



# HITech

*Technical magazine of CSE Department,  
Heritage Institute of Technology*

---

## **In This Issue:**

### **1. Foreword**

By, Dr. Poulami Das, Associate Professor and Departmental coordinator, CSE

*Page 2*

### **2. Octo's Guide to the Git-Verse**

By, Shuvam Manna, Ex-Student, CSE

*Page 3*

### **3. Development of an IoT-Based Indoor Air Quality Monitoring Platform**

By, Vivek Vatsa, 3<sup>rd</sup> Year, CSE

*Page 6*

### **4. My experience on hackathon**

By, Shaima Haque, 3<sup>rd</sup> Year, CSE

*Page 7*

### **5. Troubleshooting USB Flash Drive Detection in the Linux Kernel**

By Divya Goswami, Ex-Student, CSE

*Page 9*

### **6. Layout Optimization, Thermal Issues, and Testing of 3D Integrated Circuits - Thesis Summary**

By, Dr. Sabyasachi Banerjee, Assistant Professor, CSE

*Page 12*

### **7. Hacking through Habits**

By, Debanjan Ganguly, Ex-Student, CSE

*Page 13*

### **8. Crossword**

By, Dr. Anindita Kundu, Assistant Professor, CSE

*Page 14*

---

# Foreword

**By Dr. Poulami Das, Associate Professor & Departmental Coordinator, CSE**

“Computer science empowers students to create the world of tomorrow.” - Satya Nadella, CEO of Microsoft.

True to Mr. Nadella’s words in today’s world computers and smart phones have been playing a larger role in our daily lives. As mobile devices flourished at ever decreasing real cost, the everyday life of the citizen was not far behind in being transformed by the connectivity of the Internet and the Web with access to family and friends on social media coming to dominate everyday interactions. Upcoming generations can scarcely imagine life without the connectivity of this portable digital device. With developments in AI, cloud computing, data science, deep learning, cybersecurity and IoT paving the way to a new era of integrated cyber world. The course curriculum is the basic steps of the Computer Science knowledge of a student. To nurture this knowledge hands on project, discussions are as important as the curriculum taught in the educational institutes. Regular teacher-student interaction can increase the quality of projects, research work etc.

Computer Science and Engineering department of Heritage Institute of Technology has always been able to maintain a cordial and excellent relationship between the students and the faculty members. The department has indoctrinated a spirit of entrepreneurship and hone their professional skills through developmental activities and interaction with industry, The faculty members of the department always encourage students to learn new skills, to publish research papers and enable them to analyze and solve the problems in their fields of interest.

In the last few months, the department of Computer Science and Engineering has organized a few distinguished events. One of our prestigious alumni Dr. SatyakiSikdar is going to deliver a lecture on "Principled Structure Discovery in Graph Data".

Satyaki received his Ph.D. degree from the Department of Computer Science & Engineering at the University of Notre Dame in 2021. He is a Postdoctoral Fellow at the Luddy School of Informatics, Computing, and Engineering at Indiana University. One of another brilliant alumni Mr. Chandrasekhar Mukherjee is delivered a lecture on "Compressibility: Power of PCA in Clustering Problems Beyond Dimensionality Reduction". Our very own renowned and emeritus Professor Amitava Bagchi sir delivered a series of amazing lectures on various topics like analysis of the control structure of a workflow, Game tree etc. An amazing and exciting online session on Cross-Platform Mobile app development was organized by RAIT-ACM. The students participated in that event with great excitement.

In this edition there are a various interesting and cutting-edge related articles which I am sure will generate a lot of curiosity among the readers of this magazine. From IoT to VLSI technologies, experience in Hackathon of our students – this edition of our magazine will create interest a lot more diverge readers.

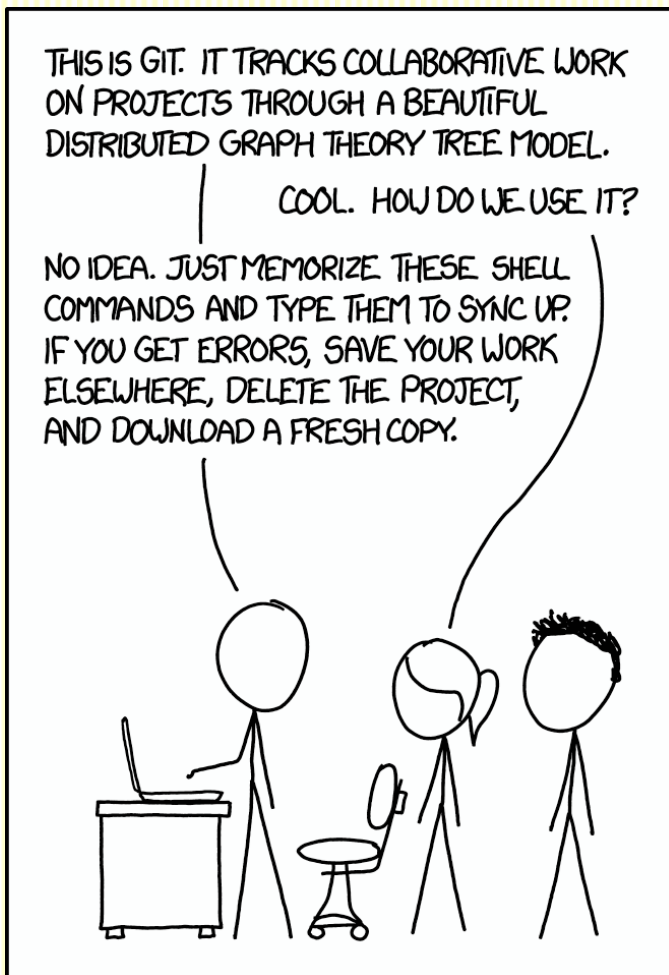
At the end I would like to thank all the faculty members and students who have enriched this magazine by contributing their article and spent a lot of their valuable time for this magazine.

Wish you all an enjoyable reading.

---

# Octo's Guide to the Git-Verse

By, Shuvam Manna, Ex-student CSE  
Graduated in 2021



Earth to Earth-22. I hope this is a time when things have gone ahead, and we have left the umpteen crises behind. This is the bajillionth article on git floating across the multiverse, and why should we consider this to be any different.

What is git? A life form in a remote planet asks. The human tears up. The excerpts of the conversation that ensued shall be duly paraphrased

and translated — because honestly, live translation is an extremely resource-intensive process and we would not want to burden you with the same. If the adjacent xkcd comic is any indication, if there are two words to describe git, one of them is definitely — confusing. The second one might lift you up a little, maybe a lot, when we say that one of git's many superpowers is that it is absolutely amazing. Now that brings us neatly back to one question — What is git? And that also ties in with a second question of why we need it in the first place when you also have things like Subversion (duh!) and Mercurial (who is in no way related to Mercury, the Roman brother of Hermes — the Greek God, not the brand)

Git is a version control system. The idea behind git is that you can track changes in your project folder and store snapshots of how the project had looked at certain points in time. It tries to unclutter the ubiquitous problem of creating umpteen folders to maintain snapshots of your code by creating a virtual Checkpoint mechanic you find in games. It is especially helpful when we are collaborating with others, or finding instances of when certain changes were made and reverting them (if they break your existing infrastructure), if necessary. And it is also useful as a documentation tool if used properly with the right set of comments and commit messages. The most powerful aspect of git, and where it ascertains its supremacy over virtually any other version control system is that it was built to be distributed. And because it is distributed, there are certain weird things and quirks that don't always make perfect sense.

With other version control systems (referred henceforth as vcs) like Subversion or Mercurial, you would notice that they are sequential — your version numbers follow a neat and simple 1,2,3,4,5... In git, you have this uber long commit ID that is a 160-bit long hash string that identifies versions or commits history checkpoints.

Now, while this was done to facilitate the unique naming of commits in a decentralized system, the question naturally pops up is: How likely is it that the hash generated for a future commit will coincide with the hash of some past commit?

The question is closely related to something we normally refer to as the Birthday paradox, from

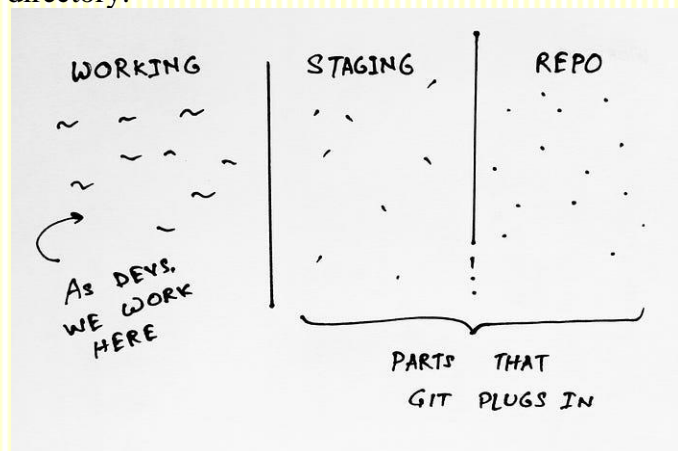
where we see that if we were to randomly select  $n$  from a set of  $N$  distinct elements, the probability of drawing the same element  $n$  more than once will be greater than half if  $n \geq 1.2\sqrt{N}$ .

Every time a commit is added to a git repository, a hash string that identifies this commit is generated. This hash is computed with the SHA-1 algorithm and is 160 bits long. Expressed in hexadecimal notation, such hashes are 40 character strings.

To go a bit further into the math of the probability of a hash collision (without using scary figures) — At a high rate where every human in the world (say, 7 billion) makes a commit every second, mankind would need nothing less than 6.66 million years to produce a number of commits large enough to create a hash collision with 50% probability!

The life cycle of a project with git version control initialized starts with the `git init` command, which initializes the entire set of operations that are to happen on the repository. What `git init` does is it initializes a directory called `.git` inside your project folder. Now, for other vcs, it mostly operates on a client-server model where you check-in your code synchronously by coordinating with, often manually, with other clients connected to the same centralized server.

In git, the `.git` directory is essentially the folder with some metadata where all operations are performed. Since it is decentralized, there is no network communication. All operations are local and file CRUD operations happening inside the `.git` directory.



The workspace:

In the git-verse, as we initialize the git tracker in a folder, the underlying mechanism slaps on these two additional zones on top of your working layer.

This brings us to another command — `git add` that takes your files and moves them to the Staging area as a rough draft that you can later publish to your repository history. In this context, we can imagine staging being this fantastical inter-dimensional layer that holds your data.

You generally use the command as `git add .` or `git add *` that adds all the relevant unstaged files in your working area to staging. You can also do a `git add <filename>` to include specific files into this layer.

To undo this step, i.e. to get back data from staging into the working layer, we use the commands `git reset` for a generic removal of everything from staging and `git reset <filename>` if we just want to do that for a specific file.

We use another command called `git commit` to move data from the staging layer to the Repository layer. This command takes the rough snapshot of your work that you have in your staging, where you add and remove files and changes, and saves that snapshot forever in the repository later.

Let's say we take a file and run `git add`, you should be ideally able to track changes as they happen in the `.git` folder, as they are just file operations. There is no actual file movement involved. The system takes the file, observes its contents and forms a BLOB, takes some header information like how long the file is, etc. and passes it on to the SHA-1 algorithms that give us our 40-character hash, and stores it in the objects subfolder under `.git`.

Now when we run the `git commit -m "Some commit message"` in terms of the git system, it creates another structure in the objects subfolder called a tree. The purpose of a commit is to create a snapshot of your project at that point in time, with the message mentioned in quotes — The tree represents what your working directory looked like at that point in time.

To see the components of a tree snapshot at any point in time, use the command `git ls-tree <first 5 characters from commit ID>`.

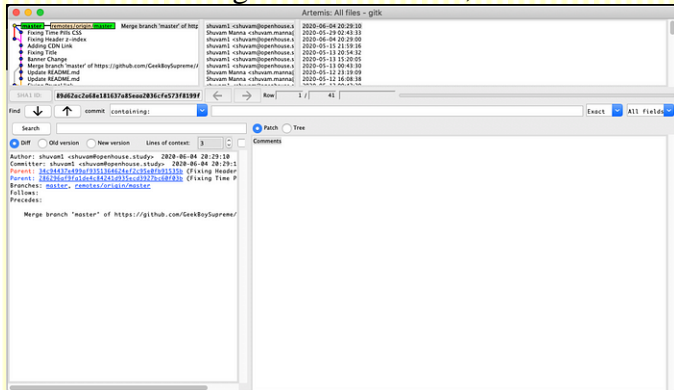
The tree essentially contains a reference of UNIX or OS (depends on your Operating System) permission code (say 644 or 755), a numeric reference to your type of file, a reference to your BLOB, the commit ID, and the actual file that was added/changed.

Now the interesting bit is, what happens when you rename a file?

On the terminal, you see an old file is deleted and a new file is created. But under the hood, you essentially create a new tree that has a pointer referencing the same old BLOB because the contents remain the exact same. This is the mechanism employed as a redundancy handler and takes up lesser memory.

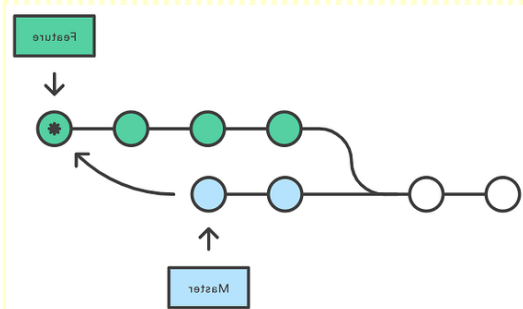
Another cool command/tool that we would explore before moving on to the repository layer is a command called `gitk`.

This is a free GUI tool that gets installed when you install git itself and helps you visualize the commit, the commit message and commit ID, etc.



The Git GUI tracker launched with `— gitk`

The last thing that we would touch base on in this post, is the idea behind branches. The basic idea that pops into our head when we talk about branches is that there is a divergence — that the trackers go sideways, and a whole new route emerges and something of that sort, and that idea is probably reinforced when we mention that git uses trees — but that’s the wrong mental model to have. A branch is essentially a pointer to the elements that the tracker is looking into. For instance, the default branch tracker when you initialize a git repository points to your first commit. Once you do a second commit, the branch pointer points to your second commit, which in turn points to your first commit and so forth.

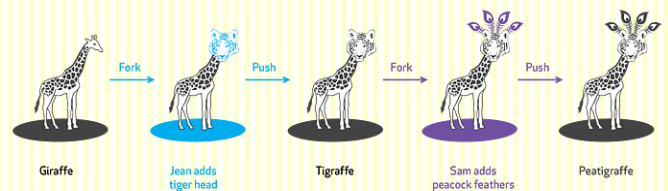


Therefore, in a broad sense of abstraction, branches are essentially pointers to a specific snapshot/commit in your git repository. They are not divergent literally like branches, and these branches are in no way related to the tree objects that we mentioned earlier.

This is the inner functioning of git in an extremely abstracted manner — an insight as to how git works under the hood. Obviously, this does not cover all aspects, there are far finer nuances to this amazing version tracking system, for instance, the intelligence layer that groups similar BLOBs to optimize memory usage, or the redundancy checks that take place to ensure that the `.git` folder does not exceed the repository source code itself in size (That would be hilarious though).

There are some pretty nifty sources to learn more about git around the Web.

Until next time, keep making Peatigraffes!



peacock feather, tiger giraffe from The NounProject

Image Courtesy: GitHub Blog

# Development of an IoT-Based Indoor Air Quality Monitoring Platform

By Vivek Vatsa, 3<sup>rd</sup> Year, CSE

According to the World Health Organization (WHO), 90 percent of the population now breathes polluted air, and air pollution is the cause of death for 7 million people every year.

According to the National Green Tribunal(NGT), indoor air is 100 times more contaminated than outside air. Most modern populations spend 80 -90 % of their time indoors; thus causing a direct impact on their health than outside air.

Indoor air quality management is very important, as it can prevent exposure through proactive precautionary measures. Therefore, efficient and effective monitoring of indoor air is necessary to properly manage air quality.

The technology used in Smart Air Systems:

The introduction of technologies such as the Internet of Things (IoT) and cloud computing has revealed new capabilities of real-time monitoring, especially in Smart-Air.

The Smart-Air device consists of a laser dust sensor, a volatile organic compound (VOC) sensor, a carbon monoxide (CO) sensor, a carbon dioxide (CO<sub>2</sub>) sensor, and a temperature-humidity sensor.

Moreover, an LED strip was installed in the center of the device to visualize air quality using colors. When the quality of air changes, the device's LED changes color and wirelessly sends an alert message to the web server via LTE. Thus, the LTE modem transmits and receives data by communicating with the web server for detailed monitoring and determination of air quality as the presentation layer of the platform.

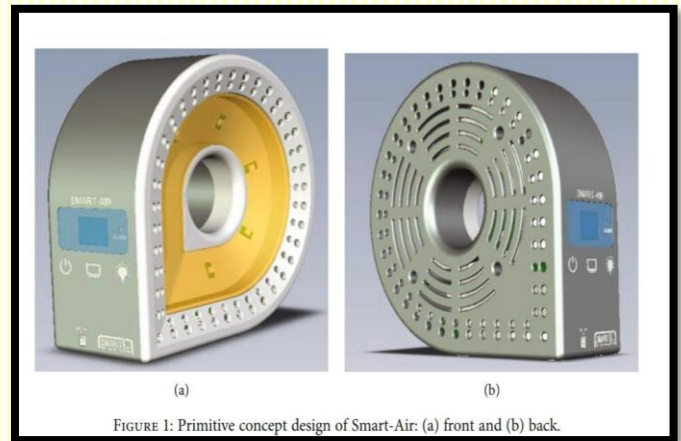


FIGURE 1: Primitive concept design of Smart-Air: (a) front and (b) back.

The sensors are built with an ARM 32-bit cortex instruction set, with a 1024 KB SRAM and 168 MHz clock frequency operating at 1.8-3.6 V under surrounding temperatures of -40 - 125 °C. They can detect particulate impurities at PM 2.5-10, any ethyl vapor content, and humidity accuracy up to 78%.

Further working and connections are shown as:

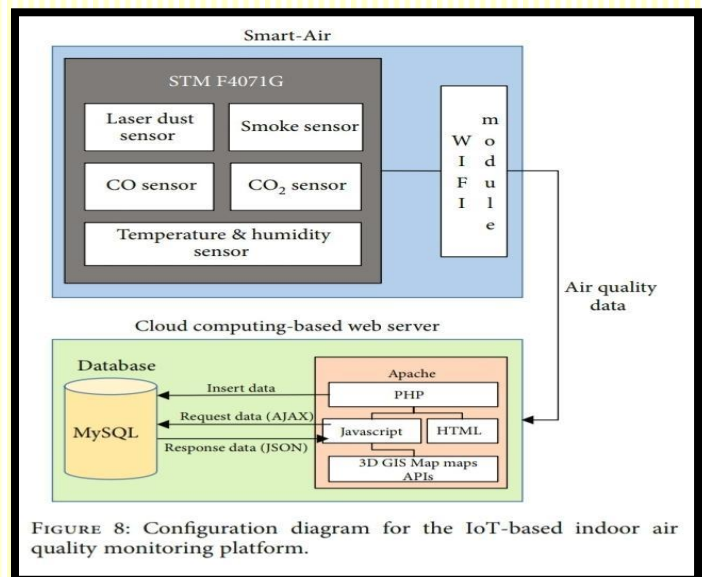


FIGURE 8: Configuration diagram for the IoT-based indoor air quality monitoring platform.

Thus, we get a real time analysis of the collected data which ensure data accuracy forlong time periods. In addition, ventilation system can be connected to the platform.

# My experience on hackathon

By, Shaima Haque, 3<sup>rd</sup> year, CSE

Hackathons are amazing networking and learning opportunities for anyone who loves to build side projects, meet people and learn some new technologies. They're usually time-restricted events ranging from about 36 hours to 48 hours to even 2 months.

My hackathon experience started in the first year itself, which was virtual because of the pandemic. How did I get to know about them? I was active on Twitter and LinkedIn and had the right connections which kept me informed.

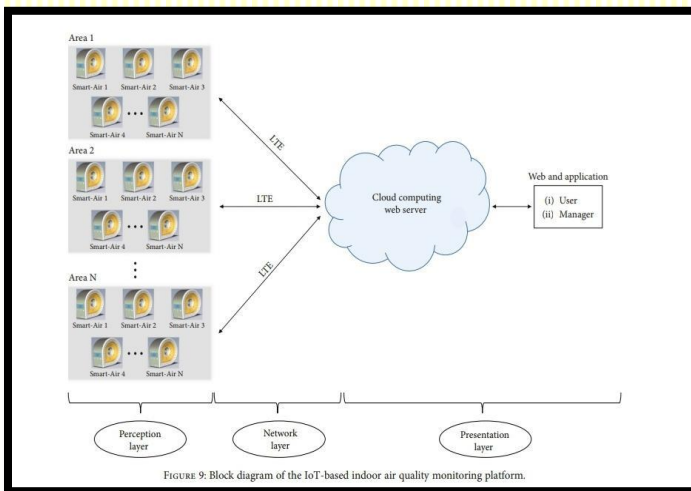
I attended Major League Hacking's Local Hack Day: Share (which has now been rebranded to Global Hack Week), which is not a hackathon in a traditional sense, but it introduced me to a wide range of domains. Every day new tasks were added to the website and for every task completed, you get a point. This goes on for a week, so, for example, in a day I was learning how to make a chrome extension and create a Simon Game, trying out a new API, attending workshops, and meeting new people on Discord.

I was exposed to a lot of technologies at once, which later helped me choose which domain I wanted to explore.

Soon after this, I attended Women/Hacks 3.0, an international hackathon by UCSB, under Major League Hacking. At that time, my skill sets included HTML/CSS, Basic JavaScript, C++, and Java. The only useful skills that I knew were HTML/CSS, from a hackathon project perspective. The best part about attending hackathons during the pandemic was the exposure to a diverse group of people from different time zones.

I introduced myself in the Discord channel of the hackathon. After this, I started interacting with other participants, and eventually, we became a team of 4.

Two of us were from India and the other two were from the USA, and only one of us was experienced. The advantage of this was, at any point in time there were a group of people working on the



Thus, the system can be automatically operated to improve the air quality whenever the air quality is not good. Hindawi Journal of Sensors, Vol. XX may be referred; one may refer Article 15.4, Developing Air Sensors, PRAGYATMIKA too.

project, hence utilizing the time allotted for the hackathon fully.

I learned valuable lessons during that hackathon.

- Knowing your limits is very important. Since we were 3 novices deciding on a full-stack project would be detrimental in a 48-hour hackathon.
- Brainstorming very early on in the hackathon and being in sync. Knowing the project inside out is very important. You need not understand the code base completely, but being clear about the work to be done is very important.
- Being ready to learn. Our team created a posture analyzer to combat bad posture due to being on Zoom all day long during the pandemic. We all experience Zoom fatigue and want to reduce the strain on our bodies by improving our and our peers' postures.

We needed data, a decent front end to run the webcam, and the TensorFlow model.

We curated the data ourselves and figured out a way to add the model to our front end.

We ended up winning the hackathon in three categories, not because of the size of the project but because of the clean execution of our initial idea.

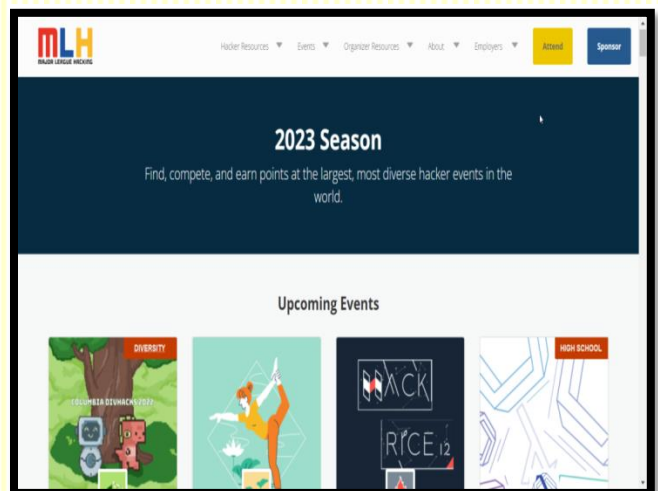
After this, I went on to attend more virtual hackathons and teaming up with people. There are always things to learn and get exposed to during a hackathon. And, if you make an effort your network stays with you.

I also got the chance to represent the Heritage Institute of Technology at the Smart India Hackathon 2022 at DY Patil College of Engineering, Pune along with Agnish Ghosh, Bikram Saha, Rajarshi Paul, and Subhranil Raha.

For the people who might not be able to travel too much to hackathons, hybrid hackathons are being held in most places, and you can participate in those. There are company hackathons that you can attend, which will assure you of a job interview later.

Places where you can find hackathons:

## - Major League Hacking



## - Angel Hacks





---

# Troubleshooting USB Flash Drive Detection in the Linux Kernel

By Divya Goswami, Ex-student CSE, Graduated in 2022

## Why you should boot a self compiled kernel at least once in your life.

### Problem Statement:

Mounting usb drive on an ubuntu machine. Easy? Well I thought so until 3yrs back:

```
divya@rachejazzblog:~$ sudo mount /dev/sdb /mnt/usb/
[sudo] password for divya:
mount: /mnt/usb: wrong fs type, bad option, bad superblock on /dev/sdb, missing codepage or helper program, or other error.
```

### UGH!!!! WHY??? Solution?

- 1.Format disk- WHAT? NO! I have sensitive content.
  - 2.Change OS - eh?
  - 3.Troubleshoot yourself - Uhh...I am not too comfy with drivers and kernel.
- Me: Let's begin with 3.

### First up. What caused that error.

Inject usb, let's get started. Now, the kernel gods check the heart beat of the kernel through `dmesg`

```
DMESG(1) User
Commands DMESG(1)

NAME
dmesg - print or control the kernel ring buffer
Starting from that: Latest entries show:
[ 42.547467] usb 1-1: new high-speed USB device number 2 using ehci-pci
[ 44.351449] usb 1-1: New USB device found, idVendor=0930, idProduct=6544, bcdDevice= 1.00
[ 44.351455] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[ 44.351459] usb 1-1: Product: DataTraveler 120
```

```
[ 44.351463] usb 1-1: Manufacturer: Kingston
[ 44.351466] usb 1-1: SerialNumber: 001E4F923A63C910A5430647
[ 44.860402] usb-storage 1-1:1.0: USB Mass Storage device detected
[ 44.861761] scsi host33: usb-storage 1-1:1.0
[ 44.862160] usbcore: registered new interface driver usb-storage
[ 44.918173] usbcore: registered new interface driver uas
[ 45.879654] scsi 33:0:0:0: Direct-Access Kingston DataTraveler 120 1.00 PQ: 0 ANSI: 2
[ 45.881809] sd 33:0:0:0: Attached scsi generic sg3 type 0
[ 45.884672] sd 33:0:0:0: [sdb] 7823296 512-byte logical blocks: (4.01 GB/3.73 GiB)
[ 45.887095] sd 33:0:0:0: [sdb] Write Protect is off
[ 45.887104] sd 33:0:0:0: [sdb] Mode Sense: 65 44 09 30
[ 45.889032] sd 33:0:0:0: [sdb] No Caching mode page found
[ 45.889103] sd 33:0:0:0: [sdb] Assuming drive cache: write through
[ 45.937467] sd 33:0:0:0: [sdb] Attached SCSI removable disk
[ 598.527900] e1000: ens33 NIC Link is Down
[ 602.552714] e1000: ens33 NIC Link is Up 1000 Mbps Full Duplex, Flow Control: None
```

So my kernel does reconginze the thumb drive. Let's filter out the "error" part

```
divya@rachejazzblog:~$ dmesg --level=err
[ 2.208683] piix4_smbus 0000:00:07.3: SMBus Host Controller not enabled!
[ 2.798407] sd 32:0:0:0: [sda] Assuming drive cache: write through
[ 45.889032] sd 33:0:0:0: [sdb] No Caching mode page found
[ 45.889103] sd 33:0:0:0: [sdb] Assuming drive cache: write through
```

We are concerned with the `sdbpart`(that's how your kernel allocates "nicknames" to call your devices) "No Caching mode page found" So we know it is an error in the kernel ring. sigh. Forget it. Let's change OS. I was kidding. I changed the kernel instead.

---

## Compiling and building kernel on your own

Now I will be using linux-next tree but you can try the stabler main tree as well.

1. Git clone
2. copy config file
3. make
4. install
5. reboot check uname -r Ah! now we can start messing up. Afterall the kernel is ours :D

### Test the problem. Hopefully it still exists.

Same config file. Let's start digging

### Is the usb listed under lsusb?

```
modprobe: FATAL: Module usb_storage not
found in directory /lib/modules/5.13.0-
rc4-next-20210603
[divya@rachejazzblog~]$lsusb
Bus 001 Device 002: ID 058f:6387 Alcor
Micro Corp. Flash Drive
Bus 001 Device 001: ID 1d6b:0002 Linux
Foundation 2.0 root hub
Bus 002 Device 003: ID 0e0f:0002 VMware,
Inc. Virtual USB Hub
Bus 002 Device 002: ID 0e0f:0003 VMware,
Inc. Virtual Mouse
Bus 002 Device 001: ID 1d6b:0001 Linux
Foundation 1.1 root hub
```

Yes, it does.

### Where else can you see the usb plugged in?

```
[divya@rachejazzblog~]$lsblk -f
NAME                FSTYPE      LABEL
UUID
FSAVAIL FSUSE% MOUNTPOINT
fd0
sda
├─sda1
└─sda2                ext4
e633212a-0588-492b-a5ce-7803b4d8348a
47.3G    25% /
sr0                iso9660      CDROM
2021-05-24-00-20-50-00
srl1                iso9660      Ubuntu-Server 20.04.2 LTS
amd64 2021-02-01-17-57-41-00
```

It does not list under SCSI block devices Recheck:

```
[divya@rachejazzblog~]$sudofdisk -l
```

```
[divya@rachejazzblog~]$sudofdisk -l
[sudo] password for divya:
Disk /dev/fd0: 1.42 MiB, 1474560 bytes,
2880 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
Disklabel type: dos
Disk identifier: 0x90909090
```

Device	Boot	Start	End
Sectors	Size	Id	Type
/dev/fd0p1		2425393296	4850786591
2425393296	1.1T	90	unknown
/dev/fd0p2		2425393296	4850786591
2425393296	1.1T	90	unknown
/dev/fd0p3		2425393296	4850786591
2425393296	1.1T	90	unknown
/dev/fd0p4		2425393296	4850786591
2425393296	1.1T	90	unknown

```
Disk /dev/sda: 70 GiB, 75161927680 bytes,
146800640 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes
/ 512 bytes
I/O size (minimum/optimal): 512 bytes /
512 bytes
Disklabel type: gpt
Disk identifier: D7F8DF7E-9C6C-4314-BFCA-
5B463C68318E
```

Device	Start	End	Sectors	Size
Type				
/dev/sda1	2048	4095	2048	1M
BIOS boot				
/dev/sda2	4096	144707583	144703488	69G
Linux filesystem				

### Subproblem no. 1 - device does not show up as block device.

It can be possible that in your newly installed kernel, there can be some modules which are not loaded. In this case, we need to load module `usb_storage` manually which is responsible for USB Mass Storage on Linux.

1. Go to the newly cloned linux tree directory.
2. Open the config file in a text editor
3. Search for "STORAGE" 8-)

So now we know why usb did not show up as a storage block device. Because it was not told to the kernel to setup! Change it to:

```
CONFIG_USB_STORAGE=y
```

Now, we build the kernel again against the new config Do a make -j3 and relax for awhile NOTE You will be asked a couple of questions about each usb brand config. Hitting enter for the time being won't be harmful. Something like this: (show kernel compile image) Install the new kernel sudo make modules\_install install reboot into new kernel verify using uname -r(will be the same with a -dirty tag)

```
[divya@rachejazzblog~]$modprobeusb_storag
e
[divya@rachejazzblog:~]$ lsmod | grep usb
usb_storage77824  luas
usbhid57344 0
hid 131072
2usbhid,hid_generic
```

Check whether usb is visible on lsblk and fdisk now or not:

```
[divya@rachejazzblog~]$sudofdisk -l
[sudo] password for divya:
Disk /dev/fd0: 1.42 MiB, 1474560 bytes,
2880 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
Disklabel type: dos
Disk identifier: 0x90909090
```

Device	Boot	Start	End
/dev/fd0p1		2425393296	4850786591
2425393296	1.1T 90	unknown	
/dev/fd0p2		2425393296	4850786591
2425393296	1.1T 90	unknown	
/dev/fd0p3		2425393296	4850786591
2425393296	1.1T 90	unknown	
/dev/fd0p4		2425393296	4850786591
2425393296	1.1T 90	unknown	

```
Disk /dev/sda: 70 GiB, 75161927680 bytes,
146800640 sectors
Disk model: VMware Virtual S
Units: sectors of 1 * 512 = 512 bytes
```

```
Sector size (logical/physical): 512 bytes
/ 512 bytes
I/O size (minimum/optimal): 512 bytes /
512 bytes
Disklabel type: gpt
Disk identifier: D7F8DF7E-9C6C-4314-BFCA-
5B463C68318E
```

Device	Start	End	Sectors	Size
/dev/sda1	2048	4095	2048	1M
BIOS boot				
/dev/sda2	4096	144707583	144703488	69G
Linux filesystem				

```
Disk /dev/sdb: 15 GiB, 16106127360 bytes,
31457280 sectors
Disk model: Flash Disk
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes
/ 512 bytes
I/O size (minimum/optimal): 512 bytes /
512 bytes
```

And

```
[divya@rachejazzblog~]$lsblk
NAME          MAJ:MIN  RM  SIZE  RO  TYPE
MOUNTPOINT
fd0           2:0      1   1.4M  0  disk
sda           8:0      0   70G   0  disk
├─sda1        8:1      0    1M   0  part
└─sda2        8:2      0   69G   0  part /
sdb           8:16     1   15G   0  disk
sr0          11:0     1  93.2M  0  rom
sr1          11:1     1   1.1G   0  rom
```

Phew! Flash drive is atleast visible now!

So, this is a small instance where a problem can be solved looking at the config of your current kernel. Interested in what else can there be? Hop in!

---

# Layout Optimization, Thermal Issues, and Testing of 3D Integrated Circuits

By, **Dr. Sabyasachi Banerjee, Assistant Professor, CSE**

This article is a summary of my PhD. thesis titled "Layout Optimization, Thermal Issues, and Testing of 3D Integrated Circuits". Hope the article interests students to take up this line of research.

With rapid shrinking of device dimensions, as Moore's law starts to fail, VLSI circuits are now being integrated using 3D-architecture. Interconnections across different layers of a 3D chip mandate the use of vertical vias that can be "Through-Silicon Vias" (TSV) or "Monolithic Inter-tier Vias" (MIV). As a consequence, optimizing the usage of TSVs and MIVs, thermal management, modular placement and parallel testing of cores that spread over the different layers of the 3D-ICs, generally have emerged as challenging problems to solve. Furthermore, 3D chips have significantly larger power densities compared to their 2D counterparts because of their compact nature and also due to the dense integration of a large number of components mostly operating with higher frequency and bandwidth. Thus, new algorithms for design and testing of 3D chips are required, which not only need to be efficient from the computation viewpoint but also have to be suitable for executing multi-objective optimization, such as TSV minimization, thermal-aware placement, wire length minimization and power-aware parallel testing.

In this thesis, we have identified a few related problems and proposed solutions supported by theoretical formulation and experimental validation. The first part of the thesis deals with design strategies, where we discuss the management of TSVs with or without thermal issues. TSVs or vertical vias introduce substantial delay and hence their count induces a constraint for 3D-ICs, due to their high cost, higher area requirement and increased IR drop. These vertical vias cannot be aligned in a perfect vertical position between the

pair of locations in the two consecutive layers that they are connecting. As a result, routing from the IO pads of TSVs to micro-bumps in the re-distribution layer (RDL) plays an important role in transmitting a signals between two adjacent layers of a 3D IC. We show that such alignment problems also can be dealt with intelligent placement of modules in each layer, so that these vertical vias (TSVs) can be aligned properly. However, in Monolithic 3D-ICs (M3D-IC), MIVs are used instead of TSVs. MIVs are free from these limitations, so unlike earlier design strategy, no constraint is imposed on the number of vertical vias, if they are made of MIVs.

The second part of the thesis is focused to testing, where we address the problem of reducing the overall test-time for 3D System-on-Chips (3D-SoCs). The test strategy also takes into account the issues of inherent power and thermal constraints, design of test-access mechanism (TAM), and constraints on TAM-width and the number of TSVs. These test aware design issues are mostly targeted for post-bond testing. We present scheduling based approach followed by an efficient design of TAM architecture. As the test-time for SoCs also depends on the wrapper chain and the distribution of scan flops, we perform experiments with balanced wrapper chains considering the usual constraints of TSVs.

Next, we extend our strategies for testing memory cores where the test-time does not depend on the TAM width (number of wires used to transmit test signal). Memory testing is essential for in-field operations as well as to provide pointers to foundries for yield enhancement. A built-in self-repair (BISR) technique is reported in this thesis for memory testing considering thermal and TSV constraints. Experimental results on ITC'02 SoC benchmark circuits are presented.

---

---

# Hacking through Habits

By Debanjan Ganguly, Ex-student CSE, Graduated in 2019

I was never a reader. Being only have read textbooks in my entire life, I struggled to pick anything which was outside the curriculum.

Probably the idea of knowing a ton of things by reading books was more enticing to me than actually reading books, that's why I couldn't stick to reading for no longer than a week for years.



*An organized bookshelf has a different appeal*

With my newly acquired knowledge of the mechanics behind habit formation, I tried a bunch of things

- Allocating the time I was spending on social media to book reading
- Keeping the book I am reading currently on my work table
- Setting up daily alarms to remind myself to read books
- Watching a loads of “Top 5 books you need to read right now” type of videos on YouTube to keep myself motivated

Guess what! All of them failed, I failed.



Reducing time on social media and allocating the extra time to the cause started well. I could see the eventual benefits I was going to experience if the method works. But it didn't take time for me to quickly digress back to my old routine. The thing with habit swapping is, both of the habits have to be similar in nature to make the swap work. Social media provided instant gratification versus delayed gratification from most of the books.

Keeping the book closer to me to have it serve as a cue failed even faster. Having a cue is useless if you don't have a proper reward system in place. And finding a reward that didn't involve sugar going inside my body was hard to find.

The problem was the same for alarms as well, I didn't have anything to back the alarms up. The habit didn't stick.

Watching YouTube videos was motivating enough for me to pick up a book but were not enough for me to complete chapter 2 of any.

Worried about not being able to read any kind of books ever again, and missing out on all the wisdom there is, I decided to document my current habits and tried to figure out why they exist.

A few of them were:

- Watching a TV series, particularly during dinner
- Playing video games after work hours
- Consuming unnecessary amount of black coffee



Playing video games had a very similar reward system to scrolling through any social media, that instant gratification.

---

Drinking coffee is a caffeine addiction, I am a slave and I don't know how to get out of it, but I don't need to. The dopamine release is wonderful and it makes my life beautiful and worth living.

Watching TV series was an interesting habit. I was never a TV person in my childhood days, and watching a sitcom makes me crave for something mentally enticing, but then, without fail, I was able to complete a 10-12 season sitcom within months. On top of it, it is not very different from reading a book, both are consuming activities, both have a delayed gratification process in place, the only difference was TV involves 2 senses (eyes and ears), where reading involves only one (eyes).

Then why the TV stuck, but not the reading?

It was not very long from there to reach my "Ahaa!!" moment. I paid attention to the timing of my watching sessions, most of them were during dinners. Not knowingly, I was implementing what many people in the self-help community call "Habit Stacking".



*I was happy, and I like cats*

Habit stacking is essentially stacking a new habit on top of an existing habit. The new habit shares the cues and the reward system of the existing one, thus eliminating the need for finding new cues and new rewards for new habits. It is very intuitive when you think about it, why create any new resource when you can reuse or recycle old ones?

Let me give you an example.

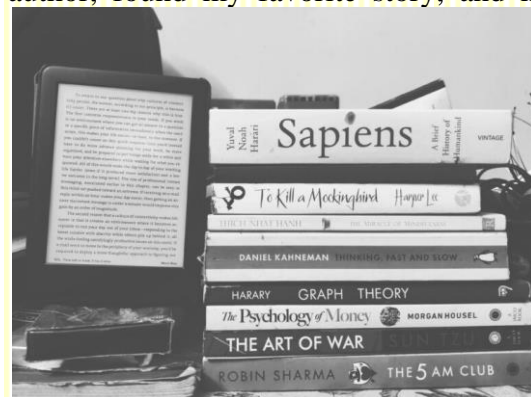
Let's suppose you want to learn a new language. Instead of allocating new timeslots or forcing yourself through motivation, what you can do is keep your learning tools in the place you sleep. As soon as you jump onto your bed (cue) for sleep (which is a current habit), you can pick up the apparatus and have a small learning session (the habit you want to develop) before your good night's sleep (your reward).

I decided to give it a try. At that time my options were to stack it with video games, or with my coffee sessions.

Playing a video game is an involved activity, it leaves you with little to no scope to stack a new habit on top of it. I cannot imagine myself reading a book while shooting bots in their head.

I was left stacking it with my coffee sessions, and it worked. Took less than 3 weeks to complete *Sapiens* by Yuval Noah Harari. It was groundbreaking for me. A rookie with a history of problematic relations with reading, not only was able to pick a book based on historical events but also complete it with little to no problems. I thought I hacked habits.

I never looked back since then, gifted myself a Kindle, read several classics, found my favorite author, found my favorite story, and many more.



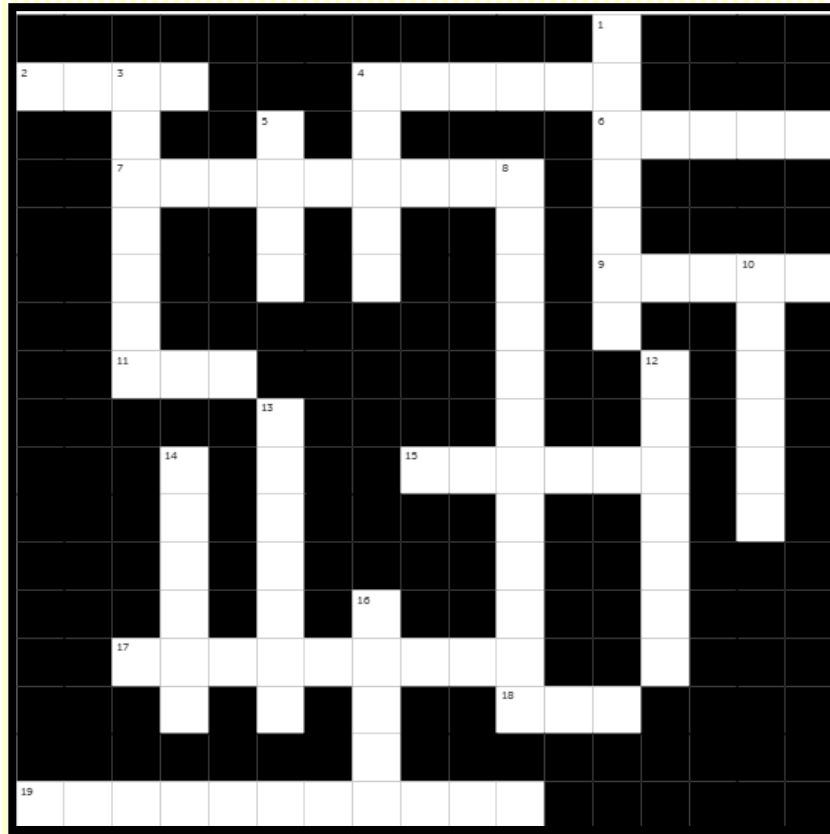
*My books*

I would really suggest trying stacking habits to form new habits, or even pick up old ones if you have. I recently stacked drawing before my sleep, and I am very hopeful it will work out just fine.

---

# Crossword

By, Dr. Anindita Kundu, Assistant Professor,  
CSE



## Across

- 2 A protocol that assigns IP automatically
- 4 A high end machine that serves the user requests
- 6 Physical Layer protocol of 5G networks
- 7 A set of finite rules or instructions to be followed in calculations or other problem-solving operations
- 9 A linear data structure that follows LIFO order
- 11 Transport Layer Protocol
- 15 machine learning model by OpenAI that can generate digital images from natural language descriptions
- 17 A token that marks the end of a statement in C
- 18 A processor required for Gaming and executing machine learning programs
- 19 Meaningful Data

## Down

- 1 A program in execution
- 3 Artificial intelligence (AI)-powered chatbot
- 4 Is an open-source Python library that is used for scientific and mathematical computing
- 5 A famous distance vector routing algorithm
- 8 way of sending multiple signals or streams of information over a communications link at the same time in the form of a single, complex signal.
- 10 A common term associated with network security
- 12 It is a preprocessor directive in C that lets you create macros or constants
- 13 The most common OS used in mobile devices today
- 14 It is a logical subdivision of an IP network.
- 16 A machine resembling a human being and able to replicate certain human movements and functions automatically

---

## Solution

