a computer or other electronic device, you can use Linux.

But Linux also is different from other operating systems in many important ways. First, and perhaps most importantly, Linux is open source software. The code used to create Linux is free and available to the public to view, edit, and—for users with the appropriate skills—to contribute to.

Linux is also different in that, although the core pieces of the Linux operating system are generally common, there are many distributions of Linux, which include different software options. This means that Linux is incredibly customizable, because not just applications, such as word processors and web browsers, can be swapped out. Linux users also can choose core components, such as which system displays graphics, and other user-interface components.

How can I contribute to Linux?

Most of the Linux kernel is written in the C programming language, with a little bit of assembly and other languages sprinkled in. If you’re interested in writing code for the Linux kernel itself, a good place to get started is in the Kernel Newbies, which will explain some of the concepts and processes you’ll want to be familiar with.

But the Linux community is much more than the kernel, and needs contributions from lots of other people besides programmers. Every distribution contains hundreds or thousands of programs that can be distributed along with it, and each of these programs, as well as the distribution itself, need a variety of people and skill sets to make them successful, including:

Testers to make sure everything works on different configurations of hardware and software, and to report the bugs when it does not.

Designers to create user interfaces and graphics distributed with various programs.

Writers who can create documentation, how-tos, and other important text distributed with software.

Translators to take programs and documentation from their native languages and make them accessible to people around the world.

Packagers to take software programs and put all the parts together to make sure they run flawlessly in different distributions.

Evangelists to spread the word about Linux and open source in general.

And of course developers to write the software itself.

WHO

Who uses Linux?

You're probably already using Linux, whether you know it or not. Depending on which user survey you look at, between one- and two-thirds of the webpages on the Internet are generated by servers running Linux.

Companies and individuals choose Linux for their servers because it is secure, and you can receive excellent support from a large community of users, in addition to companies like Canonical, SUSE, and Red Hat, which offer commercial support.

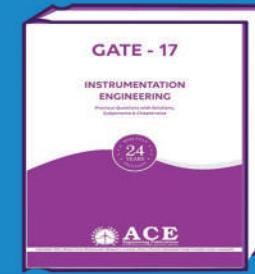
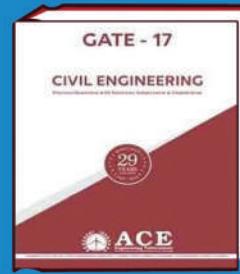
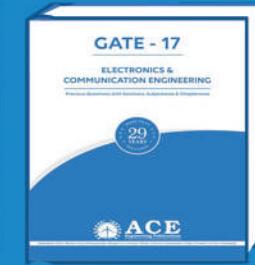
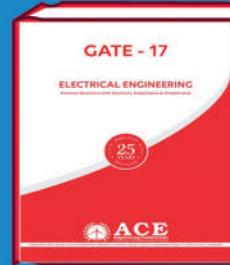
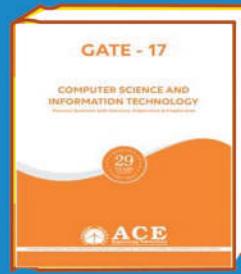
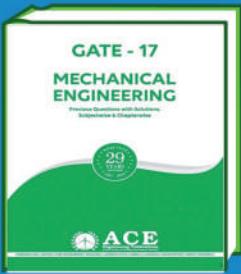
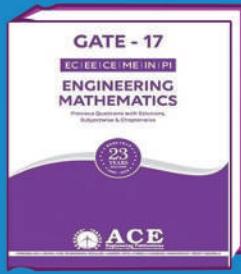
Many of the devices you own probably, such as Android phones, digital storage devices, personal video recorders, cameras, wearables, and more, also run Linux. Even your car has Linux running under the hood.

Who "owns" Linux?

By virtue of its open source licensing, Linux is freely available to anyone. However, the trademark on the name "Linux" rests with its creator, Linus Torvalds. The source code for Linux is under copyright by its many individual authors, and licensed under the GPLv2 license. Because Linux has such a large number of contributors from across multiple decades of development, contacting each individual author and getting them to agree to a new license is virtually impossible, so that Linux remaining licensed under the GPLv2 in perpetuity is all but assured.



ACE
Engineering Academy
(India's leading institute for IES/GATE/PSUs)



Hyderabad | Delhi | Bhopal | Pune | Bhubaneswar | Bengaluru
Lucknow | Patna | Chennai | Vijayawada | Visakhapatnam | Tirupati



The Hidden Face

Every person has two faces in this world, one is visible and other is invisible, the first face can be seen by everyone except you (in the mirror), the second face is visible to only you. Many of us are in the processs of discovering their second face. Successful people whom we see around us have already discovered their second face i.e. their own "TALENT". They had shaped it in the correct way and in their own passion which suits their style. Every person has their own talent in them but the problem is that they don't try to discover it, if we leave it like that it goes into the state of oblivion. So first we have to find that skill in which we are skilled at and try to pollish that skill in a way which is convenient to us. The hidden face can be discovered by our actions in our daily life, may be in sports, paintings, education or in any field etc. It should also be discovered in right time because it exo-expedites your way and gives boost to achieve goal. Discovering and using it in the correct way will provide you a path and a key to open your new dream world.

DISCOVER IT,
IMPLEMENT IT,
AND achieve your dream.....

- K. Sai Vasanth 2nd Year B.Tech

OTHERS vs LINUX



\$3,675



Autodesk Maya vs Blender



\$381.65



Adobe Acrobat X vs pdfcreator



\$15+ p/m



Microsoft Office vs LibreOffice



\$20+ p/m



Adobe Dreamweaver vs Bluefish



\$200



Microsoft Windows 8 vs Fedora



\$20+ p/m



GIMP

Adobe Photoshop vs GIMP



\$898+



MySQL

Microsoft SQL Server vs MySQL



\$500+



Microsoft Visual Studio vs KDevelop



\$4,195+



AutoCAD vs FreeCAD

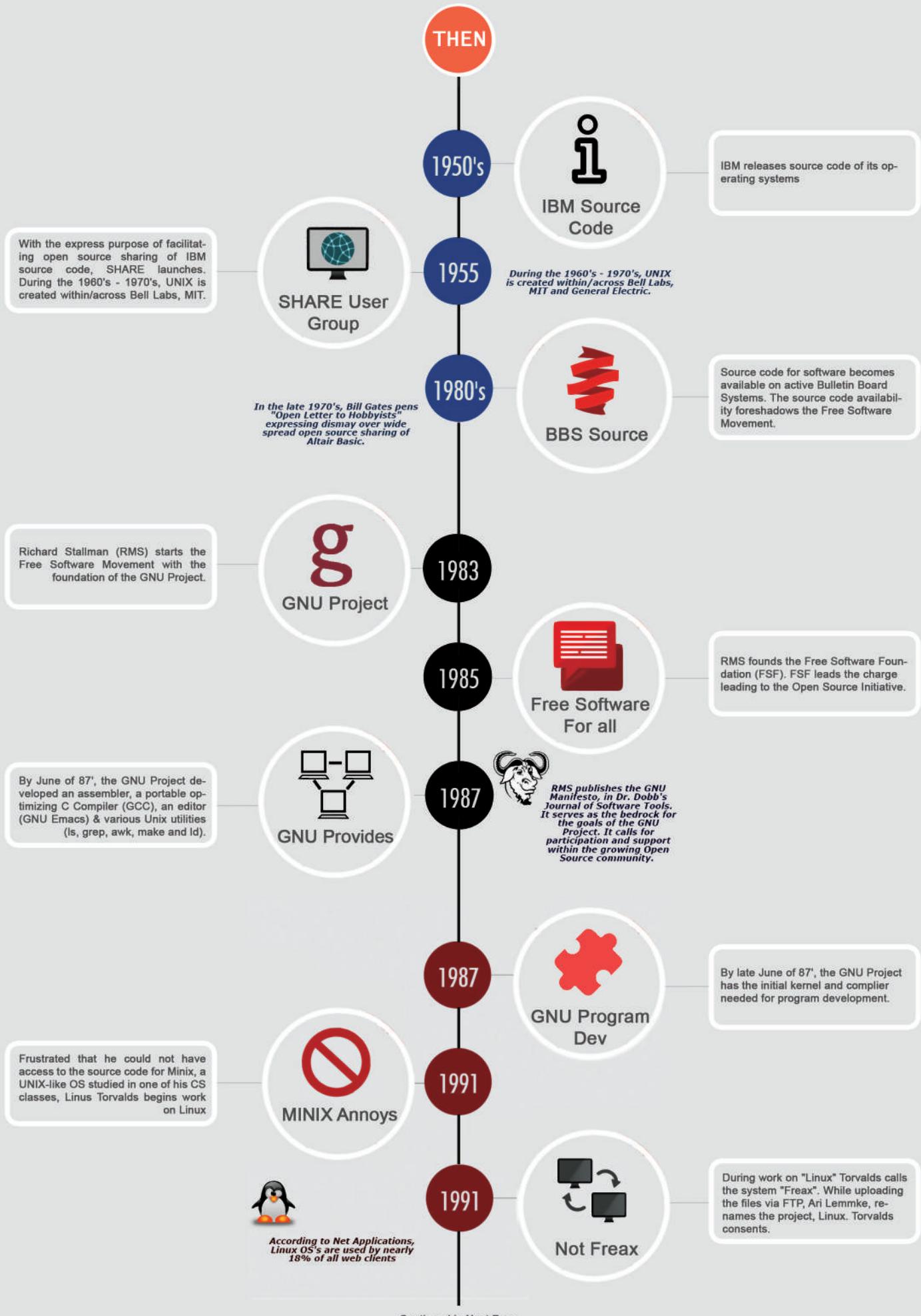


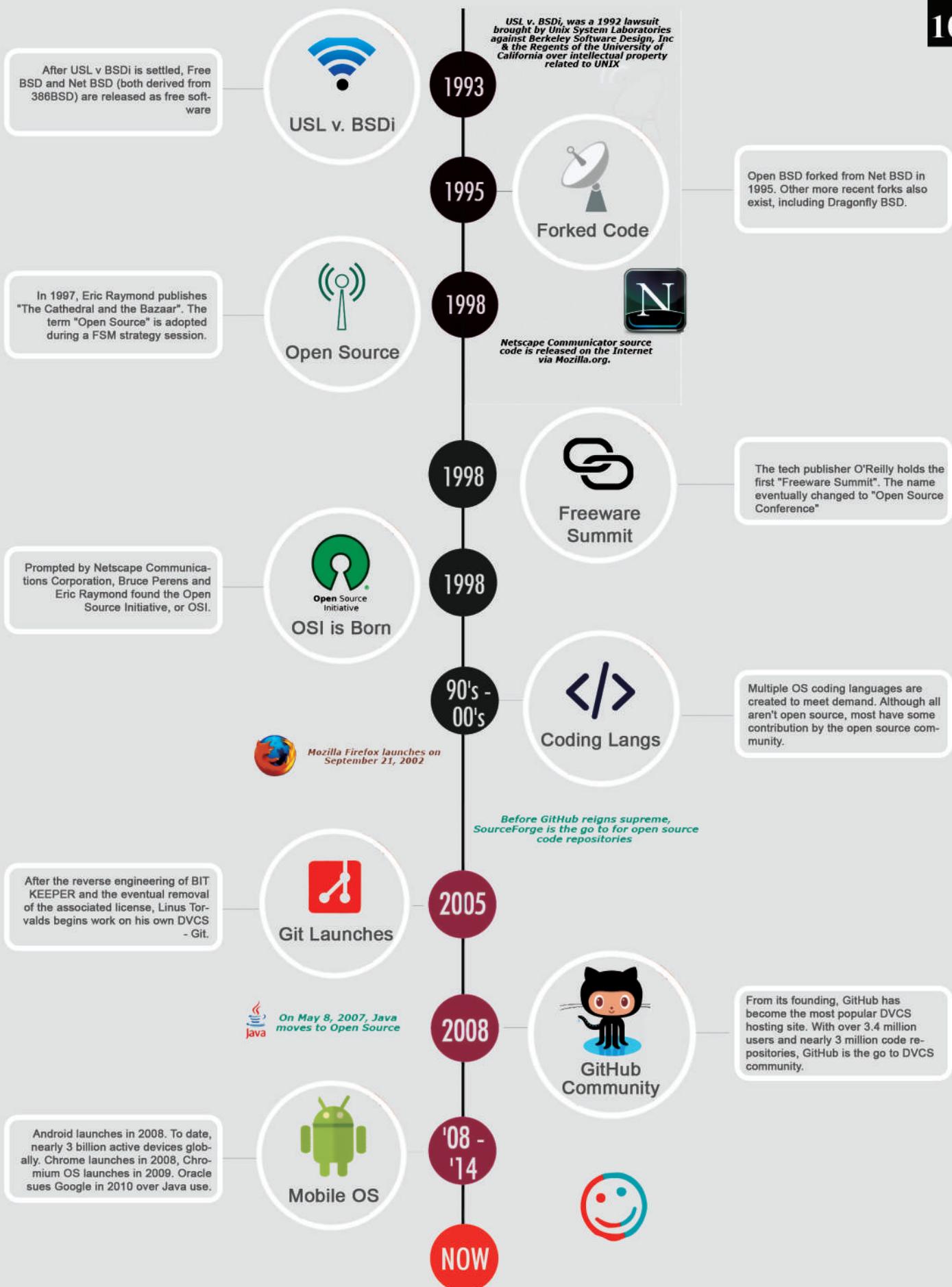
\$200



Audacity

Apple LogicPro vs Audacity





"While social change may occur as an unintended by-product of technological change, advocates of new technologies often have promoted them as instruments of positive social change." - Joel West

Composition v/s Inheritance

[Has-A] V [Is-A]



As a OO Programmer what do we seek from the language....??!!

As everyone knows we have a long list of requirements....

Let's briefly consider what all we need.....

>>MODULARITY: For maintaining large chunks of code

>>GOOD FRAMEWORK: For code libraries where supplied software component can be easily adapted a modified by the programmer.

>>ABSTRACTION: For building real life application of world

>>CODE REUSABILITY: For preventing user from writing code from scratch every time...

Now let's **LOOK INTO CODE REUSE.**

In Dark Ages code reuse was just a copy-paste mechanism, which leads to a lot of confusion as separation of code from main logic is obviously an unwise act. Time passed by and we are here with OOP..., which introduced two aspects of code reuse:

- 1] **Inheritance**
- 2] **Composition**

Let's look into the uninteresting definitions of both the features and understand them in an interesting way.....

[Pre-requisites: Idea about Class, Field & Method]

Inheritance: A class may inherit (in-built characteristics) like fields and method of its super/parent class and there is also additional capability to sub-classes for over-riding the methods. And its fundamental for all OOP languages.[Is-A Relationship]

Easy View: We design our types on what they are.

HOW INHERITANCE IS USED ?

First we take out all the common features/properties of the object and define them in a parent class and rest of the additional features/properties will be added to the sub-classes.....

Ex: Dogs, Cat. Both are animals. Both of them eat, sleep and drink. Hence ANIMAL is parent class of both having features/properties EATING, SLEEPING, DRINKING. When it comes to DOGS they bark and when it comes to CATS they purr.

Dog IS-An ANIMAL and Cat IS-An ANIMAL. Which indeed imply they both need to EAT, BREATH and DRINK. Hence, ANIMAL is super class of both cats and dogs.

```
class DOGS(ANIMAL):
    def __init__(self):
        #Do something
    def BARK():
        print("Bow-Bow")
        #I am Barking
class CATS(ANIMAL):
    def __init__(self):
        #Do something
    def PURR():
        print("Meow-Meow")
        #I am Purring
```

Composition: As when the class uses another object to provide some or all of its functionality. Even this is Fundamental to all OOP languages.[Has-A relationship]

Easy View: We design our types on what they do.

We can see Composition in our daily routine car has an engine, wall is composed of brick and mortar and so on.

HOW COMPOSITION IS USED???

First we build the features and later we integrate them to make things

Ex:

Building a Car and a Bike (Although funny consider both them have same type of engine)

First we Build a Engine having feature to convert fuel to heat. Later we make tyres. BIKE Has-An ENGINE, BIKE Has 2 TYRES. First we build raw parts later we integrate them. In similar fashion CAR Has-An ENGINE, Car Has 4 TYRES.

What to choose??? When to Choose??

Inheritance can be used when

- 1] Subclass must be proper subset of super-class
- 2] Implementation of super-class is necessary or appropriate for subclass

Composition can be used in all the application which is a far better approach compared to the Inheritance. Which states Has-A relation is better than Is-A relation. Situations Where We Cannot Use Inheritance.

Let's Suppose we are gods we design living and non-living things by programming them..... [Programming using Inheritance]. First I design a Dog based on the following specification/features

```

class LIVING_THING:
    def __init__(self):
        #I live
    def EAT():
        #EAT
    def BREATH():
        #BREATH
    def DRINK():
        #DRINK

class DOG(LIVING_THING):
    def __init__(self):
        #I live, bark and run
    def BARK():
        print("Bow-Bow")
    def RUN():
        #I am Running
    def DIGEST():
        #I am digesting what I have eaten ;)

```

Living is a super class, Dog has special features that is to bark, run and Digest.

Then I decide to design a Robot based on the specifications/features.

```

class NON_LIVING_THING():
    def __init__(self):
        #I dont live and i dont need to eat breath and drink
class ROBOT(NON_LIVING_THING):
    def __init__(self):
        #I dont live, but i clean and run
    def CLEAN():
        #I am Cleaning
    def RUN():
        #I am Running

```

Non-Living is Super class, Robot has ability to clean and run.

Now, I've decided to create a ROBOT-DOG, obviously it would be a ROBOT and It would be looking like a DOG. Hence, I inherit the features of DOG and ROBOT to my ROBOT-DOG .

```

class ROBO_DOG(DOG,ROBOT):
    def __init__(self):

```

But now if we look at the ROBOT-DOG I have programmed. More absurdly it can digest the food it takes perhaps it must malfunction by doing so. And it's is situation Inheritance fails.

[Programming using Composition]

First I design all feature separately Eating, Drinking, Barking, Cleaning, Running and Digesting. When I need to design a Living Dog all I do is integrate features like barking, digesting, eating, drinking and running.

Whenever I need to design a Robot-Dog all I do is integrate features like barking, running and cleaning.

```

class RUN:
    def __init__(self):
        #Do something
    def RUN():
        #I am Running

class CLEAN:
    def __init__(self):
        #Do something
    def CLEAN():
        #I am Cleaning

class BARK:
    def __init__(self):
        #Do something
    def BARK():
        #I am Barking

class ROBO_DOG(RUN,BARK,CLEAN):
    def __init__(self):
        #Do something
    def WORK():
        #Run, Clean and Bark

```

HENCE CHOOSE
COMPOSITION
OVER
INHERITANCE
WHEN EVER POSSIBLE

- S. Dhawal 3rd Year B.Tech

HUB
FOR YOUTH
GOKARTING



call now to know more

+91 84998 88872

SAAKSHAR ORGANISATION

- By IIIrd Year B.Tech.



Saakshar is an organisation that aims to educate underprivileged children in Government Schools. Students from third year are coming forward and joining hands with us as a volunteer which is highly appreciated. They are helping us to propagate our message easily.

The student volunteers are philanthropic and benevolent. They attend regular weekday classes in colleges/universities. But spend their Saturdays to teach under-privileged children in Government schools. This type of attitude is not common but we are fortunate to have such wonderful volunteers. They are trained by the organisation to teach those under-privileged children using innovative methodologies. To add on, whatever roles they get into, they become a leader in that role and promote for the cause on behalf of Saakshar.

The skills of the student volunteers are put to real test every week. They are really helping in expansions which are directly resulting in helping the children learn at larger extent. They also give seminars on Leadership, Communication Skills, Stress Management, and Informed Decision Making to the children on Saturday apart from standard textbook syllabus. The children well in all spheres and by this volunteers are helping us in tomorrow.

The efforts student -teers are

and sincerity of the volun
highly

appreciated and recommended. We are wishing them a bright and successful future ahead.

At Saakshar, our students go to Government Schools on Saturdays and teach the children using innovative methodologies. By teaching we do not mean only Standard Textbook Syllabus, but we also mean useful Life Skills, which means Problem Solving, Stress Management, Leadership, Basic Arithmetic, Communication Skills, and Informed Decision Making and so on.

At Saakshar Organisation, we believe in honesty, work ethics, passion and most importantly the necessity to make mistakes. We believe that making mistakes is an essential part to improve ourselves in whatever we do. Our family is ready to welcome new members. We will be very privileged to have you work with us for a better tomorrow. Do get in touch with us at sakshar.org@gmail.com. People can help us in different stages, by spreading the word, volunteering with us.

We are a team of passionate individuals who share a common vision & are willing to bring a change in the conventional education model, especially amongst underprivileged children, helping towards the building of a better tomorrow! We feel Change starts with us & so do you. Do your part and make a difference.





Our student volunteers went to teach in Primary Government schools located in Gajularega and Bondapalli on Saturdays. Our student volunteers went to two schools totally and imparted knowledge for 8 weeks by taking basic arithmetic sessions, motivational sessions, craft sessions and communication skills sessions.



Advertisements

TEXMART

MEN | WOMEN | KIDS | HOME

Texmart offers complete wardrobe solution for Women, Men & Kids Range. You will find womens casual and ethnic range, Mens formal and casual range. 7 Stores located in coastal Andhra Pradesh. We fulfill all the garment needs of a family at one place, a comfort which surpasses all other shopping experiences. We are a one-stop shop for women ethnic wear, fabrics for men, home textiles and other textile needs. One who buys our brand gets addicted.

OUR COLLECTIONS



TEXMART STORES

Vijayawada | Rajahmundry | Srikakulam
Vizianagaram | Gajuwaka | Anakapalle | Amalapuram

15 TOP 25 FACTS on LINUX

1

Linux is not an OS, but it is the kernel, gnu Linux is the OS and it comes in several hundred flavours.

2

Linux kernel was written by a 21 year Finnish college student as a part of his hobby. Yup! His name is Linus Torvalds.

3

Torvalds created Linux based on GNU general public license (gpl). Perhaps Torvalds would have never written his own kernel if gpl would be having its own kernel and driver.

4

Major part of today's Linux kernel is written in c programming language and assembly language and only 2%of today's kernel contains code written by Torvalds.

5

A standard Linux kernel of today has over 10 million lines of code and it grows at the rate of 10% every year. About 4500 lines of codes are added and 1500 lines of code are changed every day. Initially in 1991, Linux kernel version 0.01 was released with 10239 lines of code.

6

A guy named William Della Croce jr. registered the name Linux and demanded royalty for using its name and mark. However he agreed to handover the trademark to Linus, later.

7

The Linux kernel's official mascot is a penguin named tux, abbreviation of tuxedo. The idea that Linux had a pet penguin comes from Linus Torvalds himself.

8

The first commercial distribution of gnu / Linux was Yggdrasil (http://en.wikipedia.org/wiki/yggdrasil_linux/gnu/x) and was launched in cd format in 1992. Red hat was one of the first distributions to settle within companies and data centres in 1999.

9

Debian was one of the first gnu / Linux that was constituted and organized as a community of developers. Debian v. 4.0's source code contains 283 million lines of code, \$7.37 billion: projected cost to produce that amount of code in a commercial environment. Debian's code base remains the foundation for other distros such as Ubuntu, knoppix and xandros.

10

90% of the world's most powerful supercomputers are using gnu/Linux. Top ten of supercomputers use Linux. 33.8% of the world runs on Linux servers compared to 7.3% running Microsoft windows operating system.

11

There are over 300 distributions gnu / Linux activities ranging from the well-known Debian or fedora distributions through governmental or educational level.And this list seems to grow with regional and personal distros being added frequently.

12

An unmodified version of the Linux kernel is called – "vanilla kernel".

13

Linux Torvalds has been honoured by naming an asteroid after his name.

Must Know Facts

16

14

For those who think Linux can't do animation - Oscar-winning visual effects of the titanic by James Cameron came from machines with Linux and avatar was the last movie completely developed in 3d applications on Linux platform using Foss software. Exclaimed!

15

Believe it or not - in 2002, Microsoft had accumulated a \$ 421 million cost of fighting the spread of Linux, according to the register.

16

According to a study funded by the European Union, the estimated cost to redevelop the most recent kernel versions would be at \$1.14 billion usd - amazed.

17

Microsoft windows and the Linux kernel can run simultaneously in parallel on the same machine using a software called cooperative Linux (colinux).

18

IBM choose Linux for what is expected to be the world's most powerful supercomputer, sequoia, due in 2011.

19

Area of application of Linux - US department of defense, Japan's bullet trains, traffic control of san Francisco, the new York stock exchange, CERN, control of nuclear reactors of submarines and ships, Google, cisco, Facebook, twitter, linkedIn, server hosting the website of the white house (Drupal), federal government of Brazil favours Linux operating systems over all others

20

The Android operating system is primarily based on Linux kernel and Google has made several changes to make it go above and beyond the original basis of Linux kernel.

21

Linux has a strong following in smart phones - palm's webos, Google's android and nokia's maemo smart phone operating systems are built on top of the Linux kernel.

22

Last year, 75% of Linux code was developed by programmers working for corporations. Google has contributed about 1.1% of the code in the current Linux kernel.

23

As Torvald began its project, the first Linux kernel occupied only 65 KB. Today, even a regular Linux kernel contains over 10 million lines of source codes.

24

As of January 2010, Linux still only has a 1.02% market share within desktops.

25

You can run Microsoft windows and the Linux kernel side by side on the same computer using a software called Cooperative Linux (coLinux).



GitHub

<CODE>

<COLLABORATE>

<SHIP>

GitHub is a website and service that we hear geeks rave about all the time. Want to know what all the GitHub hubbub is about? Read on to find out.

The “Git” in GitHub

To understand GitHub, you must first have an understanding of Git. Git is an open-source version control system that was started by Linus Torvalds – the same person who created Linux.

Git is similar to other version control systems – Subversion, CVS, and Mercurial to name a few.

> Version control systems

So, Git is a “version control system,” what’s that mean? When developers are creating something (an application, for example), they are making constant changes to the code and releasing new versions, up to and after the first official (non-beta) release.

Version control systems keep these revisions straight, and store the modifications in a central repository. This allows developers to easily collaborate, as they can download a new version of the software, make changes, and upload the newest revision. Every developer can see these new changes, download them, and contribute.

Similarly, people who have nothing to do with the development of a project can still download the files and use them. Most Linux users should be familiar with this process, as using Git, Subversion, or some other similar method is pretty common for downloading needed files, espe-

cially in preparation for compiling a program from source code (a rather common practice for Linux geeks).

In case you are wondering why Git is the preferred version control system of most developers, it has multiple advantages over the other systems available, including a more efficient way to store file changes and ensuring file integrity.

> The “Hub” in GitHub

We’ve established that Git is a version control system, similar but better than the many alternatives available. So, what makes GitHub so special? Git is a command-line tool, but the center around which all things involving Git revolve – effectively, the Hub, is GitHub.com, where developers can store their projects and network with likeminded people.

Let’s go over a few of the main reasons that geeks like to use GitHub, and learn some terminology along the way.

> Repository

A repository is a location where all the files for a particular project are stored, usually abbreviated to “repo.” Each project will have its own repo, and can be accessed by a unique URL.

> Forking a repo

“Forking” is when you create a new project based off of another project that already exists. This is an amazing feature that vastly encourages the further development of programs and other projects. If you find a project on GitHub that

you’d like to contribute to, you can fork the repo, make the changes you’d like, and release the revised project as a new repo. If the original repository that you forked to create your new project gets updated, you can easily add those updates to your current fork.

> Pull requests

You fork a repository, make a great revision to the project, and want it to be recognized by the original developers, maybe even included in the official project/repository. You can do so by creating a pull request, so the authors of the original repository can see your work, and then choose whether or not to accept it into the official project. Whenever you issue a pull request, GitHub provides a perfect medium for you and the project’s maintainer to communicate.

> Social networking

The social networking aspect of GitHub is probably its most powerful feature. Each user on GitHub has their own profile, showing your past work and contributions to other projects via pull requests.

Project revisions are able to be discussed publicly, so a mass of experts can contribute knowledge and collaborate to advance a project forward. Before the advent of GitHub, developers interested in contributing to a project would usually need to find some means of contacting the authors, probably by email, and then have to convince them that their contribution is legit and they can be trusted.

Life - A Program

"Life is like a program code, and people around you are variables in it. You yourself are the main function of the program... Your parents and family members are 'global variables' and are 'constant'.

How you achieve something or mess up something is your 'methods'. Relations are like the 'functions' that work on variables.

Emotions are 'operators'. And if you mess up with the operators, u usually end up having 'errors'!

Similarly you make mistakes applying wrong instincts. But what is interesting is that, your life is a program code that never ends, u always have to travel through the same code, and if you wish to make a stronger bond then 'goto' the block. If you want to leave behind the past, you can always 'continue' and want to end something,'break' it.

Like the program code to be executed successfully, Life also is, you must always check what's happening and what matters, check priorities, check initials and bind upon the core values and get what is expected... And you have to keep safe all 'variables' because you don't know when you might need them, like wise you also should understand people, you can't loose them because you don't know what happens in the future and when you might need them.... After all, life is a program code with goto, continue steps and it obviously is an infinite loop...."

K. Sumanth
3rd Year, B.Tech

CSE TIMELINE







- Dennis Ritchie

[1941 - 2011]

UNIX is basically a simple operating system,
but you have to be a genius to understand the simplicity.

- Dennis Ritchie