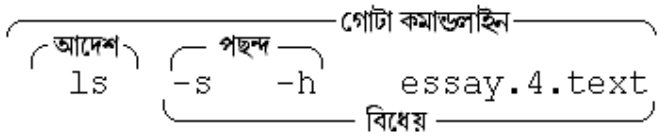
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আগেও বলেছি, আরো একবার মনে করিয়ে দিই, গ্নু-লিনাক্স ইশকুলের এই পাঠমালাটা কোনো টেক্সট-বই নয়। এটা পড়ে আপনি কিছুই শিখবেন না। বড়জোর, কী করে শিখবেন সেটা শেখার চেষ্টা করতে পারেন। কম্পিউটার, তাও আবার গ্নু-লিনাক্সের মত কোনো জ্যান্ত গতিশীল এবং সর্বব্যাপী বিষয় নিয়ে, আমার যোগ্যতার কয়েক কদম বাইরে। আর টেক্সট বই ব্যাপারটার প্রতি আমার আজন্ম অ্যালার্জি, এবং সেটা নিয়েই মরতে যাব বলে বিশ্বাস। আমি নিজে গ্নু-লিনাক্সে উত্তেজিত হয়েছিলাম, এলোপাথাড়ি শেখার চেষ্টা করে চলছিলাম, তখন চিন্তার কিছু কাঠামো কিছু খোঁচ কিছু ভাঙচুর কিছু নির্মাণ নিজের কাছে খুব গুরুত্বপূর্ণ লেগেছিল, যে এলাকাগুলো অন্যদের সঙ্গে শেয়ার করার কথা মনে হয়েছিল। যারাও আমার মত, একইভাবে, একইরকম গোলগোল এবং অশিক্ষিত উপায়ে শেখার চেষ্টা চালাচ্ছে। মনে হয়েছিল, এরকম একটা পাঠমালা তাকে সাহায্য করতে পারে নিজের চিন্তাকে নিজের কাছে গুছিয়ে নিতে, নিজের চেষ্টার সঙ্গে নিজেরই কথোপকথনে। এখানে লেখক আর পাঠকের মধ্যে কমন ওই অপারগতা এবং গোলগোল অশিক্ষাটা। 'অপর' পত্রিকায় মার্জিন অফ মার্জিন নিয়ে লেখার পর থেকেই এই অটেকনিকাল লেখার প্রতি আমার একটা ভালো লাগা তৈরি হয়, কারণ অনেকেরই সেটা কাজে লেগেছিল। সেই অশিক্ষাটাই এই বইটার শর্ত, যেমন গরিবঘরের কষ্টে থাকা মেয়েদের (এলিটঘরের ফেমিনিজম-বেচা সেলসগার্লদের কথা বলছি না) কিছু জায়গা থাকেই যা অন্য একটা মেয়ের পক্ষে বোঝা সহজ। তেমনি আপনার আর আমার একদেশতার সূত্র এটাই — দুজনেরই না-জানা, এবং জানতে-চেষ্টা-করা। এর যেকোনো একটা শর্তকে তুলে নিলেন মানেই লেখাটা হেজে গেল। এটা আপনার উপরেও একটা শর্ত চাপিয়ে দিচ্ছে। এই লেখা থেকে আলাদা করে নিজের কোনো খোঁজা এবং চেষ্টা যদি আপনার না-থাকে, পাঠমালাটা আপনার পড়ার কোনো মানেই নেই। আপনি এটা পড়ে যে কিছুতেই গ্নু-লিনাক্স শিখবেন না, এ গ্যারান্টি আমি আগে থেকেই দিয়ে দিচ্ছি। আজ আমরা শুরু করব কয়েকটা অত্যাবশ্যক কমান্ড দিয়ে, তারপর ঢুকব ফাইলসিস্টেমের প্রাথমিক আলোচনায়।

।। দিন ছয়।।

১।। কয়েকপিস কমান্ড

এবার একদম কেজো একটা জায়গা। পাঁচ নম্বর দিনে লগ-ইন করে আপনি ঢুকে পড়েছেন সিস্টেমে, সেখানে শেল এবং কারনেলের উপস্থিতি সম্পর্কেও একটা আলগা আন্দাজ তৈরি হয়েছে। কিন্তু এবার সিস্টেমটাকে নিজের সামনে চলমান দেখতে গেলে কয়েকটা কমান্ড জেনে নেওয়া খুব জরুরি। আমরা কয়েকটা কমান্ডের উল্লেখ করেছি এর মধ্যেই এক নম্বর দিনে দু-চারটে, এবং পাঁচ নম্বর দিনে গোটাকয়েক। সেগুলোকেই এবার একটা প্রথানিষ্ঠ রকমে ঝালিয়ে নেওয়া যাক, হ্যাঁ দেখবেন, ধূপটা যেন না-নেভে, উঁহু আওয়াজ নয়, এখানে কমান্ডপাঠ হচ্ছে ভাই।



আমাদের অভ্যস্ত গ্নু-লিনাক্স কমান্ডের কাঠামোটা অনেকটা এইরকম — মোট কমান্ড লাইনটার দুটো অংশ, একটা মূল কমান্ড, আদেশটা। অন্যটা হল তার আর্গুমেন্ট, বিধেয়। এই বিধেয়-তে আবার দুটো অংশ, একটা অপশান বা পছন্দ। অন্যটা হল যার উপর আদেশটা চলবে সেই অবজেক্ট বা কর্মটা। ধরুন আপনি পাঁচ নম্বর দিনের ৪ নম্বর সেকশনের আপনার লিখতে থাকা ওই প্রবন্ধের ডিরেক্টরিতে ঢুকেছেন। একটু আত্মশ্লাঘা পেতে চাইছেন নিজের প্রবন্ধরাজির গাবদা গাবদা গতর দেখে। এই মুহূর্তে আপনি যে প্রবন্ধটা লিখছেন তার নাম 'essay.4.text'। আগেরগুলো শেষ হয়ে গেছে। অবভিয়াসলি, আমারও তাই। পরশু প্লাস্টার কাটবে, ঠ্যাং ছুটি শেষ হয়ে এল, আজ নয়ই ডিসেম্বর, এখন প্রায় যুদ্ধকালীন তৎপরতায় লিখতে হচ্ছে। এবার, এই প্রবন্ধটা কতটা লিখে ফেলেছেন এর মধ্যেই, কত বাইট আপনার অবদান, জানতে চাইলেন। কমান্ড দিলেন, '`ls -s -h essay.4.txt`'। এবং এন্টার মারলেন। অহো, কী আনন্দ, বাইট সাইজ দেখছি ধ্যাড়াধ্ধ্যাড় করে বেড়ে যাচ্ছে, অহো, কী আনন্দ, মনে হচ্ছে

এই ধরিত্রীতে একটা পদচিহ্ন না রেখে কিছুতেই মরব না, বিগলিত হয়ে গেলেন ফাইলের সাইজ দেখে। এবার কমান্ডের কাঠামোটা দেখুন।

এই কমান্ডের মধ্যে, 'ls' অংশটা হল আদত কমান্ড বা মূল প্রোগ্রামটা। এবং তার সঙ্গে রয়েছে '-s -h' অংশটা, এটা সেই কমান্ডের অপশান বা পছন্দ। '-s -h essay.4.txt' এই পুরোটা হল কমান্ডের আর্গুমেন্ট বা বিধেয়। এবং 'ls -s -h essay.4.txt' পুরোটা মিলিয়ে হল গোটা কমান্ডলাইনটা। এখানে আপনার হয়ে পছন্দটা আমি করে দিয়েছি, বলে দিয়েছি কী অপশান হবে। এরকম অনেক সম্ভাব্য অপশান হয়, তার মধ্যে থেকে মাত্র দুটো আমি নিয়েছি এখানে। 's' মানে সাইজটা দেখাও, আর 'h' অপশানটা বলে দিচ্ছে ওই সাইজের একক করো বাইট বা কিলোবাইট বা মেগাবাইট এমন কিছু যা মানুষে বোঝে, হিউম্যান রিডেবল হয়, নয়তো ও সাইজটা দেবে ব্লক-এর সংখ্যা দিয়ে। ব্লক ব্যাপারটা আমাদের ভালো করেই বুঝতে হবে, ফাইলসিস্টেমের প্রসঙ্গে, কিন্তু অনেক দেরি আছে। দুটো অপশানেরই আগে, দেখুন, আমরা একটা করে '-' চিহ্ন দিয়েছি। এটা অনেক কমান্ডের সঙ্গে দিতে হয়, অনেক কমান্ডের সঙ্গে দিতে হয়না। অনেক কমান্ডের বেলায় আবার দুটো অপশানকে একসঙ্গে জুড়ে দেওয়া যায়। যেমন সেই কমান্ডটা এই একই কাজ করত যদি কমান্ড দিতাম, 'ls -sh essay.4.txt'। এবার প্রশ্ন হল, আপনি কী কাজ করতে চাইছেন সেটা আপনি জানেন। মানে, এই উদাহরণে, কমান্ডের কর্ম বা অবজেক্ট আপনি জানেন। মানে, 'essay.4.txt' নামের ফাইলটা, যার সাইজ আপনি জানতে চান। এবং ধরুন কী কমান্ড আপনি ব্যবহার করবেন সেটাও জেনে ফেলেছেন (এই জানারও একটা উপায় আছে, পরে সেই কথায় আসছি আমরা)। কিন্তু কোন কমান্ডের অপশান কী হবে, সেই অপশানটাকে কোন আকারে দিতে হবে, সেই হাজারো খুঁটিনাটি আপনি কেমন করে জানবেন? জানবেন এই টেক্সট থেকে।

```
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
-a, --all              do not hide entries starting with .
-h, --human-readable   print size in human readable format (e.g. 1K 3M 2G)
-i, --inode            print index number of each file
-l                     use a long listing format
-R, --recursive        list subdirectories recursively
-s, --size             print size of each file, in blocks
-t                     sort by modification time
-1                     list one file per line
--help                 display this help and exit
```

এটা অনেকটা ত্র্যম্বকের ত্রিনয়ন ত্রিকাল ত্রিগুণের মত হল। এই টেক্সটা গজাল কোত্থেকে? ধীরে। এই টেক্সটের গজানোর ইতিহাসটা এই টেক্সটের মধ্যেই আছে। দেখুন তো আলোচনাটা হওয়ার আগে সেটা নিজেই খুঁজে পান কিনা? পেলেন? পেয়ে যাবেন।

টেক্সটের প্রথম লাইনটা দেখুন, ব্যবহারবিধি বা ইউসেজ, ঠিক অপশান আর্গুমেন্ট কমান্ড-লাইন নিয়ে আমরা এর আগেই যা বললাম সেই কথাটাই অন্য আর এক রকমে বলা। দ্বিতীয় লাইনে বলা কী কাজ করে এই কমান্ডটা। তিন থেকে এগারো নম্বর অব্দি প্রতিটি লাইনের কাঠামোই এক। প্রথমে একটা অপশানের ছোট আকার, তারপর বড় আকার, তারপর সেই অপশানটা কী কাজ করে। ছোট আর বড় দুটোই দেওয়া যায়, ছোটটা দিতে সুবিধে হয়, আর বড়টা মনে রাখতে। শুধু দেখুন ছয়, নয় আর দশ নম্বর লাইনের পছন্দের, মানে '-l', '-t', আর '-1', কোনো বড় আকার নেই, শুধু ছোটটাই আছে। আবার এগারো নম্বর লাইনের পছন্দটার, মানে '--help', শুধু বড় আকারটাই আছে। এর মধ্যে তিন নম্বর লাইনের অপশানটা, মানে '-a' বা '--all', বুঝতে গ্নু-লিনাক্সে কম পরিচিতদের একটু অসুবিধে হতে পারে। একটা ডিরেক্টরির সব ফাইলকে স্বাভাবিক অবস্থায় 'ls' দেখায় না। যেসব ফাইল বা ডিরেক্টরির নামের আগে একটা করে '.', বিন্দু বা ডট আছে, যেমন ধরুন '.kde' বা '.mplayer' ইত্যাদি। সচরাচর এই ফাইলগুলোয় থাকে সিস্টেম সংক্রান্ত কিছু যা দৈনন্দিন কাজে দরকার পড়েনা একজন সাধারণ ব্যবহারকারীর। এই প্রথম অপশানটা, '-a', দিলে, 'ls' সেই আপাতগোপন ফাইলগুলোকেও দেখায়। চার আর আট নম্বর লাইনের অপশানদুটো তো আমরা এইমাত্র ব্যবহার করলাম। পাঁচ নম্বর লাইনের অপশানটা একটা ফাইলের ইনডেক্স নোড বা আইনোড দেখায়, সেটার জাস্ট নামটুকু করেছিলাম আমরা পাঁচ নম্বর দিনে, পরে প্রচুর আসবে। ছ নম্বর লাইনে '-l'

অপশানটা লম্বা লাইনে বড় করে ফাইল সংক্রান্ত অনেকটা তথ্যকে হাজির করার জন্যে। খুবই কাজে লাগে ফাইলের মালিকানা এবং ব্যবহারযোগ্যতা বুঝতে। এই পাঠমালাতেই বহুবার আমাদের ব্যবহার করতে হবে। সাত নম্বর লাইনে, '`-R`' বা '`--recursive`' অপশানটা 'ls' কমান্ডকে জানায়, যে ডিরেক্টরিতে বসে কমান্ড দেওয়া হয়েছে, সেই ডিরেক্টরির মধ্যে যদি কোনো সাবডিরেক্টরি থাকে, তার মধ্যে আরো আরো যত থাকুক, ক্রমনিম্নগামী সেই ডিরেক্টরি গুলোকেও একই ভাবে দেখিয়ে যাও। নয় নম্বর লাইনে '`-t`' বলে দেয় ফাইলগুলোকে অক্ষরের অনুক্রমে নয়, তাদের সময়চিহ্নের নিরিখে সাজাতে। আর দশ নম্বর লাইনে '`-1`' বলে দেয় প্রতিটি ফাইলের জন্যে একটা করে লাইন বরাদ্দ করতে।

শেষ, মানে এগারো নম্বর লাইনের '`--help`' হল সেই অপশান যা দিয়ে আমরা এই টেক্সটা বানিয়েছিলাম। সত্যি কথা বলুন তো, আপনি নিজে খুঁজে পেয়েছিলেন? আমি কমান্ড দিয়েছিলাম, '`ls --help`'। তাতে টেক্সটা স্ক্রিনে এসেছিল। কিন্তু স্ক্রিনে পাওয়া মানে তো এই পাঠমালার পাতায় পাওয়া নয়। ইধার-কা-মাল উধার করার জন্যে কমান্ড দিতে হয়েছিল, '`ls --help > ls.help.text`'। এতে একটা ফাইল তৈরি হয়েছিল, 'ls.help.text'। আর ফাইলে এসে গেল মানেই তো এবার কপি করো আর সাঁটো। এই ভাবে, 'ls' কমান্ডের বানানো স্ক্রিনবার্তাটাকে ফাইলে পাঠানোর কায়দাটাকে আমরা কী নামে ডেকেছিলাম — মনে পড়ছে? যাকগে, এই টেক্সটা কিন্তু সেই ফাইলের গোটাটা নয়, তাকে অসভ্য রকমে ছাঁটকাট করা। ঠিক যতটুকু আমার মনে হয়েছে আমাদের আলোচনার জন্যে এই মুহূর্তে প্রয়োজনীয় সেটুকুই রেখেছি। কতটা ছাঁটা জানেন? ৯৭ লাইন থেকে ১১ লাইন। গোটাটা পড়ার ইচ্ছে হলে, কোনো ব্যাপার না, গ্নু-লিনাক্স সিস্টেম আপনার জন্যে অবারিতদ্বার, বলা মাত্র বানিয়ে দেবে। বলে দেখুন। আপনি এখনো নব্যলিনাক্সী বলে যদি কোনো ডিস্ট্রিবিউশন বানাতে রিফিউজ করে, অচিরে লাগে জানান, এইসব অপমান বরদাস্ত করা উচিত নয়।

আপনাকে বানাতেও হবেনা, বানানোই আছে। এই হেল্প ফাইলটা সেই বানিয়ে রাখা ফাইলের একটা চোকলা মাত্র। ঠিক পাসওয়ার্ড নিয়ে যেটা করেছিলাম পাঁচ নম্বর দিনে — মনে আছে? '`man 5 passwd`' বলে একটা কমান্ড ব্যবহার করেছিলাম। 'man' মানে ম্যানুয়াল। 'ls' কমান্ডটাও সবচেয়ে ভালোভাবে জানার উপায় হল '`man ls`'। শুধু 'ls' কেন, যেকোনো কমান্ডের বেলাতেই তাই। কিন্তু এই 'man' কমান্ডটা ব্যবহার করতে গেলে কমান্ডটাকে তো জানতে হবে। কেন, ম্যানুয়ালের ম্যানুয়াল? '`man man`' কমান্ড দিয়ে আমরা 'man' সম্পর্কে জানতে পারব। আর ওই '`--help`' পছন্দটা তো আছেই। ওটা জুড়ে দিয়ে যে কোনো কমান্ড দেওয়া মানেই তার একটা সংক্ষিপ্ত হেল্প ফাইল যাব, সেটা শোকেস, তার গুদাম হল ম্যান বা ম্যানুয়াল। এই গোটা ম্যানুয়ালের পাতার সমগ্রতাটা ভরা থাকে গ্নু-লিনাক্স সিস্টেমের ভিতরেই। আপনার অঙ্গুলি-পীড়নের অপেক্ষায়। কিবোর্ডে টাইপ করে কমান্ড দিলেই হল। ম্যানুয়ালগুলোর কয়েকটা সেকশন আছে। সেই সেকশন অনুযায়ী আবার খোঁজা যেতে পারে বিশেষ বিশেষ কমান্ডকে।

১, এক্সিকিউটেবল বা চালানো যায় এমন প্রোগ্রাম, বাইনারি ফাইল বা শেল স্ক্রিপ্ট
২, সিস্টেম কল বা কারনেলের ফাংশন
৩, লাইব্রেরি কল বা সিস্টেম লাইব্রেরিগুলোর কোনো ফাংশন
৪, বিশেষ ধরনের কোনো ফাইল — /dev ডিরেক্টরিতে যেমন পাওয়া যায়
৫, কনফিগারেশন ফাইলের আকার-আকৃতি, লেখার নিয়ম — /etc/passwd ফাইল যেমন
৬, গেমস
৭, বিভিন্ন ম্যাক্রো প্যাকেজ এবং তার নিয়ম — যেমন man(7) বা groff(7)
৮, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড — যাদের শুধু রুট ব্যবহার করতে পারে
৯, বিশেষ ধরনের কারনেল রুটিন

এই সেকশনগুলোর নাম পড়ে যদি আপনার এরকম মনে হয়, এগুলো খায় না মাথায় দেয়, নো ফিয়ার, খায় না মাথায় দেয় এর চেয়ে খুব বেশি কিছু না-বুঝেই আমি যখন লিখতে পারছি, আপনি পড়তে পারবেন। তাতে চিন্তিত হবেন না। একবার বুঝলেন না তো পরের বার বুঝবেন। মাত্র একবার এসেছে গোটা পাঠমালায় এরকম একটা প্রসঙ্গও আমি পাইনি। বাজেবকাটা আমার বংশগত, জাতিগত। আর, তৃতীয় বিশ্বের নাগরিক, আমাদের মস্তিষ্কে

বাধ্যতামূলক ডবল-খাঁজ, সাহেবের বাচ্চাগুলো যখন একটা ভাষায় বাঁচে, আমাদের তখন গোটা তিনেক ভাষা তো জানতেই হয়, কী জানব আর কী জানব না এটা জানতে গিয়েই। একসঙ্গে সবটা বুঝে যাবেন নিখুঁত ভাবে — এটা একটা সাহেবী কুসংস্কার। যখন দেখবেন, জানেন কি জানেন না এটা একটু ঘোলাটে, কিন্তু মোটামুটি একটা আন্দাজ পেয়ে গেছেন কী কাজ বা কী ভাবে করতে হবে, তখনই বুঝতে হবে বুঝে গেছেন। বোঝা মানে ওটাই। কোনো কিছু পুরো বোঝা যায়না, কোনো সিগারেট পুরো টানা যায়না — এই একই যুক্তিতেই এই যুক্তিটা আবার পুরো সত্যি নয়।

শুধু একটা কথার উত্তর দিন। এই ৮ নম্বর সেকশনটা দেখুন, সিস্টেম অ্যাডমিন্সট্রেশন কমান্ড — যাদের শুধু রুট ব্যবহার করতে পারে। এর মানে কী? এই কমান্ডগুলোও এক একটা প্রোগ্রাম, এক একটা এক্সিকিউটেবল বাইনারি ফাইল, যাদের নাম লিখে এন্টার মারলে চলে, শুধু রুট ছাড়া অন্য কেউ চালাতে গেলে সিস্টেম তাকে মিষ্টি করে বকে দেয়। 'su' করে সুপারইউজার হয়ে নিতে হয় চালাতে চালাতে চাইলে। পাঁচ নম্বর দিনের ১১ নম্বর সেকশনে, রুটের এর '$PATH' বা পথনির্দেশটা দেখে আন্দাজ করার চেষ্টা করুন তো, ম্যানুয়ালের ৮ নম্বর সেকশনের এই সিস্টেম প্রোগ্রামগুলো কোন বা কোন কোন ডাইরেক্টরিতে থাকতে পারে? না পারলে, মাইরি বলছি, নিজের মনঃসংযোগের বিষয়ে সিরিয়াসলি কিছু ভাবার সময় এসে গেছে আপনার। ম্যানুয়ালের বা ম্যানপেজের ২ নম্বর সেকশন মানে সিস্টেম কল নিয়ে কিছু আলোচনা আছে এক আর দুই নম্বর দিনে। আর সেকশন তিন মানে লাইব্রেরি নিয়ে আলোচনা ছিল তিন নম্বর দিনে, আডার কম্পিউটার ভাষা নিয়ে ভাবনাচিন্তার প্রসঙ্গে। মনে পড়ছে?

আপনার গ্নু-লিনাক্স সিস্টেমের মধ্যে আপনার সাহায্যার্থে শুধু যে এই ম্যান ভরা আছে তাই নয়, আছে আরো একটা জিনিষ, তার নাম ইনফো (info)। এইমাত্র শেখা হেল্প দেখার তরকিবটা এখুন লাগিয়ে দেখুন 'info'-র উপর। এটা করেই চলুন, প্রতিবার, যখনি নতুন কিছু শিখছেন। '`info --help > info.help.text`' মেরে তৈরি করে দেখুন 'info'-র হেল্প ফাইল। সেই ফাইল থেকে কয়েক লাইন এখানে তুলে দিচ্ছি। এবার আর ব্যাখ্যা করছি না, দেখুন তো, আপনি নিজেই পারেন কিনা। আর এই চোকলার বদলে আসলটা পেতে চাইলে দিন '`man info`'। বা দিন '`info info`' — ইনফো কী করে সেটা ইনফো দিয়েই জানুন। আর 'info'-র ম্যানুয়ালের ফাইল পেতে চাইলে, কমান্ড দিন, '`man info > info.man.text`'। অথবা, 'info'-র নিজেকে দিয়েই নিজেকে চেনার ফাইল, '`info info > info.info.text`'। মানে স্ক্রিনের পাতাগুলোকে চালান করে দিচ্ছেন ডিস্কবাসী ফাইলে, 'info.man.text' বা 'info.info.text' নামের। পরে দেখব, এদের নিয়ে আরো বহু কিছু করা যায় এবং করতে হবে আমাদের। যাকগে, ইনফো সংক্রান্ত সেই লাইন কটা দেখে নিন। অনেক ছাঁটকাট করা কিন্তু।

```
info --help
Usage: info [OPTION]... [MENU-ITEM...]
Read documentation in Info format.
Options:
      --apropos=STRING         look up STRING in all manuals.
  -d, --directory=DIR          add DIR to INFOPATH.
      --dribble=FILENAME       remember user keystrokes in FILENAME.
  -f, --file=FILENAME          specify Info file to visit.
  -h, --help                   display this help and exit.
      --index-search=STRING    go to node pointed by index entry STRING.
  -n, --node=NODENAME          specify nodes in first visited Info file.
  -o, --output=FILENAME        output selected nodes to FILENAME.
Examples:
  info                       show top-level dir menu
  info emacs                 start at emacs node from top-level dir
  info emacs buffers         start at buffers node within emacs manual
  info --show-options emacs  start at node with emacs' options
  info -f ./foo.info         show file ./foo.info, not searching dir
```

ম্যান কমান্ডের খুঁটিনাটি এবার নিজেই পড়ে নিন, '`man man`' মেরে। দু-একটা খুব কাজের কথা বলে দিই। ম্যান দিয়ে যেকোনো কমান্ডের ম্যানুয়াল পেজ খুলে পড়বেন যখন স্পেসবার টিপলে এগোবেন, 'বি' বা 'B' সুইচটা টিপলে পিছোবেন, আর 'কিউ' বা 'Q' সুইচটা টিপলে বেরিয়ে যাবেন। অভিমন্যুর থেকে শিখুন, কোথাও ঢোকার আগে জেনে নেবেন বেরোনোর উপায়। কোনো একটা বিশেষ শব্দ বা প্রসঙ্গ খুঁজতে হলে লাগবে 'স্ল্যাশ' বা '/' সুইচটা। এটা

টেপা মাত্র দেখবেন, পাতার একদম নিচে এটা ফুটে উঠেছে, সেখানে টাইপ করে দিন আপনি যা খুঁজতে চান, এবং এন্টার মারুন। এই চারটে সুইচ শুধু এই ম্যান না, আরো বহু গ্নু-লিনাক্স কমান্ডের বেলাতেই সত্যি।

ম্যান পেজ খুলে কী করে খুঁজব সেটা জানলাম, কিন্তু ম্যানপেজ খুঁজব কী করে? ম্যানপেজ তো খুলবেন যদি কমান্ডটা জানেন। কিন্তু কমান্ডটা জানবেন কী করে? ধরুন একটা কমান্ড আপনি ব্যবহার করেছেন বা ব্যবহার করতে দেখেছেন, কাজের এলাকাটাও জানেন, কিন্তু নামটা মনে পড়ছে না। কিম্বা, কোনো একটা বিশেষ ধরনের কমান্ডের জন্য বা কোনো একটা বিশেষ প্রসঙ্গে কী কী ম্যানুয়াল পাতা দেওয়া আছে আপনার গ্নু-লিনাক্স সিস্টেমে, সেটা জানবেন কী করে? ধরুন, আপনি যা খুঁজতে চাইছেন সেটা কোনো কমান্ড নয়, একটা কোনো বিশেষ শব্দ, সেটা কোথায় কোথায় আছে, কী আছে, এটা জানবেন। ধরুন, কথাটা হল 'CD'। সিডি নিয়ে আপনার সিস্টেমের ম্যানুয়াল পেজগুলোয় কী কী বলা আছে আপনি জানতে চাইলেন। কমান্ড দিন '`man -k CD`'। দেখুন স্ক্রিনে একটা তালিকা ফুটে উঠেছে। গোটা তালিকাটার প্রত্যেক লাইনেই বাঁ দিকে একটা শব্দ, তারপর ব্রাকেটে একটা নম্বর, তারপর ডানদিকে লেখা বাঁদিকের ওই শব্দটা কী তাই নিয়ে একটা বিবৃতি। যেমন আমার সুজে ৮.২ সিস্টেমে আমি '`man -k CD`' করে যে তালিকাটা পেলাম সেটার থেকে একটা লাইন —

```
cdrecord (1)       - record audio or data Compact Discs from a master
```

একদম বাঁদিকে সিডিরেকর্ড বলে একটা কমান্ডের, মানে একটা এক্সিকিউটেবল প্রোগ্রামের বা বাইনারির নাম, তারপর ব্রাকেটে লেখা (১), কারণ, একটু আগের ম্যানুয়াল পেজগুলোর সেকশনের তালিকাটা দেখুন, ১ নম্বর সেকশনে থাকার কথা এক্সিকিউটেবল প্রোগ্রাম বা শেল কমান্ডগুলোর। এবং ডানদিকে লেখা সিডিরেকর্ড সম্পর্কে একটা কুচো মন্তব্য। 'cdrecord' প্রোগ্রামটা ঠিক কী করে। এবার আপনি যদি এই 'cdrecord' সম্পর্কে আরো জানতে চান তাহলে কমান্ড দেবেন, '`man 1 cdrecord`'। এবার ঝট করে একটা উত্তর দিন তো, একটু আগে 'passwd' কমান্ডের ম্যানুয়াল পড়তে গিয়ে আমরা '`man 5 passwd`' কমান্ড দিয়েছিলাম কেন, ১ থেকে ৫ এইরকম সংখ্যাগুলো বদলে যাচ্ছে কেন? ম্যানপেজের সেকশনের তালিকার সঙ্গে মিলিয়ে দেখুন তো, বুঝতে পারেন কিনা।

এখানে একটা কোমল ঘাপলা আছে, সেটা ফিল করার জন্যে কমান্ড দিন, '`man -k X`', এন্টার মারুন। এবার স্বচ্ছন্দে একটা সিগারেট ধরান। কারণ, বেশ কিছু সময় ধরে এক স্ক্রিন আপনি এক পলকের বেশি দেখতে পাবেন না। পাতার পর পাতা। '`X`' মানে মূলত এক্স-উইনডোজ বা গুই নিয়ে আলোচনা। আমার সুজে সিস্টেমে এই নিয়ে মোট ম্যানপেজ আছে ২৮২৯ খানা। সিস্টেমে ম্যানুয়ালের মোট সাইজ বুঝতে পারছেন? একটু আগে '`CD`' খুঁজে ম্যানুয়াল পেয়েছিলাম ৩৮, তার মানেও এক স্ক্রিনের মানে ২৫ লাইনের বেশি। অর্থাৎ, যেখানে গিয়ে থামল, তার আগে ইতিমধ্যেই অনেকগুলো লাইন পেরিয়ে গেছে। তাহলে খুঁজব কী করে? ডিসপ্লেটা যখন থেমে যাবে — তালিকার শেষ পাতায়? এটা একটা স্ট্যান্ডার্ড কথা হল? লেস (less) আছে কী করতে? লেস নিয়ে একটা মজার কথা চলে, লেস ইজ মোর দ্যান মোর (more)। মোর বলে একটা পেজার আগেই ছিল, সেই পুরোনো মোর-এর সঙ্গে আরো কিছু কাজ এনে তৈরি করা হয় এই 'less' নামের পেজারটা। পেজার বলতে একটা সফটওয়ার যা কোনো টেক্সটকে পাতা বাই পাতা পরপর দেখিয়ে যায়। আর বলা যাবে না, স্ক্রিন একটি নদীর নাম, স্বপ্নের ছন্দে সে বহিয়া যায়। তাকে বেঁধে ফেলা হয়েছে লেস-এর শেকলে। লেস-এর পোষাকের অনির্বচনীয়তা হয়তো নয়, কিন্তু কাজ চলে যাবে।

'`man -k CD | less`' — মানে, মূল কমান্ডের আউটপুট বা ফলাফলটাকে পাইপ করে দিলাম 'less' দিয়ে। এবার ধীরে সুস্থে পাতাটাকে নামান। ঠিক একটু আগে ম্যান পেজের বেলায় যে চারটে সুইচ বলেছিলাম, স্পেসবার, বি, কিউ এবং স্ল্যাশ — সেই চারটেই একই কাজ করে এখানেও। ইনফো-তেও করে, এবং আরো অনেক কমান্ডে। আচ্ছা, ম্যান-এর ফলাফলে মোট কত লাইন ছিল, ২৮২৯, বললাম কী করে? গুণে গুণে? প্লিজ, আমার বাবার দিককার কাস্ট কায়স্থ, মানে ক্লার্ক, কিন্তু অতো মাছি আমি কোনোদিন মারিনি। লাইন গোনার কোনো প্রোগ্রাম আছে?

এবার দেখুন, লেস করেও যদি খুঁজতে চান, বহু বহু পাতা অনেক ক্ষেত্রেই, তখন খুঁজে পাওয়াটা অসম্ভব আপনি যা খুঁজছেন। তখন অন্য একটা তরকিব কাজে লাগান। তার নাম গ্রেপ (grep)। ধরুন, আপনি জানতে চাইছেন, গ্নু-লিনাক্স সিস্টেমে অডিয়ো সিডি নিয়ে কী করা যায়। ম্যান-এর আউটপুটটাকে পাইপ করে দিন 'grep'-এ। আগের

কমান্ডটাকেই বদলে কমান্ড দিন, 'man -k CD | grep 'audio''। এর থেকে স্ক্রিনে যা পাবেন সেটা এই পাতার আড়ে পাশাপাশি ধরানোর জন্যে একটু কেটেছেঁটে অনেকটা এইরকম —

```
cdda2wav(1) -        a sampling utility that dumps CD audio into wav files
cdrecord(1) -        record audio or data Compact Discs from a master
cdparanoia(1) -      an audio CD reading utility with data verification
cdrdao(1) -          writes audio CD-Rs in disc-at-once mode
cda(1) -             Compact disc digital audio player utility
xmcd(1) -            CD digital audio player utility for X11/Motif
```

এবার ধরুন, এইটা যখন স্ক্রিনে দেখছেন, হঠাৎ খেয়াল করলেন, আরে, চারপাশটা এত নিস্তব্ধ লাগছে কেন, স্ক্রিনে 'date' মেরে বা ঘড়িতে দেখলেন, আরে আড়াইটে বেজে গেছে, ধুর ছাতা, এত রাতে অডিয়ো সিডির এত প্রোগ্রাম খুঁজে পেয়েই বা কী হল, বাজানো তো যাবেনা। পাড়ার আগেও বাড়ি, বাজালেই বাড়ি মাথায়, মানে মাথায় বাড়ি। তারপর আবার কালকের অফিস। একটা দীর্ঘশ্বাস ফেলে আপনি লাইনকটা তুলে রাখলেন একটা ফাইলে, কমান্ড দিলেন 'man -k CD | grep 'audio'> audio.cd.program.text'। এবার দেখে নিতে চাইলেন, ফাইলটা ঠিক সেভ হয়েছে কিনা, বা তার সাইজ ঠিক কতটা। সেই কমান্ডটা আমরা আগেই দেখেছি, একটু বদলাচ্ছি এবার। খেয়াল করুন। কমান্ড দিলেন 'ls -sh audio.'। শেষে কিন্তু একটা বিন্দু আছে। কমান্ডটা টাইপ করে দিলেন, কিন্তু সাবধান, এবার কিন্তু এখনি এন্টার মারবেন না, অহিংস হতে শিখুন। বা, হিংস্রতাটার চেহারা বদলে দিন, তাতে সব কূলই বজায় থাকবে। এন্টারের (Enter) জায়গায় মারুন ট্যাব (Tab)। কিন্তু কেন?

এটা ব্যাশ শেলের কমান্ড কমপ্লিশন বা নিজে থেকে আদেশের শূন্যস্থান পুরন করে দেওয়ার অভ্যেশ। সেটাকে আপনি কাজে লাগাচ্ছেন। যদি এই ডিরেক্টরিতে আর কোনো ফাইল না থাকে যার নাম 'audio.' দিয়ে শুরু, তাহলে ইতিমধ্যেই দেখুন, আপনার স্ক্রিনে এসে গেছে, 'ls -sh audio.cd.program.text'। মানে, ব্যাশ নিজেই পুরন করে দিয়েছে আপনার কাঙ্খিত গোটা কমান্ডটা। আর যদি ডিরেক্টরিতে একের বেশি ফাইল থাকে যারা প্রত্যেকেই 'audio.' দিয়ে শুরু, তাহলে স্ক্রিনে পাবেন আপনার কমান্ডটা যতটা আপনি টাইপ করেছেন, তার নিচে ওই কটা ফাইলের নাম। এবার আপনি টাইপ-করা অংশটুকুর পরে সেই কটা অক্ষর যোগ করে দেবেন, যে অব্দি গেলে এই নামটা সম্পূর্ণ নিজস্ব চেহারা পেয়ে যাবে, আর অন্য ফাইল-নামের সঙ্গে কোনো মিল থাকবে না। তারপর গোটাটুকু ভরে দেবে ব্যাশ নিজেই। যেদিন, কমান্ড দেওয়ার সময়, একটা দুটো বর্ণ টাইপ করার পরই আপনার বাঁ হাতের কড়ে আঙুল ট্যাব সুইচের দিকে যেতে শুরু করবে, সেদিন বুঝবেন আপনি ব্যাশাক্রান্ত হতে শুরু করেছেন। ধরুন, আপনি 'ls -sh audio.' অংশটা টাইপ করার পর দেখলেন স্ক্রিনে রয়েছে টাইপ-করা কমান্ডটুকু এবং নিচে দুটো ফাইলের নাম, 'audio.cd.program.text' এবং 'audio.filetypes.list'। এবার আপনি জাস্ট একবার 'সি' (c) সুইচটা টিপে আর একবার ট্যাব মারলেই ব্যাশ গোটা কমান্ডটা ফুটিয়ে তুলবে, কারণ, 'c'-র থেকে দুটো ফাইল আলাদা, ওই 'c'-টা আছে শুধু আপনার দেখতে চাওয়া ফাইলের নামেই। এবার এন্টার মারলেই স্ক্রিনে ফুটে উঠবে ফাইলটার নাম এবং মানববোধ্য সাইজ, আগেই তো বলেছি।

ব্যাশে আর এক খাবলা অভ্যেশ করে নেওয়া যাক। ধরুন, সাইজটা দেখার পর নিজের কৃতিত্বে মুগ্ধ আপনার মনে হল, পরের বার আর এতোটা টাইপ যাতে না করতে হয়, তার জন্যে 'audio.filetypes.list' ফাইলটার নামটা তো বদলে দিলেই হয়, ব্যাশের কমান্ড কমপ্লিশনের পুরো মজাটা যাতে পাওয়া যায়। অন্য নামে, 'sound.filetypes.list', ফাইলটা কপি করতে চাইলেন। কপি করার কমান্ড 'cp' তো আমরা জানি, পাঁচ নম্বর দিনে ৩ নম্বর সেকশনে ছিল। কমান্ড দিলেন, 'cp audio.filetypes.list sound.filetypes.list'। এবার এন্টার মারলেন, কমান্ড প্রম্পট সঙ্গে সঙ্গে চলে এল, মানে কাজটা হয়ে গেছে। কাজটা না হলে জানাতে পারে, চুপচাপ ফিরে এল মানে কাজ সমাধা, ব্যাস — ব্যাশ শেল এভাবেই কাজ করে, বেশি কথা বলেনা। এবার একবার 'ls' মেরে দেখে নিতে চাইলেন পুরোটা। দেখতে গিয়েই মনে পড়ল, ও হরি, পুরোনো 'audio.filetypes.list' ফাইলটাও তো রয়ে গেল। কোই পরোয়া নেই। ফাইল ওড়ানোর কমান্ড 'rm' — সেটাও আপনি জানেন। সহজেই উড়িয়ে দিলেন ফাইলটা। 'rm audio.filetypes.list'। এন্টার মারলেন। প্রম্পট ফেরত এল নিঃশব্দে। মানে ফাইল উড়ে গেছে। আর একবার 'ls' , মিলিয়ে নিলেন। তারপর মাথায় এল, আরে প্রথমেই 'cp' করে, তারপর 'rm' না-

করে, একবারেই তো করা যেত গোটা কাজটা, 'mv' কমান্ড দিয়ে, সেই কমান্ডটাও দেখেছি আমরা পাঁচ নম্বর দিনে। কমান্ডটা ফাইলকে অন্য ডিরেক্টরিতে সরায় বা নাম বদলায়। দুটো কমান্ডের বদলে এখন একটাই কমান্ড দিতে পারতেন, '`mv audio.filetypes.list sound.filetypes.list`'। যাকগে। পরে হবে ওসব।

শুতে যাওয়ার আগে এখনো একটা কাজ বাকি আছে আপনার। আপনি একবার দেখে নিতে চাইলেন, ফাইলটা ঠিক কোথায় আছে, আপনার হোম ডিরেক্টরির কোথায়? কোন সাবডিরেক্টরিতে? এর জন্যে আপনি যে কমান্ডটা ব্যবহার করবেন সেটা হল '`pwd`'। মানে, কোন ডিরেক্টরিতে কাজ করছি সেটা দেখাও। প্রিন্ট ওয়ার্কিং ডিরেক্টরি। এর পরে আপনি যে ডিরেক্টরিতে সেভ করেছেন ফাইলটা, তার ঠিকানাটা ফুটে উঠবে। ধরুন এটা আমি করছি। আমি ফাইলটা সেভ করেছি গার্বেজ বলে একটা ডিরেক্টরিতে। এই গার্বেজ ডিরেক্টরিটা, ধরুন অডিয়ো নামের ডিরেক্টরিতে। যে অডিয়ো ডিরেক্টরিটা আছে আমার হোম ডিরেক্টরির মাল্টিমিডিয়া বলে একটা ডিরেক্টরিতে। এই অবস্থায় আমি '`pwd`' মারলে স্ক্রিনে ফুটে উঠবে, `/home/dd/multimedia/audio` — এটাই এই মুহূর্তে আমার কাজের ডিরেক্টরি। কাল এসে আর খুঁজে পেতে এবং অডিয়ো সিডির গান বাজাতে অসুবিধে হবেনা।

পাঁচ নম্বর দিন আর আজকের শুরু থেকে আমরা যে ছাবিবশটা কমান্ড এখনো পেয়েছি সেগুলো হল, 'mkdir', 'rmdir', 'cd', 'ls', 'mv', 'rm', 'cp', 'grep', 'tar', 'bzip2', 'cat', 'who', 'w', 'wc', 'less', 'more', 'man', 'info', 'pwd', 'lilo', 'init', 'login', 'su', 'useradd', 'passwd', 'echo'। কয়েকটা নিছক উল্লেখই করা হয়েছে, কয়েকটা নিয়ে প্রাথমিক আলোচনাও হয়েছে। এর মধ্যে আবার 'mkdir', 'rmdir', 'cd' কমান্ডগুলো আলোচনায় আনতে পারিনি কারণ এখনো আমরা রিলেটিভ এবং অ্যাবসলিউট পাথ, আপেক্ষিক এবং চূড়ান্ত পথনির্দেশ বলতে কী বোঝায় সেটা জানিনা। এক ডিরেক্টরি থেকে অন্য ডিরেক্টরিতে ফাইল নড়ানোর কাজে 'mv' কমান্ডটাকে কী ভাবে ব্যবহার করা যায় সেটাও বলা যায়নি একই কারণে। সেই পাথ বা পথ-এর কথা বলতে গেলে একটু ফাইল সিস্টেম জানতেই হবে। অচিরেই আসবে, ফাইল, ডিরেক্টরি সিস্টেম, তারপর ফাইলসিস্টেম। কমান্ডগুলোর মধ্যে কয়েকটা আবার, 'lilo', 'init', 'login', 'su', 'useradd', 'passwd' এইগুলো সিস্টেম কমান্ড। এদের নিয়ে কাজ করার আছে অনেককিছু সিরিয়াস জিনিষ আছে বোঝার।

কিন্তু যা বলেছিলাম, এই লেখাটায় আসলে আমরা দুজনেই খেলছি, নইলে খেলাটাই হবেনা। আপনি এখন থেকে যত দূর সম্ভব তার চেয়ে একটু বেশি পরিশ্রম করে এই কমান্ডগুলোয়, এদের ম্যানুয়ালে আর ইনফোপাতায় অভ্যস্ত হতে শুরু করুন আপনার সিস্টেমে। যত পারা যায় নানা আলাদা আলাদা অপশান কমান্ডের সঙ্গে লাগিয়ে লাগিয়ে দেখুন, কী হয় কী হয় না। যদ্দূর সম্ভব ম্যানুয়ালের সঙ্গে বাঁচতে শিখে নিন। সামনে দুঃখের দিন আসছে। এখানে যত বেশি পরিশ্রম করবেন, তত আপনার পরে সুবিধে হবে। অনেকটা গানের রিওয়াজের মত। টাচ-টাইপিং বা দশ-আঙুলে টাইপ করার মত। এই টাচ-টাইপিং-এর জন্যে একটা চমৎকার গ্নু সফটওয়ার আছে, জিটাইপিস্ট (gtypist)। পরে আমরা আসব সেসবে।

এখানে হঠাৎ একটা কথা মাথায় এলো, এই যে বারবার আ মি লিখে যাচ্ছি পেজ-আপ পেজ-ডাউন করে পড়ার কথা, তার মানে আমি কি ধরে নিচ্ছি যে আপনি এই পাঠমালাটা মেশিনেই পড়ছেন? যদি তাই হয়, তাহলে আপনার ওএস কী? উইনডোজ? যদি তাই হয়, তাহলে তো আপনি পড়াকালীন অভ্যেশগুলো করতে পারবেন না? যদি গ্নু-লিনাক্স হয়, তাহলে সেখানে শুধু কমান্ড প্রম্পট নয়, একটা এক্স-উইনডোজ-ও থাকতে হবে। গুহনোম হোক, কেডিই হোক, ব্ল্যাকবক্স বা ফ্লাক্সবক্স বা অন্যকিছু যাই হোক। যে ফাইলে রাখা হচ্ছে দিনগুলোকে, সেগুলো পিডিএফ। যা কমান্ড লাইন থেকে প্রিন্ট করা যায়, কিন্ত কমান্ড মোডে পড়তে পারার কোনো প্রকৌশল আছে বলে জানিনা। কার্সেস দিয়ে ফ্রেমবাফার দিয়ে কিছু কখনো তো দেখিনি। এফবিআই (fbi) দিয়ে যেমন কমান্ড প্রম্পটে গ্রাফিক্স দেখি, বা এমপ্লেয়ার (mplayer) যেমন কমান্ড মোডেই দিব্য সিনেমা দেখায়। গান শোনার অমন প্যাকেজ আছে কয়েক লাখ। অর্থাৎ, গোটাটা মিলিয়ে দাঁড়াল, আপনার মেশিনে ওএস-টা গ্নু-লিনাক্স, এবং সেখানে এক্স-উইনডোজ চলছে। এই অবস্থায় আপনি একটা টার্মিনাল অ্যাপ্লিকেশন খুলে নিন। কেডিই (KDE) হলে কনসোল (konsole) বা গুহনোম (GNOME) হলে জিটার্ম (gterm) হল ডিফল্ট টার্মিনাল অ্যাপ্লিকেশন। টার্মিনাল অ্যাপ্লিকেশন মানে একটা জানলা বা উইনডো, যেখানে বিশুদ্ধ কমান্ড প্রম্পটটাকে সিমুলেট বা নকলনবিশি করা হয়। ঠিক যেভাবে কমান্ড প্রম্পট কাজ করে, এইটাও করে। ডিসপ্লে এক ইউজার থেকে অন্য ইউজারে বদলালে একটু খোঁচ হয়, 'su' দিয়ে কাজ করার

সময়, কিন্তু সে অনেক পরের ব্যাপার। সেখানে কমান্ডগুলো পড়তে এবং করতে থাকুন। আর সঙ্কর্ষণরা যেমন প্রাণপণ দৌড়চ্ছে হার্ড প্রিন্টে মানে বই আকারে বার করার জন্যে, সেটা যদি শেষ অব্দি ঘটে যায়, টাকাটা তো বিরাট, তাহলে কমান্ড প্রম্পটেই কাজ করতে পারবেন পড়ার পাশাপাশি। গুই বলে কিছু আছে কিনা, মনেই রাখতে হবেনা।

এখানে একটা কেজো সমস্যার কথা বলি, যে জিএলটি ক্লাসের সূত্রে প্রথমে ক্লাসনোটস এবং তারপর ক্রমে গতরবৃদ্ধি হতে হতে ক্রমে এই গাবদা আকারে চলে আসছে, তার ক্লাস চলাকালীনই অশেষ সমস্যাটার কথা বলেছিল। ও তখনো কেবল সিস্টেম অল্পস্বল্প বুঝতে শুরু করেছে, চেনাজানা তৈরি হয়নি, ওর নাকি এটা খুব ধাঁধার মত লাগছিল। কোন কমান্ডগুলো শিখব, কাদের ম্যানুয়াল পড়ব? কমান্ডগুলোর বিবরণ ম্যানুয়াল পেজে লেখা আছে, 'ls' জানব 'man ls' কমান্ড দিয়ে, সেটা বুঝলাম। কিন্তু এই 'ls' কমান্ডটার কথা জানব কী করে? কাকে ম্যান দিয়ে খুঁজতে হবে, সেটা বুঝব বা জানব কী করে? আপাতত এটুকুই মাথায় রাখুন, '/bin' ডিরেক্টরিতে যে যে ফাইল আছে, তাদের ম্যান পেজ পড়া দিয়ে শুরু করা ভালো। এই bin-বিরাজীরা হলেন একজিকিউটেবল বা বাইনারি ফাইল। মানে প্রোগ্রাম, যাদের চালানো যায়। আর একটু ভালো করে এটা বুঝে যাব আমরা আজকের আলোচনাতেই। এই ডিরেক্টরির প্রোগ্রামগুলো পরপর ম্যানপেজ পড়ে যেতে পারেন। আর, সব সময়েই কি লোকে পড়ে নাকি, ছি, গান বাজনা সিনেমা এরা কী জন্যে। এই মাল্টিমিডিয়া ঘাঁটার চেষ্টা করা, বা দুচারটে গেম খেলা, সিস্টেমের সঙ্গে ঘনিষ্ঠতা বাড়ানোর একটা ভালো রাস্তা। শুধু দেখবেন শেষ অব্দি গোঁফের আমি গোঁফের তুমি না-হয়ে যায়। কিন্তু, কারা বিরাজ করেন এই '/bin' ডিরেক্টরিতে, সেটা জানবেন কী করে? সিস্টেমের যে কোনো জায়গায়, যে কোনো ডিরেক্টরিতে দাঁড়িয়ে কমান্ড দিন 'ls /bin'। ফাইলে রাখতে চাইলে, স্ক্রিনের বস্তুটাকে রিডাইরেক্ট করে দিন টেক্সট ফাইলে, 'ls /bin > binary.file.list', বা যে নামে চান ফাইলটাকে। একটা কথা মাথায় রাখুন, এই সমস্যাটাও হতে দেখেছি জিএলটি ইশকুলে, যাকে তাকে কিন্তু 'cat' করা যায়না। বা রিডাইরেক্ট করে ফাইল বানানো যায়না। যেমন, একটা জ্যান্ত ক্যাটকে ক্যাট করার প্রোগ্রাম লেখার কাজ চলছে, এখনো শেষ হয়নি, কাজটা তো সোজা নয়, লেজ নখ বিঁটলেমি এবং ইঁদুরের স্বপ্ন সহ একটা আস্ত জ্যান্ত বেড়াল। গ্নু-লিনাক্স সিস্টেমে কাজ করতে করতে মাঝে মাঝে মনে হতেই পারে, জ্যান্ত ক্যাটের বাইরে আর বাইরে প্রায় সবকিছুকেই 'cat' করা যায় (সাউন্ডকার্ড মাউস ভৌত হার্ডডিস্ক ইত্যাদি 'cat' করার কায়দা আমরা পরের সাত নম্বর দিনেই দেখব) কিন্তু সেটা সত্যি নয়। মানে, চাইলে 'cat' করা যায় সবাইকেই, কিন্তু একটা বাইনারি ফাইলকে যদি আপনি 'cat' করেন বা রিডাইরেক্ট করে একটা ফাইল বানান, এবং তারপর সেটাকে 'less' দিয়ে পড়ার চেষ্টা করেন, শূন্য নম্বর দিনের ভিশুয়াল চ্যাংড়ামির গোত্রের জিনিষপত্তর পাবেন। বাইনারি ফাইল পড়া যায়না, কিন্তু বাইনারি ফাইল মানে তো প্রোগ্রাম, সেই প্রোগ্রামের ম্যানুয়াল পেজ কিন্তু দিব্য পড়া যায়। তাকে 'cat' করুন, বাইনারি ফাইলটাকে নয়।

২ ।। ফাইল

প্রাথমিকভাবে, ফাইল একটা বিমূর্ত জিনিষ। পাঠমালার এক নম্বর দিনে আমরা একটা সাঙ্কেতিক বা সিম্বলিক ব্যবস্থার কথা আলোচনা করেছিলাম, বিমূর্ত একটা চিহ্ন-কাঠামো, যা দিয়ে আমি ভৌত মেশিনটাকে বুঝছি আমরা। এই যে হার্ড ডিস্ক বুঝছি, '/dev/hda' বা '/dev/hdb' দিয়ে, ফ্লপি ডিস্ক '/dev/floppy' দিয়ে, সিডিরম '/dev/cdrom' দিয়ে — এই নামগুলো কী? এক একটা নাম, চিহ্ন, যার পিছনে একটা ভৌত বাস্তবতা রয়েছে, যাকে আমরা ভুলে যেতে পারছি কারণ গোটাটাকে নিজের মাথায় রেখে দিচ্ছে কারনেল তথা ওএস। সিম্বলাইজেশন বা প্রতীকীকরণটা তৈরি হতে শুরু করে ফাইল থেকে। একটা ফাইল, তার নিজের নামের মূর্ত পরিচিতির আশ্রয়ে বিমূর্ত করে তোলে কিছু বাইট সমাহারকে যাদের ভৌত বাস্তবতাটাকে আমরা তখন আর ভাবছি না, আমার কাছে তখন ফাইল মানেই ডিস্কের গায়ে বরাদ্দ করা ওই বাইট সমাহারে লিখিত তথ্যটা। এতে ওই ঘসঘস করে ঘুরতে থাকা চৌম্বক প্রলেপ লাগানো গোলাকৃতি প্ল্যাটারগুলোকে, তাদের ভীষণ কাছ দিয়ে পরিক্রমণশীল হেডগুলোকে, তাদের নড়াচড়াকে, চৌম্বক তলের উপর ওই হেড দিয়ে তথ্য লেখা বা পড়ার কায়দাকে, ভুলে থাকছি আমি। শূন্য নম্বর দিনে এই নিয়ে আলোচনাটা মনে করুন। যে বাস্তবতাটাকে ফাইল গোপন করে রাখে তার নামের পিছনে। এখন থেকে ওই বাইট সমাহার পরিচিত হবে ওই নাম দিয়ে। মাথায় রাখুন, তথ্য মানে তথ্যহীনতাও হতে পারে। ব্ল্যাংক বা ফাঁকা ফাইল তৈরি করতে শিখব আমরা, যেখানে কোনো তথ্য আপাতত লেখা নেই, কিন্তু চাইলেই লেখা যেতে পারে।

প্রথমে এই ফাইল নামকরণের প্রক্রিয়াটাকে বোঝার চেষ্টা করা যাক। ফাইলের নাম শুধু যে আপনি দেন তা নয়, আপনি যে প্রোগ্রামগুলো চালাবেন তারা বহু ফাইল তৈরি করে, অপারেটিং সিস্টেম তার নিজের কাজ করার প্রক্রিয়ায় বহু ফাইল বানায়। ফাইল বানানো মানে নির্দিষ্ট কিছু ভৌত বাইট সমাহারে গেঁথে ফেলা এবং গেঁথে নেওয়া তথ্য সমাহারের জন্যে আলাদা একটা নামের আলাদা পরিচিতি তৈরি করা। অজস্র ফাইল থাকে যারা একটা প্রসেস চলাকালীন তৈরি হয়, প্রসেস মরে গেলে তারাও নিশ্চিহ্ন হয়ে যায়, তাদের আলাদা করে আর সেভ করা হয়না ডিস্কে। প্রসেসটা যতক্ষণ চলে ততক্ষণই তারা বেঁচে থাকে, র‍্যামে বা ভার্চুয়াল মেমরির সোয়াপ ফাইলে। সোয়াপ ফাইল মানে, বলেছি, হার্ডডিস্কে আলাদা করে রাখা অস্থায়ী ফাইল অস্থায়ীভাবে লেখার জায়গা। ভারচুয়াল মেমরি বা ভৌতিক স্মৃতিতে রাখা অস্থায়ী ফাইলের ধারণাটা ভালো করে বোঝা যায় কোনো কিছু লেখার সময়। লিখছি মানেই নতুন শব্দগুলো জমা হচ্ছে। কিন্তু জমা হচ্ছে কোথায়? ফাইলটার স্থায়ী শরীরে নয়, সেটা বোঝাই যায়। এখুনি যদি আপনি বেরোতে চান, আপনাকে সিস্টেম জিগেশ করবে, ফাইলটা কি সে সেভ করবে? তার মানে এখনো তাকে সেভ করা হয়নি। হঠাৎ কারেন্ট চলে গেলে অনেক সময় যেমন এই না-সেভ করা অংশটা আর পাওয়া যায় না। তার মানে, এই অংশটা অস্থায়ী ভাবে লেখা থাকে কোথাও। এবং, সেভ করা মানে, অস্থায়ী অংশটাকে স্থায়ী ফাইলের শরীরে ঢুকিয়ে দেওয়া। অর্থাৎ, লেখার অস্থায়ী অংশটা, অস্থায়ী ভাবেই কোথাও একটা রাখা থাকছে সেই সময়টুকুর জন্যে। সেটাই ওই অস্থায়ী ফাইল, যাকে লেখা থাকছে ভারচুয়াল মেমরির সোয়াপ ফাইলে। হোক অস্থায়ী, তাও তো সেটা একটা লেখা, ওই সময়টুকুর জন্যে থাকতে গেলেও একটা নিজস্ব পরিচিতি নিয়ে থাকতে হবে তাকে। ধরুন, একইসঙ্গে দুটো ফাইল এডিট করছেন, একটার অস্থায়ী অংশ তো অন্যটায় সেভ করা হচ্ছেনা, তার মানে, ওই অস্থায়ী সময়টুকুর জন্যেই তার একটা পরিচিতি আছে — অস্থায়ী একটা নামও আছে, সিস্টেম যে অস্থায়ী নামের ফাইলে তথ্য লিখছে বা পড়ছে। কাজ মিটে গেলেই মুছে দিচ্ছে। আপনাকে জানতে হচ্ছে না, কিন্তু নাম একটা আছেই।

২.১।। ফাইলনাম এবং এক্সটেনশন

এই নাম কী ভাবে দেওয়া হবে তার খুঁটিনাটি নিয়ম আবার ওএস থেকে ওএস-এ আলাদা। ফাইলনামে কটা অব্দি চিহ্ন ব্যবহার করা যাবে সেটা অপারেটিং সিস্টেম থেকে অপারেটিং সিস্টেমে ৮ থেকে ২৫৫ অব্দি হয়। কোন কোন সংখ্যা বা অক্ষর বা অন্য কোনো চিহ্ন ব্যবহার করা যাবে তারও নিয়মের কিছু পার্থক্য থাকে এক ওএস থেকে অন্য ওএসে। কোনো কোনো সিস্টেমে বড় হাতের বা ক্যাপিটাল কেস অক্ষর এবং ছোট হাতের বা স্মল কেস অক্ষর আলাদা বলে গণ্য হয়, কোথাও হয়না। যেমন গ্নু-লিনাক্সে ‘dog’ বা ‘Dog’ বা ‘dOg’ বা ‘doG’ বা ‘DOg’ বা ‘DoG’ বা ‘dOG’ বা ‘DOG’ প্রত্যেকেই আলাদা ফাইল, কিন্তু এমএসডস বা এমএস-উইন্ডোজে এই সব কটা নামই চিহ্নিত করবে একই ফাইলকে, সেটা সে নিজে দেখাবে ‘dog’ বা ‘Dog’ বা ‘DOG’ চেহারায়। আগের আটটার যেকোনো একটা নামে একটা ফাইল থাকলেই সেই ডিরেক্টরিতে আর ওই আটটার অন্য কোনো নামে কোনো ফাইল তৈরি করতে দেবেনা। এমএসডস বা উইন্ডোজ ব্যবস্থায় ফাইলনামে দুটো অংশ থাকে। কাঠামোটা হল ‘*.*’। পড়ার সময় সচরাচর ডাকা হয় ‘স্টার-ডট-স্টার’ বলে। এই কাঠামোয় প্রথম ‘*’-টা এমএস-ডসে হত ৮ বা তার কম অক্ষরের, পরের ‘*’-টা তিন বা তার কম। প্রথম ‘*’-টাকে বলে ‘ফাইলনেম’ (filename), আর পরের ‘*’-টাকে বলে ‘এক্সটেনশন’ (extension)। উইন্ডোজে একই সঙ্গে প্রতিটা ফাইলনামের দুটো সংস্করণ লিখে রাখা হয় — একটা ওই এমএসডসের ছক মেনে, ‘৮.৩’ ছকে, অন্যটা ২৫৫ চিহ্নের, ওই ছক না-মেনে। দুটোই ব্যবহার করা যায়। উইন্ডোজের এমএস-ডস প্রম্পটে যেমন দেখবেন, দুটো নামই দেখায়। প্রাথমিক ভাবে বাঁদিকে দেখায় ডস ছকের নামটা, একদম ডানদিকে থাকে উইন্ডোজ নামটা। যদি উইন্ডোজে ‘C:\’-এ, মানে রুট ডিরেক্টরিতে, দাঁড়িয়ে ‘cd My Documents’ লিখে এন্টার মারেন, ডস কমান্ড প্রম্পট আপনাকে জানাবে, ‘Too many parameters – Documents’। মানে, সিস্টেম এই নামটার মানেই বুঝতে পারছে না, সে তো আভ্যন্তরীণ ভাবে ফাইলনামকে চেনে এমএস-ডস ছকে। কিন্তু, ‘cd mydocu~1’ লিখে এন্টার মারলে পৌঁছে দেবে ‘C:\My Documents\’ ডিরেক্টরিতে। কিন্তু উইন্ডোজ এক্সপ্লোরার দিয়ে সিস্টেম ব্রাউজ করার সময়ে এমএস-ডস ছকটা টেরই পাবেন না, গোটা নামটাই দেখাবে। আসলে এটা কিন্তু তার দেখানোর নাম, সিস্টেম তাকে বোঝে ‘mydocu~1’ দিয়ে, যার অক্ষরের সংখ্যা ‘৮.৩’, সেই এমএস-

ডস নিয়ম। ফাইলসিস্টেম আলোচনায় ফ্যাট-এর (FAT — **F**ile-**A**llocation-**T**able) কথা আসবে। উইন্ডোজের ফ্যাটে আসলে এই দুটো ছকেই নামটা লিপিবদ্ধ থাকে। যাকগে, এটা আমাদের পাঠমালার আলোচনার এলাকা নয়।

গ্নু-লিনাক্স সিস্টেমের কাছে এই এক্সটেনশন ব্যাপারটার কোনো মানেই নেই। মানেটা আপনার কাছে। একই ধরনের অনেকগুলো ফাইলকে একই এক্সটেনশন দিয়ে রাখলে পরে আপনার চিনতে সুবিধে হবে, নামের সঙ্গে পদবিতে যে সুবিধেটা হয়। যে কারণে আমরা প্রবন্ধগুলোর নাম দিচ্ছিলাম 'essay.1.text', 'essay.2.text', 'essay.3.text', ... ইত্যাদি। এখানে দেখুন, নামগুলোতে দুটো নয়, তিনটে করে খণ্ড। এই নামগুলো সংখ্যা ব্যবহার করেছে, সেখানে শুধু অক্ষর ব্যবহার করা যেতে পারত। ব্যাশের কথা মাথায় রেখে, এদের নামগুলো দিতে পারতাম, '01.essay.text', '02.essay.text', '03.essay.text', ... ইত্যাদি। তাতে ব্যাশ কমান্ড কমপ্লিশনের কাছে শুধু গোড়ার দুটো অঙ্ক দিয়ে ট্যাব মারলেই কাজ চলে যেত। আর প্রথম ডট বা বিন্দুটা দেওয়ার একটা কারণ এই যে, 'ls' কমান্ড দিয়ে পাওয়া তালিকাটায় পরপর দেখতে বেশ সুন্দর লাগে। একদম ডানদিকের যে অংশটা '.text' — এটা দেখে আপনি এদের টেক্সট ফাইল বলে চিনতে পারেন। খেয়াল করুন, সিস্টেম নয়, আপনি। আপনি এই এক্সটেনশন বা পদবিটা তিদে পারতেন '.txt', বা '.essay', দিতে পারতেন '.probondho', বা যা আপনার খুশি। অনেক লোকের অনেক ব্যবহারকে মিলিয়ে এর আবার কিছু প্রথা বা কনভেনশনও গড়ে উঠেছে। যেমন, '*.c' মানে সি প্রোগ্রামের কোড লেখা ফাইল, '*.o' মানে কম্পাইলেশনের সময়ে তৈরি অবজেক্ট ফাইল, '*.html' মানে নেটের হাইপার টেক্সট মার্কআপ ল্যাংগোয়েজের ফাইল, '*.ps' মানে পোস্টস্ক্রিপ্ট ফাইল, '*.mp3' মানে এমপেগ-থ্রি মানে যার থেকে গান শুনি, '*.avi' মানে যার থেকে সিনেমা দেখি, ইত্যাদি। আচ্ছা, '*.h' বা '*.tar.bz2' বা '*.pdf' কী ধরনের ফাইল? মনে করার চেষ্টা করুন, প্রত্যেকটাই উল্লেখ করা আছে আগের দিনগুলোর লেখায়। আর, আন্দাজ করার চেষ্টা করুন তো, '*.conf' কী ধরনের ফাইল? একটা জিনিষ দেখুন, এই যে আমরা এক্সটেনশনগুলোকে উল্লেখ করছি '*.' দিয়ে, এর মানে ওই '*' অংশে যা খুশি আসতে পারে, তার পর একটা বিন্দু বা ডট '.' থাকবে, তারপর আসবে পদবিটা। এটা একটা রেগুলার এক্সপ্রেশন, মানে একটা ছক, শেল যা দিয়ে সবগুলো প্রবন্ধের ফাইলকে চিনেছিল পাঁচ নম্বর দিনে। পরে খুব দীর্ঘ আলোচনা আসবে 'যা খুশি আসতে পারে' এই পাইকিরি ছক বা রেগুলার এক্সপ্রেশন নিয়ে।

এই যে বললাম না, 'যা খুশি আসতে পারে' — এটা যে কী ভীষণ বিপজ্জনক, তা টের পাওয়ার জন্যে আপনি হাতে নাতে একটা জিনিষ করতে পারেন। একটা ফাইল তৈরি করুন। ফাইল তৈরি করার দুটো উপায় টাচ (touch) এবং ক্যাট (cat)। টাচ আসলে অন্য কাজের জন্যে তৈরি, তার একটা ব্যবহার এই ফাইল বানানো। ধরুন কমান্ড দিলেন '`touch essay`'। এর মানে 'essay' নামে একটা ফাইল তৈরি হয়ে যাবে আপনি যে ডিরেক্টরিতে আছেন সেখানে। কিন্তু এটা হবে শূন্য বা ব্ল্যাংক ফাইল। এত শূন্য, এত ব্ল্যাংক যে তাদের মধ্যে খড়ও ভরা নেই, বা ভাঙা কাঁচের উপর ইঁদুরের পায়ের শব্দের মত শুকনো গলায় তারা ফিসফিসও করেনা। শুধু একটা ফাইলনাম, এবং তার সঙ্গে একটা টাইমস্ট্যাম্প : কখন বানানো হল, আর আরো দু-চারটে তথ্য। একই ফাইল 'cat' দিয়ে বানাতে হলে দিতে হত '`cat > essay`'। এখন আমরা আর এন্টার মারার কথা উল্লেখ করছিনা, এত বার করেছি যে ওটা আমাদের মাথায় এন্টার করে গেছে বলেই মনে হয়। 'cat' দিয়ে কাজ করতে চাইলে কাজটা কিন্তু এখানেই শেষ নয়। ('cat' কী করে, মনে পড়ছে?) 'cat'-এর পরের চিহ্নটা দেখুন, রিডাইরেক্ট বা চালান করছে। কিন্তু কাকে? আপনাকে, মানে, এখন আপনি যা টাইপ করবেন। দেখুন, আপনার টাইপের জায়গা করার জন্যে 'cat' কমান্ড প্রম্পটটাকেই হাওয়া করে দিয়েছে, সেখানে জ্বলজ্বল করছে কার্সর, আপনার টাইপ করা অক্ষর যেখানে ফুটে উঠবে। প্রতি লাইনের শেষে একবার করে এন্টার, নতুন লাইন শুরু হবে, যখন মনে হবে, টাইপ করতে করতে আপনি নিজেই প্রায় টাইপরাইটার হয়ে গেছেন, তখন আর এন্টার নয়, একজিট করবেন, কন্ট্রোল-ডি মেরে। এতক্ষণ অদৃশ্য কমান্ড প্রম্পট ফের সগৌরবে ফিরে আসবে। এবং, এখন 'ls' দিলে দেখতে পাবেন সদ্য বানানো ফাইলটা। তার মধ্যে আপনার যে সূক্তিরত্নাবলী টাইপ করে রেখেছেন, সেটা যদি দেখতে চান, মারুন '`cat essay`'। স্ক্রিনে দেখিয়ে দেবে। এবার আর এটা শূন্য ফাইল নয়, আপনার টাইপ করা লেখা ভরা রয়েছে তাতে।

এবার, খ্যাঁচটা এইখানে শুরু, নামটা 'essay' না-দিয়ে দিন '–essay'। মজার কথা কী বলুন তো এই নামে ফাইল আপনি '`touch -essay`' দিয়ে তৈরিই করতে পারবেন না। যদিও, '`cat > -essay`' দিয়ে পারবেন না। 'ls' সেভাবেই দেখাবে এটাকে যেভাবে অন্য ফাইল দেখায়, আপনি যে নাম দিয়েছেন সেই নামেই। তাহলে? সমস্যাটা

কোথায়? এবার আপনি ফাইলটাকে '`rm -essay`' দিয়ে ওড়ানোর চেষ্টা করুন, বা '`mv -essay`' দিয়ে নাম বদলানোর, বা '`cp -essay essay`' দিয়ে এটাকে '`essay`' নামে কপি করার? পারলেন? এখন অব্দি যতটুকু গ্নু-লিনাক্স জানেন, তা দিয়ে চেষ্টা করে করে যদি এটা করে ফেলতে পারেন, তাহলে, এই মুহূর্তে একটা এপিঠ-ওপিঠ করান, নিজের নাম বদলে ফেলুন আইনস্টাইনে। আগে হাইফেনটা থাকলেই সেটাকে অপশান বলে মনে করে নিচ্ছে 'touch' বা 'rm' বা 'mv' বা 'cp'। অবজেক্ট না-ভেবে অপশান ভাবছে। কাজটা হবে কী করে? চেষ্টা করে দেখুন।

অনাইস্টাইনদের জন্যে পরিজ্ঞাতব্য তথ্য এটাই যে, এই ফাইলটাকে সাইজ করার কায়দা আমি এখন বলে দেবনা, ইচ্ছে করেই। যাতে গ্নু-লিনাক্স ডকুমেন্টেশন ঘাঁটার অভ্যেশটা হয় আপনার। যা গ্নু-লিনাক্স শেখার একটা প্রাথমিকতম শর্ত। যদি নিজে নেট থেকে টেক্সট নামিয়ে নেন বা বই কিনে নেন, তাহলে তো ভালোই। নইলে জিএলটির সিডিটার জন্যে মেল করবেন, একশো টাকা করে নিচ্ছি আমরা, ডাউনলোড আর সিডি পোড়ানোর খরচ বাবদ, তাতে গ্নু-লিনাক্স সংক্রান্ত ছহাজারের উপর ওয়েব পেজ এবং ছশোর মত পিডিএফ ফাইল রাখা আছে, যার গোটাটা শেখা হয়ে গেলে আপনি অনায়াসে আন্তর্জাতিক গ্নু-লিনাক্স খেতাব পরমলিনাক্সচক্রের জন্যে দরখাস্ত করতে পারেন। ওটা পেলে বিনা পয়সায় ইন্টারনেট করা যায়, আর সারা বছরই ঠ্যাং-ভাঙার ছুটি দেয়। যারা দূরে থাকেন, লিখবেন, জিএলটি-র সিডি পোস্টে পাঠিয়ে দেব।

উইন্ডোজে কাজ করার সময়, উইনডোজ এক্সপ্লোরার এক একটা বিশেষ ধরনের ফাইলের পাশে এক একটা বিশেষ ধরনের ছবি বা আইকন দেখায়। এদের যে কোনোটায় মাউস দিয়ে ক্লিক করলেই একটা বিশেষ প্রোগ্রাম তাদের খোলে। যেমন যেসব ফাইল '*.doc', তাদের যেকোনোটায় ক্লিক করলেই সেটা খোলে এমএস-ওয়ার্ড, '*.xls' হলে খোলে এমএস-একসেল, '*.txt' হলে নোটপ্যাড, বা খুব বড় সাইজ হলে, উইন্ডোজ ওয়ার্ডপ্যাড। এদের সিস্টেম চিনছে কিন্তু ফাইল এক্সটেনশন দিয়েই, মানে ডস ফাইলনাম কাঠামোর ওই '৮.৩'-এর পরের তিনটে অক্ষর দিয়ে। সিস্টেমের কী বিকট গ্যাঁড়ামো দেখুন, আপনি যখন ৪ অক্ষরের কোনো এক্সটেনশন ব্যবহার করছেন, 'glinux.html', তখনো সিস্টেম তাকে চিনছে ওই ৮.৩-এই। এমএস-ডস প্রম্পটে গিয়ে ডির (dir) কমান্ড মেরে দেখুন, সিস্টেম দেখাচ্ছে 'glinux~1.htm'। উইন্ডোজে 'dir' কমান্ডটা ফাইলের তালিকা দেখায়। এবার আর একটা কমান্ড দিন, '`type glinux~1.htm`'। 'type' হল গ্নু-লিনাক্সের 'cat' কমান্ডের মত, তার চেয়ে অনেক কম শক্তিশালী এমএস-উইন্ডোজের একটা প্রোগ্রাম, যা দিয়ে কমান্ড প্রম্পটে টেক্সট ফাইল দেখা যায়। দিব্য দেখুন, সোনামুখ করে 'type' ওয়েবপেজটার এইচটিএমএল কোডিং দেখিয়ে দিচ্ছে আপনাকে। আবার আপনি উইনডোজ এক্সপ্লোরারে যান, দেখবেন ওই ফাইলটার পাশে ওয়েবপেজের আইকন ফুটে আছে, এবং ক্লিক করলেই তাকে ইন্টারনেট এক্সপ্লোরার দিয়ে খুলছে। এই কাজের নাম আর দেখানোর নামের কথা তো আমরা আগেই বলেছি।

এবার মজাটা হল, আপনি কোনো একটা নামের এক্সটেনশন মানে পরের তিন অক্ষরের অংশটা বদলাতে যান, দেখবেন কী চেঁচায়, 'তুমি এসব করলে যদি আর কখনো ফাইলটা খুলতে না-পারো, আমি কিন্তু কিছু জানিনা', আসলে জানেনা ও নিজেই। ফাইলনামের পদবি বা এক্সটেনশন বদলে ক্লিক করে দেখুন, ফাইলটা ও খুলতেই পারবে না। এবং প্রত্যেকটা ব্যবহারকারীকেই নিজের মত নির্বুদ্ধি বলে ধরে নেয় উইন্ডোজ। গ্নু-লিনাক্স-এর মত নিরাপত্তা ব্যবস্থা এখানে না-থাকায়, কোনো ফাইল অন্য কারোর খোলাটা বন্ধ করতে চাইলে, আমি উইন্ডোজে কাজ করার সময় যেটা করে রাখতাম সেটা হল, কোনো '*.doc' বা '*.txt' ফাইলের এক্সটেনশনটা বদলে কোনো সিস্টেম ফাইল এক্সটেনশন দিয়ে রাখা যেমন '*.drv' বা '*.sys' ইত্যাদি। কেউ যদি ফাইলটা পায়, এবং উইন্ডোজ সিস্টেমের মতই পাঁঠা হয়, খুলতেই পারবে না। এবং সিস্টেমের গ্যাঁড়ামো এখানেই শেষ নয়, নিজের এই বোকামিটা আবার গোপন করারও চেষ্টা করে উইনডোজ। ডিফল্ট সেটিং-এ করা থাকে ফাইলনাম এক্সটেনশন না-দেখানোর ব্যবস্থা। উইনডোজ এক্সপ্লোরারের 'টুলস' মেনুর 'ফোল্ডার অপশান'-এ গিয়ে অফ করে দিতে হয় 'হাইড ফাইলনেম এক্সটেনশন'। তবে ফাইলনামের পদবিটা দেখায়। মানে লোকে দেখবে না, অথচ ও দেখবে, আর এক্সপ্লোরারে পাশে প্রোগ্রামের আইকনটা দেখাবে, যেন ওর কত বুদ্ধি, ম্যাজিক করে জেনে ফেলছে, কোন ফাইলটা কোন প্রোগ্রামের। এই ম্যাজিকটা সত্যিই 'ম্যাজিক' (magic) নামেই সত্যি গ্নু-লিনাক্স-এর বেলায়। এখানে তো কোনো এক্সটেনশন-এর চক্কর নেই, তাহলে কোনো ফাইলকে তার নিজের প্রোগ্রাম কী করে চিনে নেবে এক্স-উইনডোজের ব্রাউজার, বা, আমি নিজে যখন ব্যবহার করতে

চাইব কোনো ফাইল, সেটা কী কাঠামোর ফাইল কী করে বুঝব? এতেই ব্যবহার হয় 'magic'। আমরা সে কথায় পরে আসছি।

২.২।। ফাইলের গঠন

একটা গ্নু-লিনাক্স সিস্টেমে ফাইল হয় নানা রকমের। এর পরের সাবসেকশনে আমরা জানব সেসব। এখন আমরা ফাইলের গঠন নিয়ে কিছু কথা বলে নেব। এখানে ফাইলের গঠন বলতে আমরা রেগুলার বা সাধারণ ফাইলের গঠনই বোঝাচ্ছি। কারণ, অন্য অনেক ফাইলের গঠন কারনেলের দ্বারা নির্দিষ্ট, যেমন ডিভাইস ফাইল বা ডিরেক্টরি ফাইল। রেগুলার ফাইলের গঠনেও কিছু বৈচিত্র আছে বৈকি, সেই তফাতের কথায় আমরা পরে আসছি, যখন রেগুলার ফাইলের প্রকারভেদের আলোচনা আসবে। কিন্তু সেই গঠন ফারাকের পরও সব রেগুলার ফাইলের গঠনেরই একটা মৌলিক বৈশিষ্ট্য থাকে সব অপারেটিং সিস্টেমেই। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স বা উইনডোজে রেগুলার ফাইলের গঠনে একটা একদেশতা আছে (যদিও, চিরকাল সব অপারেটিং সিস্টেমের রেগুলার ফাইলের আকার বলতে যে এটাই ছিল তা নয়, এবং এখনো যে রেগুলার ফাইলের অন্য কোনো গঠন হয়না তা-ও নয়, কিন্তু সেগুলো আমাদের আলোচনায় আসবে না, বড্ড বেশি টেকনিকাল সেসব)। আমরা এখানে গ্নু-লিনাক্সের রেগুলার ফাইলের কিছু মৌলিক চরিত্র, বাইট রাখার কিছু সার্বজনীন কায়দা নিয়ে দু-একটা কথা বলে নেব।

যুগ থেকে যুগে, অপারেটিং সিস্টেম থেকে অপারেটিং সিস্টেমে আম রেগুলার ফাইলের গঠনের কায়দা এক নয়, একাধিক। তার মধ্যে একটা কায়দা হল যেখানে একটা ফাইল মানেই পরপর কিছু বাইটের সমষ্টি। যে বাইটগুলো আলাদা করে খুব ব্যক্তিত্ববান চরিত্রবান ইত্যাদি হতেই পারে, কিন্তু তাদের ব্যক্তিত্ব বা চরিত্র বিষয়ে অপারেটিং সিস্টেম কোনো খোঁজই রাখে না, শত চাইলেও কোনো ক্যারেকটার সার্টিফিকেট দিতে পারবে না। কোন বাইটে কী ধরনের জিনিষ আছে, কী আছে, তাতে অপারেটিং সিস্টেমের কিছু এসে যায়না। অপারেটিং সিস্টেম শুধু একটা খোঁজই রাখে, তাদের বাইটবানতা। মানে, তার ভাণ্ডারে কত বাইট আছে, কোন বাইট, কোন এলাকার কত নম্বর বাইট, ইত্যাদি। বাইটগুলো অর্থময় হয়ে ওঠে ব্যবহারকারীর স্তরে, বাইটগুলো থেকে অর্থ তৈরি হয় ইউজার লেভেল প্রোগ্রামগুলোর হাতে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্স, এবং উইনডোজও, ফাইলগুলোকে এভাবেই দেখে। অপারেটিং সিস্টেমের কাছে ফাইলগুলো বাইট-সমাহারের চেয়ে বেশি কিছু না-হওয়াটা কাজের পক্ষে সবচেয়ে সহজ এবং নমনীয় একটা অবস্থা তৈরি করে। ইউজার লেভেল প্রোগ্রামগুলো তাদের যেভাবে প্রয়োজন যা প্রয়োজন তাই রাখতে পারে ফাইলে, নিজেদের কাছে সবচেয়ে সুবিধেজনক রকমে যা খুশি নাম দিতে পারে ফাইলের। অপারেটিং সিস্টেম যেমন কোনো সাহায্য করেনা, তেমনি কোনো বাধাও দেয়না। কোনো ঝুঁকিপ্রবণ ব্যবহারকারী হ্যাকার যখন তার নিজের রকমে অপারেটিং সিস্টেম নিয়ে পরীক্ষা-নিরীক্ষা বা খেলাধুলো করতে চায়, সেখানে কিন্তু এটা একটা জরুরি ব্যাপার হয়ে ওঠে, তার স্বাধীনতা অনেক বেশি থাকে।

এছাড়া অন্য যে ধাঁচের আকারের বা ধর্মের ফাইল আগে হত, বা আজো কোনো বিশেষ বিশেষ কাজের জন্যে হয়, তাদের আমরা আর আলোচনায় আনছি না। কৌতূহল মেটানোর জন্যে বলে রাখি, এক ধরনের ফাইলে তথ্য থাকত বিশেষ একটা নথি বা রেকর্ডের ছকে। যেমন আশি-কলাম পাঞ্চড কার্ডের যুগে, অনেক মেইনফ্রেম অপারেটিং সিস্টেমেই ব্যবহার হত এই ফাইল। এই ফাইলে কোনো নতুন তথ্য লেখা মানে আর এক ব্লক রেকর্ড যোগ করা, যে ব্লকে আছে আশিটা চিহ্ন বা ক্যারেকটার। এই একটা রেকর্ড বা এক ব্লক নতুন তথ্য নতুন করে লেখা, বা পুরোনো একটা রেকর্ড উড়িয়ে সেখানে এই নতুন রেকর্ডটা গুঁজে দেওয়া — এটাই তথ্য যোগ করার একমাত্র তরিকা। তথ্য মানেই তার একক একটা রেকর্ড, এক ব্লক তথ্য, যাতে আছে আশিটা চিহ্ন। বা ১৩২ কলাম লাইন প্রিন্টারের জন্যে রেকর্ডের সাইজ হত ১৩২ টা ক্যারেকটারের। এই ধরনের ফাইল থেকে তথ্য পড়া মানে আবার একটা একটা করে রেকর্ড পড়া। এখন আর এই ফাইলের ধরনের ব্যবহার হয়না। কিন্তু অন্য আর এক রকমের ফাইল আছে যেখানে তথ্যটা থাকে গ্নু-লিনাক্স ফাইলের মত অসংগঠিত বিট সমাহারের আকারে নয়, সেখানে তথ্যের সবসময়ই একটা কাঠামো থাকে, একটা ট্রি বা ডালপালাসহ গাছের মত করে সাজানো থাকে তথ্যটা। এই ধরনের ফাইল এখনো ব্যবহার হয় কোনো কোনো বাণিজ্যিক ডেটাবেসের মেশিনে। এই ধরনের ফাইলে তথ্য থাকে নির্দিষ্ট কিছু 'কি' (key) বা বিশিষ্ট বিন্দুকে কেন্দ্র করে। তথ্য রাখা যোগ করা এবং খোঁজা সবটাই হয় এই 'key'-গুলোকে কেন্দ্র করে। যখন

সেই 'key' বা বিশিষ্ট বিন্দুর বিষয়ে কোনো তথ্য খোঁজা হয়, তখন জানার দরকার পড়েনা ফাইলের কোন জায়গায় তথ্যটা আছে। ওই বিশিষ্ট বিন্দুর বা 'key'-র সাপেক্ষে তথ্যটাকে অপারেটিং সিস্টেমই বার করে দেয়। অর্থাৎ এখানে ফাইলে তথ্য রাখা এবং পড়ার ব্যাপারে অপারেটিং সিস্টেম সরাসরি অংশগ্রহণ করে। কিন্তু এই দু ধরনের কোনো ফাইলই আমাদের প্রয়োজন পড়বে না, আমাদের কাছে ফাইল মানেই কিছু বাইটের সমষ্টি, যাদের বাইট কাঠামো ছাড়া আর কিছু অপারেটিং সিস্টেম জানেনা। সেসব জানে সেই প্রোগ্রাম যে ফাইল নিয়ে কাজ করে।

কিন্তু তাই যদি হয়, তাহলে, অপারেটিং সিস্টেম যদি নাই জানে কোন ফাইলে কী আছে, তাহলে এক নম্বর দিনে আমরা যে বিভিন্ন ধরনের ফাইলের পার্থক্যের কথা বললাম, কোনো ফাইল টেক্সটের, কোনো ফাইল কোডের, কোনো ফাইল প্রোগ্রামের, যাদের নাম লিখে এন্টার মারলেই চলতে শুরু করে, সেই তফাতটা ঘটছে কী করে? সিস্টেম তো আমাদের গোটা কাজটাকেই নিয়ন্ত্রণ করে। কোন ফাইল কী ধরনের, কী কাজ করে, সেটা না-বুঝলে সিস্টেম নিয়ন্ত্রণের কাজটা করবে কী করে?

২.৩।। রেগুলার ফাইল

একটা গ্নু-লিনাক্স সিস্টেমে তিন ধরনের ফাইল থাকে, রেগুলার বা সাধারণ ফাইল (Regular File), ডিরেক্টরি ফাইল (Directory File), এবং ডিভাইস ফাইল (Device File)। এইমাত্র ফাইলের তথ্যের কাঠামো নিয়ে আমরা যে আলোচনাটা করলাম সেটা এই রেগুলার বা অর্ডিনারি বা সাধারণ ফাইল। কারণ, ডিরেক্টরি ফাইল আর ডিভাইস ফাইলের চরিত্র একদম ভিন্ন ধাঁচের। তার কথায় আমরা পরে আসছি। রেগুলার বা সাধারণ ফাইলে থাকে বিভিন্ন ধরনের ব্যবহারকারীর স্তরের তথ্য। এমনকি যখন সিস্টেমের কোনো প্রসেস এই রকম কোনো ফাইল লেখে, সেই মুহূর্তে সেই প্রসেসও একজন ব্যবহারকারী। যেমন পরে আমরা যখন ইউজার আইডি বা ব্যবহারকারীর পরিচিতি কাকে বলে জানব, দেখব, একটা গ্নু-লিনাক্স অপারেটিং সিস্টেমে ব্যবহারকারীদের তালিকায় নাম থাকে 'man', 'games', 'mail', 'news', 'postfix' ইত্যাদি একাধিক সিস্টেম প্রসেসের। পাঁচ নম্বর দিনের আলোচনা থেকে /etc/passwd ফাইলটা মিলিয়ে দেখুন। একবার পাঁচ নম্বর দিনের ৯ নম্বর সেকশনে গিয়ে '/etc/passwd' ফাইলে একদম নিচের পাঁচজন চেনা ব্যবহারকারীর উপরে লেখা লাইনগুলোকে দেখুন তো আর একবার — লাইনের গোড়াতেই একটা করে প্রসেসের নাম, সেই প্রসেসটাই তখন একটা ইউজার বা ব্যবহারকারী।

এখানে একটা ছোট্ট খ্যাঁচ আছে, যার পুরো তাৎপর্যটা আমরা ফাইলসিস্টেমের বিশদ আলোচনা হওয়ার আগে অব্দি পুরোটা বুঝে উঠতে পারব না। এখন শুধু ব্যাপারটা মাথায় নিয়ে রাখুন। আমরা আগেই বলেছি, যে কোনো ইউজার না-হলেও, সুপারইউজার মানে রুট যে কোনো সময়েই কোনো ফাইলে কী আছে দেখতে পারে। অথচ কোনো ডিরেক্টরি ফাইল এমনকি রুট-ও পড়তে পারেনা। ডিরেক্টরির মধ্যে কী কী ফাইল বা সাবডিরেক্টরি আছে সেই তালিকাটা দেখতে পারে, কিন্তু খোদ ওই ডিরেক্টরি ফাইলটা পড়তে পায়না। যদিও সে যে কোনো সময় যে কোনো ডিরেক্টরি ওড়াতে পারে বা নাম বদলাতে পারে। ডিরেক্টরি ফাইলরা হল সিস্টেম ফাইল, যারা সিস্টেমের কাঠামো ধরে রাখে। তাদের সরাসরি পড়ার অধিকার আছে একমাত্র কারনেলের। কিন্তু এই ডিরেক্টরি ফাইলকে বাদ দিলে আর যে কোনো ফাইলই পড়ার অধিকার আছে রুটের। অন্য যে কোনো ইউজার কোনো ফাইল পড়তে পারে, যদি সেই ফাইল পড়ার অনুমতি থাকে তার। অনুমতির ব্যাপারটা নিয়ে আমাদের আলোচনা একটু বাদেই আসছে।

রেগুলার বা সাধারণ ফাইলগুলোয় সচরাচর ভরা থাকে অ্যাসকি টেক্সট বা বাইনারি তথ্য। সিস্টেমের কিছু এসে যায়না, ফাইলে বাইনারি আছে না অ্যাস্কি আছে না গুষ্টির পিন্ডি আছে, এসে যায় ইউজারের। সিস্টেম এই বাইনারি বা অ্যাসকির ভিতর কোনো তফাত করেনা, সবই তার কাছে রেগুলার ফাইল। শূন্য নম্বর দিনের আলোচনাগুলো মনে করুন, এই তথ্যের কাঠামো নিয়ে আমরা যেখানে কথা বলেছি। আর এক নম্বর দিনে আমরা বলে এসেছি বাইনারি, অ্যাস্কি ইত্যাদি বিভিন্ন ফাইলের কাজ কী? মনে করতে পারছেন? আগেই বলেছি, অ্যাসকি ফাইলে থাকে লাইন লাইন টেক্সট। এর সুবিধেটা এই যে খুব সহজে এদের স্ক্রিনে দেখা যায়, প্রিন্ট নেওয়া যায়, কোনো এডিটর দিয়ে এডিট করা যায়। সাদামাটা এডিটরদের প্রসঙ্গ মনে পড়ছে?

আর একবার মনে করিয়ে দিই — ওয়ার্ড প্রসেসর নয়, এডিটর। ওয়ার্ড প্রসেসর মানে ওয়ার্ডকে প্রসেস করে, প্রসেসিত ওয়ার্ডের নানা চিহ্ন গুঁজে দেয় নির্জলা র টেক্সট-এর মধ্যে। সিস্টেম তখন আর তাকে অ্যাসকি টেক্সট

হিশেবে কাজে লাগাতে পারেনা। '/etc/lilo.conf' ফাইলটা। একই রকম লাইনগুলো দেখতে বোর লাগছে বলে, এর প্রথম লাইনটাকে আপনি বোল্ড করলেন আপনার ওয়ার্ড প্রসেসরে, লাইনগুলো বাঁদিক থেকে আধ ইঞ্চি সরিয়ে এনে ইনডেন্ট করলেন। ফাইলের এই ফেস-লিফট শেষ করলেন, ফাইলটায় আপনার ওয়ার্ড প্রসেসর এই সব করার আদেশগুলো নানা বিচিত্র রকমে এবং চিহ্নে গুঁজে দিল। এবার, সিস্টেম যখন, বুট করার সময়ে, '/etc/lilo.conf' পড়তে গেল, কোথায় কোন কারনেল আছে জানার জন্যে তাকে তো পড়তেই হবে, আগেই বলেছি। সে তখন ফাইলের মধ্যে গিয়ে পেল এইসব দেবশিশুদের পদচিহ্ন, কিন্তু সিস্টেম তার বাপের জন্মে, ইউনিক্স মাল্টিক্স এফএমএস, কোনোদিন এসবের মানে জানেনা, পড়তে পারল না। এবং ভারি হিংসুটে সে, কর্নিশ অগরের কাছে যাওয়ার আগের অবস্থাটা, সিস্টেমের দৈত্য কিছুতেই বুট করতে রাজি হলনা। স্ক্রিনে ফুটে উঠল 'বুট ফেইলিওর . . . কারনেল প্যানিক'। প্যানিক শুধু কারনেলের? তখন আপনার প্রাণও ইঁক ইঁক করছে। ভাবছেন, কী দরকার ছিল এসব অক্ষরের শব্দের প্যারার কালচারাল সৌন্দর্যবোধ ওই কেলে চাঁড়াল দৈত্যকে শেখাতে যাওয়ার? এর চেয়ে ভালো কোনো ম্যাটার-অফ-ফ্যাক্ট এডিটর, যে নিখাদ টেক্সট আর লাইন-ভাঙার চিহ্ন ছাড়া আর কিছু গুঁজবে না ফাইলের মধ্যে। যেমন ইম্যাক্স (emacs), ভিম (vim), জো (joe), গেডিট (gedit), কেডিট (kedit), ইত্যাদি। ওপেন অফিস রাইটার (Ooo-writer), অ্যাবি-ওয়ার্ড (abiword), কে-অফিস (koffice) ইত্যাদি ওয়ার্ড প্রসেসর দিয়ে হবেনা।

এই অ্যাস্কি ফাইলগুলোকে শুধু যে এডিট করা যায় তাই নয়, যদি অনেক প্রোগ্রামই ইনপুট আউটপুট হিশেবে এই অ্যাস্কি ফাইল ব্যবহার করে, তখন এদের একটার আউটপুটকে অন্যটার ইনপুটে জুতে দেওয়া যায়, শেলের পাইপলাইন দিয়ে, পথনির্দেশ করে। অডিয়ো সিডি নিয়ে আমাদের তত্ত্বতালাশ ফাইলে তুলে রাখার সময়, একটু আগে দেখেননি, '`man -k CD | grep 'audio'> audio.cd.program.text`' কমান্ডে? 'man' প্রথমে সমস্ত ম্যানুয়াল পাতা ঘেঁটে দেখল, কোথায় কোথায় সে পাচ্ছে 'CD' শব্দটা। '`man -k CD`' কমান্ডের ৩৮ লাইন অ্যাস্কি আউটপুট চলে গেল 'grep' কমান্ডের কাছে ইনপুট হয়ে। '`grep 'audio'`'। 'grep' এবার তার সামনে পেল ওই আটত্রিশ লাইন ইনপুট, তাদের মধ্যে সে খুঁজে দেখল কোনো লাইনে 'audio' শব্দটা আছে কিনা। এবার 'grep' কমান্ডের যে ছ লাইন অ্যাস্কি আউটপুট, যেটা আগে আমরা দেখছিলাম স্ক্রিনে, সেটা রিডাইরেক্ট বা চালান হয়ে গেল একটা অ্যাস্কি টেক্সট ফাইলে, যার নাম 'audio.cd.program.text'।

সাধারণ বা রেগুলার ফাইলদের মধ্যে সবচেয়ে সংখ্যাগরিষ্ঠ এই অ্যাস্কি টেক্সট ফাইল। শূন্য নম্বর দিনে আমরা যে চিহ্ন বা ক্যারেকটারের কথা বলেছিলাম, যাদের স্ক্রিনে দেখা যায়, বা ছাপা যায়, তাদের দিয়েই ভরা থাকে এই ফাইলগুলো। 'ls', 'who', 'tar', 'less', 'wc', 'man' ইত্যাদি যে কমান্ডগুলো আমরা ব্যবহার করছিলাম তারা কিন্তু এই অ্যাস্কি ফাইল নয়। বা সি কোড লেখা কোনো ফাইল আপনি কম্পাইল করার পর যে একজিকিউটেবল বা চালানো-যায়-এমন বাইনারি ফাইল পাবেন, তারাও অ্যাস্কি টেক্সট ফাইল নয়। টেক্সট ফাইলের গোটা টেক্সটা ছোট ছোট দলে ভাঙা থাকে, এক একটা লাইনে। এক এক লাইন মানে এক এক দল চিহ্ন। শূন্য নম্বর দিনের শেষতম আলোচনায় লাইনফিড বা 'LF' আর ক্যারেজ-রিটার্ন বা 'CR' নিয়ে আলোচনাটা একবার উল্টে নিন। তখন যদি অনেকটা ট্যান গিয়ে থাকে, এখন দেখুন, তার চেয়ে বেশি বুঝবেন। লাইনবদলের চিহ্নটার বিশেষত্বটা দেখুন, চিহ্নটাকে আপনি দেখতে পাননা, না স্ক্রিনে না প্রিন্টে, দেখেন তার কাজের নমুনাটাকে। যেই একটা লাইন ভেঙে আর একটা নতুন লাইন শুরু হয়। টেক্সট লেখার সময় আপনি যেই এন্টার মারেন, অমনি সেখানে একটা লাইনফিড ঢুকে পড়ে। পরিচিত ওয়ার্ড-প্রসেসরগুলোয়, যখন আপনি অপশানে দেন 'শো নন-প্রিন্টিং ক্যারেকটারস', অদৃশ্য চিহ্নগুলো দেখাও, তখন অন্য অনেক বাড়তি চিহ্নের সঙ্গে এই লাইনফিড-কেও দেখায়, সচরাচর এর চিহ্নটা থাকে '¶'।

অ্যাসকি ফাইল ছাড়া অন্য ফাইল হল বাইনারি ফাইল, মানে, এক কথায় না-অ্যাস্কি রেগুলার ফাইল। স্ক্রিনে ক্যাট করলে বা প্রিন্ট নিলে ঠিক সেই জাতের জঞ্জাল তৈরি হয়, যার সাথে আমরা পরিচিত হয়েছিলাম, শূন্য নম্বর দিনে, ভিশুয়াল চ্যাংড়ামোয়। আমাদের কাছে মনে হয় বিশুদ্ধ এবং পরিণামহীন পরিচ্ছন্ন জঞ্জাল, কিন্তু তার মধ্যেও অনেক লুপ্ত অ্যাটলান্টিস রয়ে যায়, বুঝো লোক যে জানো সন্ধান। 'ls' একটা প্রোগ্রাম, মানে একটা বাইনারি ফাইল। প্রতিবার যখন 'ls' মারি, শেল তাকে খুঁজে বার করে '/bin/ls' থেকে, সিস্টেম তাকে পড়ে ফেলে এবং এক্সিকিউট করে। মেশিন পড়ে বুঝবে এমন আদেশমালা ভরা থাকে তাতে। সি-তে লেখা কিছু কোডকে কম্পাইল করে যে

মেশিনবোধ্য আদেশমালা পাওয়া গেছিল। 'ls' নামের প্রোগ্রাম ফাইল কী বস্তু সে প্রতিবার পড়ে, এই কৌতূহল থেকে, ফাইলটাকে টেক্সট এডিটর দিয়ে খুলেছিলাম। তার একদম গোড়াতেই, অন্তত দুটো মানববোধ্য চেনা শব্দ খুঁজে পেয়েছিলাম, 'gnu' আর '/lib/ld-linux.so.2'। দ্বিতীয়টা ঠিক কী ধরনের চীজ কোনো আন্দাজ করতে পারছেন? কোনো আলগা আন্দাজ? করার চেষ্টা করুন। কম পড়ো, বেশি আন্দাজ করো, আগেই তো বলেছি, এটাই আমাদের মোটো। 'ls' নামের বাইনারি ফাইলে যে এলোপাথাড়ি জঞ্জালগুলো দেখছি, অবভিয়াসলি তার মধ্যেও একটা কাঠামো আছে, মেশিন ভাষার কাঠামো। যা থেকে প্রোগ্রামটা কাজ করে। যদিও গোটাটাই হল পরপর কিছু বাইটের সমাহার, কিন্তু এর মধ্যেই একটা কিছু থাকে যাতে কোনো ফাইলকে সিস্টেম একজিকিউট করে, চালায়, অন্য প্রোগ্রামকে চালায় না, তার মানে সিস্টেম একজিকিউটেবল ফাইলকে একজিকিউটেবল বলে চিনতে পারে। কী করে?

একটা একজিকিউটেবল বাইনারি ফাইলের মূলত পাঁচটা খণ্ড থাকে — হেডার, টেক্সট, ডেটা, রিলোকেশন বিট, এবং সিম্বল টেবল। আর এই হেডার শুরু হয় সেই ম্যাজিক নাম্বার দিয়ে, যা থেকে সিস্টেম এই ফাইলটাকে একটা একজিকিউটেবল ফাইল বলে চিনতে পারে। এই ম্যাজিক নাম্বার দিয়ে বিভিন্ন ধরনের ফাইলকে চেনে সিস্টেম। গ্নু-লিনাক্সে একটা কমান্ড আছে, ফাইল (file), যা ফাইলের মধ্যেকার হেডারের ওই ম্যাজিক নাম্বার আর অন্য প্রয়োজনীয় তথ্য পড়ে আপনাকে জানিয়ে দেয় ফাইলটার সম্পর্কে আপনি যা জানতে চান। সিস্টেমও ওই একই ভাবে জানে ফাইলটার বিষয়ে। কোনো ফাইল এক্সটেনশন জাতীয় কায়দা বা তাকে জানানোর জন্যে আরো আরো কায়দা, এত কায়দাবিলাসী নয় গ্নু-লিনাক্স। 'file' কমান্ডটা ব্যবহার করে দেখুন। একবার দিন, '`file /bin/ls`' আর একবার, প্রবন্ধের ডিরেক্টরিতে দাঁড়িয়ে দিন, '`file essay.1.text`'। দেখুন তো কী দেখাচ্ছে। অন্য ফাইলকেও দেখুন। ফাইলের কোনো ব্যবহারগত বৈশিষ্ট্য বা অ্যাট্রিবিউট, যেমন কার মালিকানা এই ফাইলের উপর, কে কে একে পড়তে বদলাতে বা চালাতে পারবে, এই জাতীয় কোনো তথ্যই গ্নু-লিনাক্স সিস্টেমে কোনো ফাইলের মধ্যে থাকেনা। এমনকি, হায়, ফাইল নিজে নিজে নিজের নামটা অব্দি জানেনা, সেটাও ফাইলের মধ্যে থাকেনা। আচ্ছা মন দিয়ে যেমন সিস্টেম পড়তে বলছি, এই পাঠমালাটাও পড়ছেন তো। আপনি কি খেয়াল করেছেন, '/bin/ls' ফাইলটার উপর 'file' কমান্ড মারার সময় আমি ডিরেক্টরির কথা কিছু বলিনি। কিন্তু 'essay.1.text' ফাইলটার বেলায় বলেছি। আপনার কি কোনো কৌতূহল জেগেছিল, কেন এটা আমি করলাম? কেন?

২.৪।। ডিরেক্টরি ফাইল

ডিরেক্টরি ফাইলে কোনো বাইরের তথ্য থাকেনা, শুধু তার মধ্যে যে ফাইল বা সাবডিরেক্টরিগুলো আছে, তাদের কিছু খুঁটিনাটি বিবরণ থাকে। গ্নু-লিনাক্স ফাইলসিস্টেম সাজানো থাকে এরকম ডিরেক্টরির মধ্যে ডিরেক্টরির মধ্যে ডিরেক্টরি দিয়ে। নিজের মালিকানার মধ্যে যেসব ডিরেক্টরি আছে, ঠিক এমনি ফাইলের মতো, যখন খুশি ওড়াতে পারে, নড়াতে পারে, নাম-বদলাতে পারে কোনো ইউজার। আবার নিজের মালিকানার এলাকায় যেখানে খুশি যেমন খুশি ডিরেক্টরি বানাতে পারে। ডিরেক্টরি ওড়ানোর কমান্ড হল 'rmdir', আর বানানোর কমান্ড 'mkdir'। কিন্তু, একটা ডিরেক্টরিকে 'rmdir' দিয়ে ওড়ানো যায় শুধু একটা শূন্যগর্ভ ডিরেক্টরিকে, মানে যখন তার পেটে আর কোনো ডিরেক্টরি বা সাবডিরেক্টরি বা ফাইল নেই। সেই অবস্থায় কমান্ড হল '`rm -fr`'। তবে এই 'rm -fr' কমান্ডটা একটা বিষাক্ত জিনিষ। আপনি রুট (root) মানে সুপারইউজার হয়ে, যখন আপনার সব জায়গায় সমস্ত কাজের অনুমতি আছে, যদি কমান্ড দেন '`rm -fr /`', অশ্লেষা মঘা ত্র্যহস্পর্শ একত্রে ঘটে যাবে (রুট-ডিরেক্টরি কাকে বলে সেটা আপনার কাছে এখনো স্পষ্ট নয়, চিন্তা নেই)। আপনার গ্নু-লিনাক্স সিস্টেম তার যাবতীয় ফাইল ওড়াতে শুরু করে দেবে। একদম গোড়া থেকে, রুট ডিরেক্টরি মানে '/' থেকে শুরু করে, প্রতিটি ডিরেক্টরির প্রতিটি সাবডিরেক্টরির প্রতিটি ফাইলকে উড়িয়ে যখন থামবে তখন আপনার সিস্টেম নিছক একটা শূন্য, হার্ডডিস্কের আর সার্কিটের ভৌত অবয়ব। নতুন করে বুট করলে সিস্টেম বুটই করবেনা, দেখাবে 'সিস্টেম রিকোয়ার্ড'। আমি সত্যিই করে দেখেছিলাম, সমস্ত প্রয়োজনীয় ফাইলের ব্যাকআপ সিডিতে নিয়ে। আর কোনোদিন কোনোকিছু ফেরত পাওয়া যায়না। এই শূন্য করে দেওয়াটা হয়, টেকনিকাল ভাষায় 'মিলিটারি ওয়াইপিং'। যদি প্রতিটি সফটওয়ার প্রতিটি ডেটাফাইল নতুন করে পয়দা করার উপায় না-থাকে, সাবধান, বিষ নিয়ে খেলা করবেন না — শূন্য করে ভরে দেওয়া যাহার খেলা তারই সাজে।

ডিরেক্টরি বানাতে হয়, আমরা দেখেছি, যখন কিছু নির্দিষ্ট ধরনের ফাইলকে একই সঙ্গে রাখার দরকার পড়ে। ফাইলগুলোকে সংগঠিত করার দরকার পড়ে। একটা ডিরেক্টরিতে একই নামে একটার বেশি ফাইল থাকতে পারেনা, কিন্তু, আপনি যখন প্রবন্ধের সংকলনের পর সংকলন লিখে চলেছেন, নতুন নতুন ডিরেক্টরি, 'book.one', 'book.two', 'book.three' ... , তার প্রতিটাতেই একটা করে 'essay.1.text', 'essay.2.text', 'essay.3.text' ... ইত্যাদি থাকতেই পারে। নিজের পেটের ভিতর ফাইল আর সাবডিরেক্টরিদের নিয়ে প্রতিটি ডিরেক্টরি স্বয়ংসম্পূর্ণ। প্রতিটি ডিরেক্টরির জন্যে থাকে একটা করে ডিরেক্টরি ফাইল, হার্ডডিস্কের ভৌত জ্যামিতিতে এই ডিরেক্টরি ফাইলের গঠনের ব্যাকরণ পরে আমরা আরো একটু ভালো করে বুঝতে পারব, ফাইলসিস্টেমের আলোচনায়। প্রতিটি ডিরেক্টরি ফাইলে, ডিরেক্টরির মধ্যের প্রতিটি ফাইলের জন্যে রাখা থাকে দুটি করে তথ্য রাখার জায়গা বা ফিল্ড। একটা ফিল্ড হল ফাইলটার নাম, অন্যটা ফাইলটার আইনোড নম্বর। আইনোড ব্যাপারটাও এখনি খুব স্পষ্ট হবেনা, তাতে চিন্তা নেই, হয়ে যাবে। একটা ডিরেক্টরিতে যদি দশটা ফাইল থাকে, তার জন্যে ডিরেক্টরি ফাইলে থাকবে দশটা এন্ট্রি, প্রতিটিতে ওইরকম দুটো করে ফিল্ড। এই ডিরেক্টরি ফাইলেই লেখা থাকে তার ভিতরকার ফাইলগুলোর নাম, যে নাম আপনি দিয়েছেন, বা কোনো প্রোগ্রাম দিয়েছে। কিন্তু আগেই বলেছি, সরাসরি ডিরেক্টরি ফাইলে লেখার অধিকার কারনেল ছাড়া আর কারোর নেই, এমনকি রুটেরও না। কেউ কোনো ফাইল বা ডিরেক্টরি বদলানো মাত্র, বা কোনো ফাইলের বা ডিরেক্টরির উপর পড়ার বা বদলানোর বা চালানোর এই তিন ধরনের অনুমতির কোনোটায় কোনো বদল ঘটানো মাত্র, সেই ফাইল বা ডিরেক্টরি যে ডিরেক্টরির মধ্যে আছে, সেই ডিরেক্টরির ডিরেক্টরি ফাইলে সেই বদলটা লিখে দেয় কারনেল। এই অনুমতিগুলো নিয়ে আরো ভালো করে আলোচনায় আসছি আমরা ফাইল বৈশিষ্ট্য বা অ্যাট্রিবিউট-এর আলোচনায়।

২.৫।। ডিভাইস ফাইল

আমরা দু-নম্বর দিনের ৬ নম্বর সেকশনে ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল ডিভাইসের কথা বলেছিলাম, সেই দুরকম ডিভাইসের জন্যে গ্নু-লিনাক্সে থাকে দু-রকমের ফাইল — ব্লক স্পেশাল এবং ক্যারেকটার স্পেশাল ডিভাইস ফাইল। আগেই বলেছি, গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সে সবকিছুই এক একটা ফাইল। হার্ডডিস্ক, প্রিন্টার, সিডি-ড্রাইভ, কনসোল, মোডেম — সবকিছুই এক একটা ফাইল। ক্যারেকটার স্পেশাল ফাইল বরাদ্দ থাকে ক্যারেকটার ডিভাইসের জন্যে — অক্ষর বা চিহ্ন বা ক্যারেকটারের ইনপুট এবং আউটপুট করাই যার কাজ। আমি যখন আমার কম্পিউটার থেকে মেল করছি কোনো ইমেল আইডিতে, আমার সিস্টেম কিছু বিশেষ চিহ্ন লিখছে একটা বিশেষ ফাইলে, আউটপুট করছে। যে ফাইলের নাম '/dev/modem'। '/dev/modem' ফাইলের ধর্মই হল তার শরীরে যে চিহ্নপ্রবাহই আসুক, তাকে বিদ্যুৎপথ বেয়ে উদ্দীষ্ট কোনো কম্পিউটারে পাঠিয়ে দেওয়া, যে কম্পিউটারে আমার উদ্দীষ্ট ইমেল আইডি-টা আছে, বা যে কম্পিউটার আমার চিঠির চিহ্নপ্রবাহকে সেই দিশায় পাঠিয়ে দিতে পারে। মানে, আমার চিঠিটা মোডেম নেটে পাঠিয়ে দিচ্ছে। আবার যখন আমি নেট থেকে একটা ওয়েবপেজ পড়তে চাইছি, তখন আবার ব্রাউজারে দেওয়া ঠিকানা থেকে চিহ্নপ্রবাহর আমদানি হতে শুরু করছে আমার মেশিনে। আমার মেশিন নেট থেকে মোডেমে আসা তথ্য পড়ে ফেলছে ওই একই '/dev/modem' ফাইল থেকেই, এবার কিছু চিহ্ন বা অক্ষর বা ক্যারেকটারের ইনপুট। ব্লক স্পেশাল ফাইল থাকে হার্ডডিস্ক ইত্যাদি ব্লক ডিভাইসের জন্যে। নির্দিষ্ট মাপের ব্লক থাকে একক হিশেবে। সেই ব্লকের মাপে মাপে এখানে তথ্য ইনপুট আউটপুট করা হয়, ইনপুট মানে যখন হার্ডডিস্কে ফাইল লিখছি, আর আউটপুট যখন হার্ডডিস্ক থেকে ফাইল পড়ছি। যখন আমি আমার দ্বিতীয় হার্ডডিস্কের তিন নম্বর পার্টিশনে কোনো তথ্য রাখছি, বা সেখান থেকে পড়ছি, সেই কাজটা ঘটছে '/dev/hdb3' নামের একটা ব্লক ডিভাইস ফাইল দিয়ে। এই ব্লক ডিভাইস এবং ক্যারেকটার ডিভাইস ফাইল এই দুইরকম ফাইলকে মিলিয়ে আমরা এক কথায় বলি ডিভাইস ফাইল।

এই যে লিখছি, প্রতিটি ভৌত উপাদানই এক একটা ডিভাইস ফাইল, হার্ডডিস্ক, সিডি-ড্রাইভ, ফ্লপি-ড্রাইভ, কনসোল বা টার্মিনাল, প্রিন্টার, মোডেম সমস্ত কিছু — প্রথম প্রথম এটা একটু ঘোলাটে লাগতে পারে, কিন্তু পরে দেখবেন, বেড়ে সুবিধে আছে এর। যেসব কমান্ড দিয়ে একটা রেগুলার ফাইলকে কব্জা করা যায়, একই ভাবে সেগুলো কাজ করে ডিভাইস ফাইলের উপরেও। যেমন আমাদের চেনা কমান্ড, 'cat' কনক্যাটেনেট করে, মানে, স্ক্রিনের ডিসপ্লেতে

পেতে দেয় একটা ফাইলকে। শুধু স্ক্রিনে না, যেখানে আপনি চাইবেন, যে কোনো ফাইলে। ইতিমধ্যেই রয়েছে এমন কোনো ফাইল হতে পারে, আবার নতুন করে বানিয়ে নেওয়া ফাইলও হতে পারে। আপনি 'cat essay' কমান্ড দিলে, 'essay' ফাইলটাকে স্ক্রিনে পড়ার জন্যে পেতে দেয় 'cat'। মজার কথা হল, এই একই ভাবে আপনি যে কোনো ডিভাইস ফাইলকেও 'cat' করতে পারেন।

মাউস ডিভাইস বা '/dev/mouse' ফাইলটাকে 'cat' করতে পারেন 'cat /dev/mouse' কমান্ড দিয়ে। এন্টার মারার পর দেখুন, কমান্ড প্রম্পট ফেরত আসেনি, আদেশ পালনের কাজটা চলছে। এবার মাউস নেড়ে দেখুন, প্রফেসর হিজিবিজবিজের সাম্প্রতিক কিছু যুগান্তকারী গবেষণা আপনার সামনে ভেসে উঠবে। 'cat' করতে পারেন সরাসরি আপনার ভৌত হার্ডডিস্কের একটা ভৌত পার্টিশনকে। করে দেখুন, 'cat /dev/hda1'। কত বুদ্ধিদীপ্ত প্রবন্ধে ভরে রেখেছেন হার্ডডিস্ক, তারা আসলে এই? একটা মজার খেলা হয়, কারনেলের কণ্ঠস্বর শোনা। '/dev/dsp' হল আপনার মেশিনের সাউন্ডকার্ডের ডিভাইস ফাইল। 'ডিএসপি' মানে ডিজিটাল সিগনাল প্রসেসর (**D**igital-**S**ignal-**P**rocessor) থেকে। আর পাঁচ নম্বর দিনের 'lilo'-র আলোচনায় বলেছি, '/boot/vmlinuz' হল কারনেল। সাউন্ডকার্ড কনফিগারেশন যদি ঠিক থাকে তাহলে 'cat /boot/vmlinuz > /dev/dsp' কমান্ড দিয়ে আপনার সিস্টেমের কারনেলকে সশব্দ সগর্জন টের পাবেন। খেয়াল করুন, মাউস আর হার্ডডিস্কের বেলায় আমরা 'cat' করেছি স্ক্রিনে, আর কারনেলের বেলায় করেছি '/dev/dsp' ফাইলে। বললাম না, যেখানে খুশি করা যায়।

ডিভাইস ফাইলটা এমনিতে কিন্তু ফাঁকা। আপনি তো ফাইলের সাইজ দেখতে জানেন। 'ls –sh /dev/hda1' বা 'ls –sh /dev/mouse' কমান্ড দিয়ে দেখুন, দুটোই দেবে শূন্য। অথচ 'cat' দিলে দেখাচ্ছে। ঠিক এটাই মজা ডিভাইস ফাইলের। ডিভাইস ফাইলরা একদম শেখভের ডার্লিং-এর মত। যখন কোনো নাটকের ম্যানেজারকে ভালোবাসে, তখন তার ভারি দুঃখ, মানুষ কেন ভালো নাটক দেখতে শিখছে না। আবার কোনো কাঠব্যবসায়ীকে ভালোবাসার পর, গির্জে যাওয়ার পথে কিছুতেই রাগ সামলাতে পারেনা, এত যে দেরিতে বর্ষা নামল, ওক কাঠের দর এবার কত পার্সেন্ট বাড়বে বুঝতে পারছো, এদিকে মানুষ শুধু বাজে খরচ করছে থিয়েটার দেখে। ডিভাইস ফাইলের নিজের শরীরে কোনো তথ্যই থাকেনা, সব তথ্যই তার প্রিয়তমের। সব তথ্যই আসে তার দোসর ভৌত ডিভাইসের কাছ থেকে। তেমনি, নিজে কিছু ধরেও রাখেনা, সবই পাঠিয়ে দেয় ওই ভৌত উপাদানকে। আপনি কোনো ফাইল পাঠালেন 'cp essay /dev/lpt1' করে, মানে, আপনার 'essay' নামের ফাইলটা '/dev' ডিরেক্টরির 'lpt1' নামের ডিভাইস ফাইলে কপি হতে, ঠিক যেভাবে আপনি একটা রেগুলার ফাইলকে অন্য একটা নামে অন্য কোনো জায়গায় কপি করেন, ওমা, দেখলেন পুরো শ্রমজীবী কোলাহল নিয়ে আপনার পাঠানো ফাইলটা প্রিন্ট হতে শুরু করেছে 'lpt1' নামের প্রিন্টার-পোর্টে লাগানো ডট-ম্যাট্রিক্স প্রিন্টার থেকে। শ্রমজীবী ঘটাং-এর জায়গায় ওটা এলিট গুজগুজ-ও হতে পারে, যদি ডেস্কজেট বা লেজারজেট হয়। এবং আপনার কমান্ড প্রম্পট সঙ্গে সঙ্গেই ফেরত এসেছে, মানে, সিস্টেমের কপি করা হয়ে গেছে, এখন যা ঘটার সব প্রিন্টারে ঘটছে। বা, ফ্লপির ডিভাইস ফাইলে ফাইল কপি করছেন মানে ফ্লপিতে রাখছেন ফাইলটা। আপনি ওই কপি (cp) বা মুভ (mv) কমান্ড দিয়েই খালাস, পরের টুকু নিয়ে মাথা ঘামাচ্ছে কারলেন।

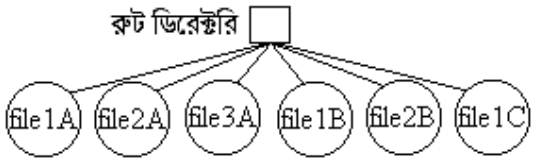
২.৬।। এক খাবলা পুনশ্চ

আপাতত আমাদের ফাইল কাকে বলে জানলাম। এবার আমরা যাব ডিরেক্টরি সিস্টেম নিয়ে আলোচনায়। সেখানে আমরা পথনির্দেশের খুঁটিনাটি জানব, কী ভাবে একটা ফাইলের ঠিকানা দেওয়া হয় একটা গ্নু-লিনাক্স সিস্টেমে। তার আগে, দুটো কথা বলে নেওয়ার আছে। ওই '–essay' নামের ফাইলটাকে কিছু করার উপায় না-দেওয়ায় আমার দুজন প্রাথমিক পাঠক খুব চেঁচানোয় উত্তরটা দিয়েই রাখছি। এই ধরনের যে কোনো ফাইলকে কব্জা করার উপায় হল মধ্যে একটা '--' দিয়ে নেওয়া। মধ্যে দেওয়া ওই '--' অংশটা শেলকে বলবে পরের অংশটাকে আক্ষরিকভাবেই একটা গোটা নাম হিশেবে পড়তে। কেন, সেটা আমরা জানতে পারব ব্যাশ শেল নিয়ে বিশদ আলোচনায়। এবার, যে কোনো কমান্ডেই মধ্যে এই '--' ডান্ডাটা দেওয়া থাকলে আর কোনো ঝামেলা হবেনা। যেকোনো কমান্ডই কাজ করবে। তবে, সবচেয়ে সহজ হল 'mv -- -essay essay' করে ফাইলটার নামের হাইফেনটা উড়িয়ে দেওয়া। এখানে বলে রাখি, শুধু হাইফেন না, এই রকম ঝামেলা হতে পারে '$', '`', '?' বা '*' চিহ্ন নিয়েও। এই চিহ্নগুলোর

শেলে আলাদা অর্থ আছে। তাই, এই চিহ্নগুলো ফাইলনামে ব্যবহার করলে শেল কুপথে চালিত হওয়ার সম্ভাবনা থাকে। নইলে বারণ নেই এর কোনোটা ব্যবহারেই। যে কোনো ল্যাটিন মানে ইংরিজি অক্ষর, তার বড় বা ছোট হাতে, 'A-Z' বা 'a-z', যে কোনো অঙ্ক, '0-9', পিরিয়ড মানে ডট বা বিন্দু বা '.', আন্ডারস্কোর মানে '_', এর কোনোটাতেই কোনো সমস্যা নেই। পাঠমালার শেষ মানে দশ নম্বর দিনে এদের নিয়ে বড় আলোচনা আসবে।

৩ ।। এক স্তরের সরলতম ডিরেক্টরি সিস্টেম

এক একটা সিস্টেমে মোট থাকে হাজার হাজার লাখ লাখ ফাইল। আমার মেশিনের দুটো হার্ড ডিস্কে, মোট সাতটা পার্টিশন, তার মধ্যে একটা সোয়াপ ফাইল, মোট আশি জিবিতে, ফাইলের সংখ্যা ৫৫৬৯১৩। একটা ব্যক্তিগত সিস্টেমেই এই সাড়ে পাঁচ লাখেরও বেশি ফাইল, এবার একটা অফিস বা ইনস্টিটিউট-এর মোট ফাইলের সংখ্যা ভাবুন। এই অগণিত ফাইল থাকার জন্যে, অবশ্যই একটা ডিরেক্টরি কাঠামো দরকার পড়ে। চার নম্বর দিনের সেকশন ১৩-য় আমাদের আলোচনায় ছিল, ১৯৬০-এর দশকের পৃথিবীর সবচেয়ে শক্তিশালী মেইনফ্রেম কম্পিউটার সিডিসি ৬৬০০-র কথা। যাকে 'সুপারকম্পিউটার' বলে ডাকা হয়েছিল। সেখানে যাবতীয় ব্যবহারকারীর জন্যে ডিরেক্টরি ছিল সাকুল্যে একটা। প্রতিটি ইউজারের প্রতিটি ফাইলই থাকত সেই একমেবাদ্বিতীয়ম ডিরেক্টরিতে। ভাবুন তো সেইভাবে যদি রাখতে হত একটা ব্যক্তিগত সিস্টেমেরই ছয় লাখ ফাইল, তাহলে লোকে কাজ করবে কী, ফাইলের আলাদা আলাদা নাম আবিষ্কার করতে গিয়েই তো কাঁধ ঝুলে যাবে। আর, কারুর মেশিন ঝুলিয়ে দেবার জন্যে একবার 'ls' মারাই যথেষ্ট হত। আসলে সিডিসি ৬৬০০-র পরের চল্লিশটা বছরে পৃথিবী, মানুষ, তার কাজ, কাজের হাতিয়ার সবই বদলে গেছে বিরাট রকমে। তাই এখন ডিরেক্টরি কাঠামোও দরকার পড়ে অনেক বিশদ জটিল এবং স্তরবিন্যস্ত।



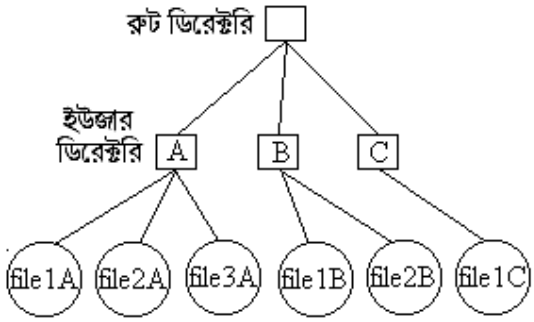
সিডিসি ৬৬০০ গোছের মেশিনে, এবং এর পরেও কিছুদিন অব্দি, ব্যবহৃত হত, এই সরলতম ডিরেক্টরি কাঠামো, যাতে একটাই ডিরেক্টরির উদরে সুখে বসবাস করত আপামর সমস্ত ব্যবহারকারীর যাবতীয় ফাইল। পরবর্তী ডিরেক্টরি কাঠামোর সঙ্গে তুলনা করে আলোচনা করতে গিয়ে এটাকেও রুট ডিরেক্টরি বলে ডাকা হয়, কিন্তু এক ভাবে দেখলে সেটা অর্থহীন। কারণ ডিরেক্টরিই তো সাকুল্যে এক পিস। এখানে এই ডিরেক্টরিটায় আছে ছটা ফাইল। বাঁদিক থেকে প্রথম তিনটের মালিক ব্যবহারকারী 'A', ফাইলগুলো হল 'file1A', 'file2A' এবং 'file3A'। পরের দুটো ফাইলের মালিক 'B', ফাইলদুটো 'file1B' এবং 'file2B'। ছয়-নম্বর ফাইল 'file1C', তার মালিক 'C'। এখানে আমরা ফাইলের নাম না লিখে শুধু মালিকের নাম লিখলেই পারতাম, কারণ, ফাইলের নামটার এখানে কোনো গুরুত্বই নেই তেমন। একজন ব্যবহারকারীর কটা ফাইল আছে, এবং কত নম্বর ফাইলটা আমরা খুঁজছি, এটাই জরুরি। এবং এখানে সেই খুঁজে পাওয়াটা হবে অসম্ভব দ্রুত, কারণ খোঁজার জায়গাটা তো মাত্র এক।

এই ধরনের ডিরেক্টরি কাঠামোয় বিপজ্জনক জায়গাটা এই যে, কোনো একটা নামে আমি আমার একটা ফাইল লিখেছি, এবার আপনি যদি ওই একই নামে কোনো ফাইল সেভ করেন, সেটা আমার ফাইলটাকে ওভাররাইট করবে। মানে, পুরোনোটাকে মুছে, সেটার জায়গায় নিজেকে লিখে দেবে ডিরেক্টরিতে। এই ধরনের ডিরেক্টরি কাঠামো আমাদের পিসি সিস্টেমে আর ব্যবহার হয়না, হওয়া সম্ভবও না। কিন্তু কোনো কোনো এমবেডেড সিস্টেমে ব্যবহার হয়। এই এমবেডেড সিস্টেম হল আর একটা এলাকা যেখানে প্রতিদিন গ্নু-লিনাক্সের ব্যবহার আগের চেয়ে বেড়ে যাচ্ছে। যাকগে, যা বলছিলাম, এমবেডেড সিস্টেমে এই একস্তর ডিরেক্টরি কাঠামো ব্যবহার হওয়া সম্ভব। ধরুন একটা যন্ত্রের নিরাপত্তা ব্যবস্থা — কে কে সেটা ব্যবহার করতে পারবে সেটা ভরে দিতে হচ্ছে যন্ত্রের সিস্টেমে। এখানে কয়েকজন ব্যবহারকারীর প্রত্যেকের একটা করে প্রোফাইল ভরে দেওয়া হচ্ছে। পরিচয়ব্যবস্থা, হয়তো তাতে ছবি থাকছে, আঙুলের ছাপ থাকছে, সই থাকছে — ইত্যাদি। মূল ব্যাপারটা হল, প্রতিজন পিছু একটা করে ফাইল এবং

সেই ফাইলের জায়গায় অন্য কারুর ফাইল সেভ করে দেওয়ার কোনো সম্ভাবনা নেই। যেমন আজ, ১২-ই নভেম্বর, খবরের কাগজে দেখছিলাম, বিভিন্ন জায়গায় এটিএম, যার মূল কথাটা হল অটোমেটিক টেলর মেশিন, কিন্তু কিরকম করে জানি এনিটাইম-মানি কথাটাই চালু হয়ে গেছে, সেই এটিএম ব্যবস্থা ভেঙে পড়ছে ভাইরাস আক্রমণে, এবং এই ব্যাপারে কথা বলতে গিয়ে একাধিক বিশেষজ্ঞ বলেছেন, উইনডোজে কাজ হচ্ছে বলেই দুশ্চিন্তা, গ্নু-লিনাক্সের তুলনায় অন্যান্য ওএস-এ নিরাপত্তাব্যবস্থার অভাবের কথা তো আগেই বলেছি, যদিও তার কাঠামোটার আলোচনা এখনো আসেনি।

৩.১।। দুই স্তরের ডিরেক্টরি কাঠামো

এর পরের স্তরে, বিভিন্ন ব্যবহারকারীর মধ্যে গোলযোগ এড়াতে, এল দুই স্তরের ডিরেক্টরি কাঠামো। এই ছবিটাতেও দেখুন, আমরা চৌকো দিয়ে ডিরেক্টরি আর গোল দিয়ে ফাইল বুঝিয়েছি। এই দুই-স্তরের ডিরেক্টরি কাঠামোয় প্রতিটি ইউজার বা ব্যবহারকারীকে দেওয়া হল তার নিজের নিজের ডিরেক্টরি, যাকে আমরা A, B আর C দিয়ে দেখিয়েছি, তিন জন ব্যবহারকারীর নামে। এখন দেখুন প্রথম জনের তিনটে ফাইল, দ্বিতীয় জনের দুটো ফাইল আর তৃতীয় জনের একটা ফাইল আছে এদের নিজের নিজের ডিরেক্টরিতে। তাই ফাইলনাম এক না আলাদা তাতে আর কিছু এসে যায়না। একাধিক ইউজারের কম্পিউটারে, বা, খুব সরল নেটওয়ার্ক সিস্টেমেও এই কাঠামো ব্যবহার করা যায়।

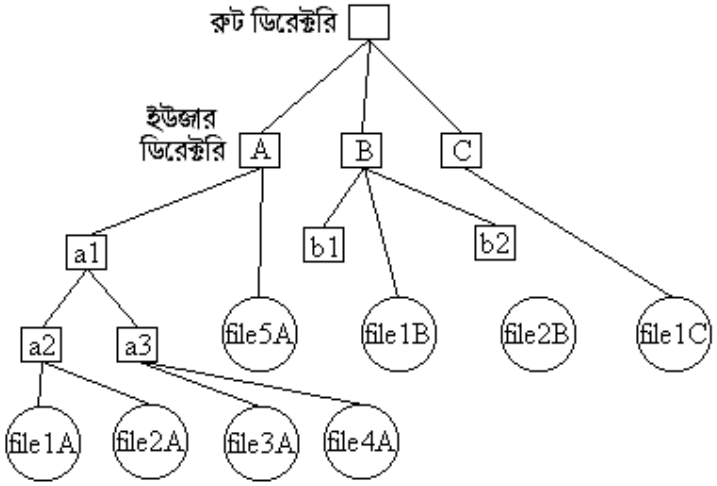


এই ধরনের ডিরেক্টরি কাঠামোয় সিস্টেমের একটা ভূমিকা চলে এল এই অর্থে যে, যখনই কোনো একজন ইউজার একটা ফাইল খোলার চেষ্টা করবে, সিস্টেমের জানা থাকতে হবে, এই ব্যবহারকারী কে, তার পরিচিতি। এবং সেই অনুযায়ী সেই ইউজারের ডিরেক্টরিতে গিয়ে ফাইলটা খুঁজবে। ওই এলাকাটারই উপর ওই ইউজারের অধিকার, অন্য এলাকা তার এক্তিয়ারের বাইরে। তার মানে, খেয়াল করুন, এইবার লগ-ইন ধারণাটা আসছে। এর সঙ্গে মেলান চার নম্বর দিনে মাল্টিপ্রোগ্রামিং থেকে মাল্টিইউজারে বিবর্তনের আলোচনাটা। লগ-ইন থাকছে মানেই এখানে প্রতিটি ব্যবহারকারীর একটা করে আইডেন্টিটি বা পরিচিতি এবং সেই পরিচিতি প্রমাণের দায় থাকছে, যা এক স্তরের ডিরেক্টরি কাঠামোয় ছিলনা। তার মানে পাসওয়ার্ড ইত্যাদি সমস্ত কিছুও এসে পড়ছে একই সঙ্গে।

এই স্তরের ডিরেক্টরি কাঠামোয় অন্য আর একটা ধরনের প্রয়োজন তৈরি হতে শুরু করল। ধরুন, প্রতিটি ইউজার শুধু তার নিজের ডিরেক্টরির ফাইলই খুলতে পারে, অন্য কোথাও কোনো ফাইলে তার অধিকার নেই। এই অবস্থায় যে সাধারণ কাজগুলো প্রতিটি ব্যবহারকারীকেই করতে হয়, তার প্রোগ্রামগুলোকেও প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই থাকতে হবে। আমরা যে প্রোগ্রামগুলোকে দেখেছি, '/bin' ডিরেক্টরিতে, 'ls' 'rm' 'cp' ইত্যাদি, সেগুলোর একটা করে কপি থাকতে হবে প্রতিটি ব্যবহারকারীর ডিরেক্টরিতেই। এটা একটা সিস্টেমের ইনএফিশিয়েন্সি, ক্যাবলামো। কাজের পরিমাণ, প্রোগ্রামের পরিমাণ যত বাড়ে এই ক্যাবলামোটা ততই আরো ব্যয়বহুল হয়ে পড়ে। অর্থাৎ, এমন একটা সিস্টেম ডিরেক্টরির দরকার পড়ছে যার মধ্যে থাকবে একজিকিউটেবল বাইনারি প্রোগ্রামগুলো — যা যে কোনো ইউজার ব্যবহার করতে পারবে। যে ডিরেক্টরির ফাইলদের উপর সব ইউজারেরই চালানোর অনুমতি থাকবে।

৩.২।। ক্রমানুসারী ডিরেক্টরি কাঠামো

এবার এল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার বা ক্রমানুসারী ডিরেক্টরি কাঠামো। দুই স্তরের কাঠামোয় নাম নিয়ে ঝামেলাটা চলে যায়, কিন্তু বহু ইউজারের বহু ফাইলের একটা ব্যবস্থায় খুব একটা সুবিধে হয়না। এমনকি একটা ব্যক্তিগত পিসি সিস্টেমেও খুব একটা কাজের নয়। আরো কাজের পরিমাণটা তো প্রতিদিনই বাড়ছিল। ইউজার তার কাজের যুক্তি মাফিক তার নিজের ফাইলগুলোকে আলাদা আলাদা ডিরেক্টরিতে রাখতে চাইবেই। যেমন আপনার ওই প্রবন্ধ আর বইয়ের চক্করটা। এই রকম প্রয়োজন ক্রমে তো বাড়তেই থাকে। লেখার ফাইল, গানের ফাইল, কোডের ফাইল, কম্পাইল করার পর বাইনারি একজিকিউটেবল ফাইল, ইমেল ফাইল, ওয়েবপেজ, ইত্যাদি। এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোয় সেটা করার সুযোগ এল ইউজারের। এই ছবিটায় দেখুন, সিস্টেমের বরাদ্দ করা তিনটে ইউজার ডিরেক্টরিকে আমরা দেখিয়েছি তিনজন ব্যবহারকারীর নামে — 'A', 'B', আর 'C'। আর তাদের নিজেদের তৈরি করা ডিরেক্টরিগুলোকে দেখিয়েছি ছোট হাতে, 'a1' 'a2' 'a3' 'b1' 'b2' — এইরকম। এখানে দেখুন, ইউজার 'A' তার নিজের ইউজার ডিরেক্টরি 'A'-র মধ্যে 'a1' ডিরেক্টরি তৈরি করেছে। 'a1' ডিরেক্টরির মধ্যে আবার সাবডিরেক্টরি 'a2' আর 'a3'। যার প্রত্যেকটায় দুটো করে মোট চারটে ফাইল। ইউজার 'A'-র পাঁচ নম্বর ফাইলটা আছে সরাসরি ইউজার ডিরেক্টরিতেই। ঠিক এই ভাবে যে কোনো ব্যবহারকারী তার নিজের ইউজার ডিরেক্টরিতে এবং তার মধ্যের যে কোনো সাবডিরেক্টরিতে যত খুশি ডিরেক্টরি এবং ফাইল বানাতে বা রাখতে পারে। একটা আধুনিক কম্পিউটার ব্যবস্থায় এই হায়েরার্কিকাল ডিরেক্টরি কাঠামোই ব্যবহৃত হয়। কিন্তু, খেয়াল করুন, ক্রমানুসারী বা হায়েরার্কিকাল ডিরেক্টরি কাঠামো আসা মাত্রই দরকার পড়তে শুরু করল পথনির্দেশ বা ফাইলের ঠিকানা। আসছি সেই কথায়।



৪।। পথনির্দেশ

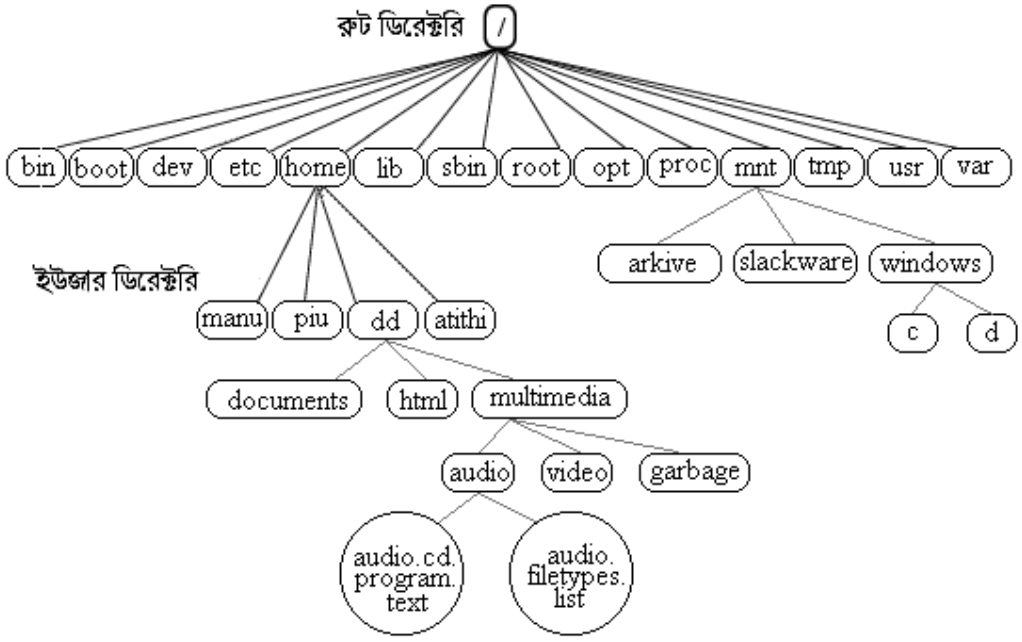
এবার ডিরেক্টরি কাঠামোয় সাজানো একটা ফাইল ব্যবস্থায় আমরা একটা জায়গা থেকে অন্য জায়গায় যাবার পথনির্দেশ করব কী করে? এর দুটো আলাদা উপায় আছে। একটা হল চুড়ান্ত পথনির্দেশ বা অ্যাবসলিউট পাথ (Absolute Path)। আর অন্যটা আপেক্ষিক পথনির্দেশ বা রিলেটিভ পাথ (Relative Path)। ধরুন, এতক্ষণ ধরে, কোনো ডিরেক্টরি বলতে হলেই আমরা যে শুধু তার নাম না বলে সঙ্গে একটা স্ল্যাশ জুড়ে দিচ্ছিলাম, এই আগের আগের প্যারাগ্রাফেই দেখুন বাইনারি ফাইলগুলো কোথায় থাকে বলতে গিয়ে, 'bin' না-বলে, বললাম '/bin'। কেন? এটা বোঝাতে যে ওই 'bin' ডিরেক্টরিটা আছে '/'-এর মধ্যে, যা হল একটা গ্নু-লিনাক্স কাঠামোয় রুট ডিরেক্টরি, মানে আর সমস্ত ডিরেক্টরিরাই যার মধ্যে রয়েছে। হয় সাবডিরেক্টরি, বা সাবডিরেক্টরির সাবডিরেক্টরি। চূড়ান্ত পথ বা অ্যাবসলিউট পাথটা শুরুই হয় এই রুট ডিরেক্টরি স্ল্যাশ '/' থেকে। আবার এই স্ল্যাশ-ই ব্যবহার হয় একটা মা-ডিরেক্টরির সঙ্গে তার ছানা-ডিরেক্টরির সম্পর্ক বোঝাতে। যেমন ধরুন, আমরা বললাম '/usr/local/bin'। এর মানে, মূল মা ডিরেক্টরি বা রুট ডিরেক্টরি '/', তার ছানা হল '/usr'। আবার '/usr' ডিরেক্টরির ছানা '/usr/local'। '/usr/local' ডিরেক্টরির '/usr/local/bin' — এটাই তার চূড়ান্ত ঠিকানা, সেই 'bin' ডিরেক্টরির কথা বলছি আমরা। যেই চূড়ান্ত

ঠিকানাটা দিয়ে দিলাম, '/usr/local/bin' আর '/bin' আলাদা ডিরেক্টরি বলে চেনা গেল, দুজনের একই নাম 'bin' হওয়া সত্ত্বেও। শুধু 'bin' বলে যা কখনো বোঝানো যেতনা।

চূড়ান্ত বা অ্যাবসলিউট পথ সবসময়ই শুরু হয় চূড়ান্ত বিন্দু বা মেরু মানে রুট ডিরেক্টরি থেকে। গ্নু-লিনাক্সে যাকে আমরা লিখি স্ল্যাশ (/) দিয়ে। খেয়াল রাখবেন, গ্নু-লিনাক্সে ডিরেক্টরি থেকে সাবডিরেক্টরিকে আলাদা করার উপায় সেপারেটরও ওই স্ল্যাশ (/)। যাবতীয় ইউনিক্স মেক রিমেক মিক্স রিমিক্সেই এটা প্রথা। উইন্ডোজের সেপারেটরটা হল ব্যাকস্ল্যাশ (\), মাল্টিক্স-এ ছিল 'গ্রেটার দ্যান' চিহ্ন (>)। ধরুন, 'usr' ডিরেক্টরির মধ্যে আছে 'local' ডিরেক্টরি, তার মধ্যে আছে 'share' ডিরেক্টরি, তার মধ্যে আছে 'doc' ডিরেক্টরি — এই পথনির্দেশটা গ্নু-লিনাক্স তথা যেকোনো ইউনিক্স, উইনডোজ আর মাল্টিক্স, এই তিন রকম প্রথায় লেখা হবে তিন ভাবে।

গ্নু-লিনাক্স — /usr/local/share/doc/
উইনডোজ — \usr\local\share\doc\
মাল্টিক্স — >usr>local>share>doc>

এখানে উইনডোজ পাথটা কিন্তু কোনো চূড়ান্ত পথ বা অ্যাবসলিউট পাথ নয়, কারণ, উইনডোজ '\' থেকে শুরু হয়না। উইনডোজ থেকে গ্নু-লিনাক্সে রুট ডিরেক্টরির ধারণাটা একদম আলাদা, সেটা আমরা পরে আরো ভালো করে বুঝতে পারব, মাউন্ট বোঝার সময়, এমনকি যদি 'C:' তার রুট ডিরেক্টরি বলে ধরে নিই, তাহলেও সেটা এখানে উল্লেখিত নেই। কিন্তু, গ্নু-লিনাক্স-এর বেলায় '/' দিয়ে শুরু মানেই সেটা অ্যাবসলিউট বা চূড়ান্ত পথ। সরাসরি রুট ডিরেক্টরি থেকে পুরো পথটা দেখাচ্ছে — যে অব্দি আমরা যেতে চাই।



অন্য রকমের পথনির্দেশটা হল আপেক্ষিক বা রিলেটিভ। আপেক্ষিক মানেই তার মধ্যে নিহিত আছে 'ওয়ার্কিং ডিরেক্টরি' বা 'কাজের ডিরেক্টরি' বলে একটা ধারণা, আপেক্ষিক বলতে সেই ওয়ার্কিং ডিরেক্টরির সাপেক্ষে আপেক্ষিক। এই শব্দবন্ধ 'ওয়ার্কিং ডিরেক্টরি' এসেছে আমাদের এই লেখায়, কোথায় মনে করতে পারছেন? একটা কমান্ডে, যে কমান্ডটা আপনি ব্যবহার করেছেন? কমান্ডটা ছিল 'pwd', 'প্রিন্ট ওয়ার্কিং ডিরেক্টরি', কাজের ডিরেক্টরি দেখাও। যে কমান্ড ব্যবহার করে আপনি দেখে নিয়েছিলেন রিডাইরেক্ট করা audio.cd.program.text ফাইলটা কোথায় সেভ করা হল। সেই '/home/dd/multimedia/audio' ডিরেক্টরিটাই তখন ছিল আপনার ওয়ার্কিং ডিরেক্টরি।

এবার যেই ওয়ার্কিং ডিরেক্টরি বলে একটা কিছু আছে, অমনি আমাদের পথ দেখানোর উপায়টাও তার সাপেক্ষে বদলে যাবে। প্রথমে ফাইলদুটোকে দেখে নিন, গোল চিহ্নে, 'audio.cd.program.text' আর 'audio.filetypes.list', যারা

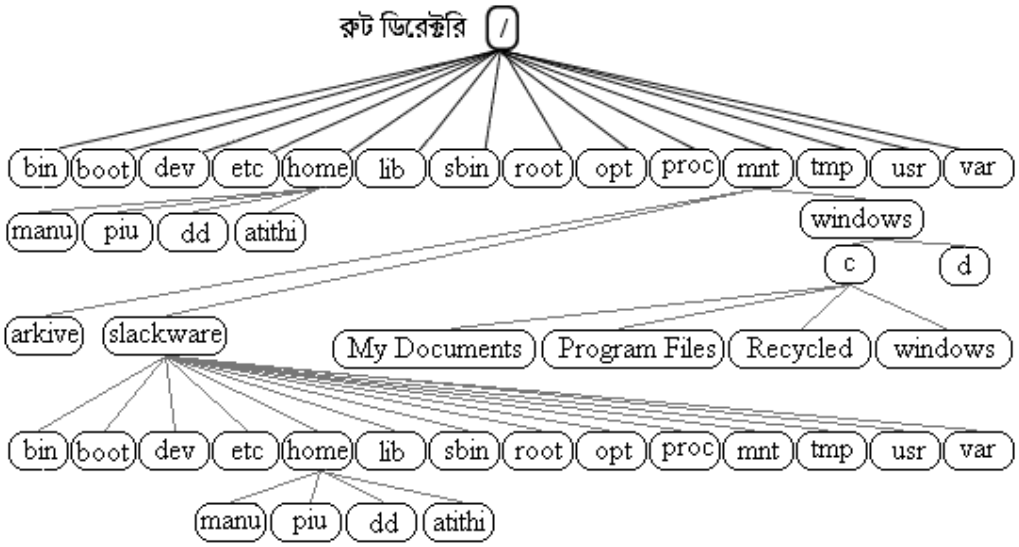
আপনাকে কিছুতেই ব্যাশ শেলে কমান্ড কমপ্লিশনের করার ট্যাবারাম পেতে দিচ্ছিল না দুটোরই শুরুতে ওই 'audio.' অংশটা থাকায়। ফাইলদুটোর বজ্জাতি এখানেই শেষ নয়, ছবিতে ডিরেক্টরির ছকগুলো বানিয়েছিলাম আমীর সাইজের, তার মধ্যে এই বচ্চন সাইজের ফাইল, কী বিকট দেখাচ্ছে, কিন্তু, সরি, কিছু করার নেই, আজ তেরোই ডিসেম্বর, শনিবার, পরশু জয়েন করছি, এখন আর সময় নেই নতুন করে এতগুলো পাতা ধরে ফাইলনাম বদলে আসার, এবং ভাঙার পক্ষে আমার আরো একটা ঠ্যাং এই মুহূর্তেই তৈরি আছে ঠিকই, হাত নয় কারণ লেখা যাবেনা, কিন্তু আর ছুটি নেই। ইন ফ্যাক্ট যা শুনছি, এতেই বোধহয় উইদাউট পে — তখন তো আপনাদের পাবনা। এবার ওই বারো হাত কাঁকুড়ের তেরো হাত বীচি ব্রান্ডের ফাইলদুটোকে দেখুন। এর অ্যাবসলিউট বা চূড়ান্ত ঠিকানাটা ভাবুন, শুরু হয়েছে একদম উপরের রুট ডিরেক্টরি স্ল্যাশ '/' থেকে, তারপর 'home', তারপর 'dd' মানে একজন ব্যবহারকারী, এইটা হল ইউজার ডিরেক্টরি, আগের হায়েরার্কির ছকের সঙ্গে মেলান, তার মধ্যে 'multimedia', তার মধ্যে 'audio' — এইখানে রয়েছেন 'audio.cd.program.text' আর 'audio.filetypes.list'।

তাহলে চূড়ান্ত পথনির্দেশটা মেলানো গেল '/home/dd/multimedia/audio/audio.cd.program.text'। এবার এটাকে আপেক্ষিক ভাবে ভাবুন। ধরুন, আগের দিন ফাইলটাকে সেভ করে বেরিয়ে গেছি। এবার পরের দিন লগ-ইন করলাম। আমি লগ-ইন করেছি আমার ইউজার ডিরেক্টরিতে, মানে '/home/dd'। ঢুকেই আমি 'pwd' কমান্ডটা লিখে যদি এন্টার মারি, কমান্ড প্রম্পটে ফুটে উঠবে '/home/dd'। এবার এই '/home/dd' ডিরেক্টরিতে আমার ঠিক সামনে আছে তিনটে ডিরেক্টরি, 'documents', 'html' এবং 'multimedia'। আমি যেতে চাই 'multimedia' ডিরেক্টরির মধ্যে 'audio' ডিরেক্টরিতে। এইজন্যে এবার কমান্ড দেব 'chdir'। 'chdir multimedia'। এবার, কমান্ড প্রম্পট ফিরে এলে, 'pwd' করলে দেখাবে '/home/dd/multimedia'। তার মানে, গোটা ঠিকানার বদলে শুধু 'multimedia' নামেই কাজ হল, কারণ 'multimedia' ডিরেক্টরিটা ঠিক আমার ওয়ার্কিং ডিরেক্টরিতেই বিরাজ করছে। এখানে যদি বড় কমান্ড দিতাম, 'chdir /home/dd/multimedia', তাহলেও একই কাজ হত। তার মানে, তখন চূড়ান্ত বা অ্যাবসলিউট পথ ব্যবহার হত, ওয়ার্কিং ডিরেক্টরির সাপেক্ষে রিলেটিভ পথ নয়।

'mv' কমান্ডটা উল্লেখ করার সময় বলেছিলাম, একটা কাজ নাম-বদলানো, অন্য কাজটা হল একটা ফাইল বা ডিরেক্টরিকে একটা জায়গা থেকে অন্য জায়গায় নড়ানো। 'জায়গা' শব্দটাকে খেয়াল করুন। এটা আর আমাদের 'স্থান' অর্থে ভৌত ভৌগোলিক জায়গা নয়। একটা প্রতীকী ভূমি, যা তৈরি হয়েছে ফাইল আর ডিরেক্টরিদের মিলিয়ে, এক নম্বর দিনের আলোচনায় যার কথা বলেছি। পরে যখন হার্ডডিস্কের ভৌত জ্যামিতিটা বুঝব আমরা তখন দেখব, ভৌত ডিটেইলদের সঙ্গে এর কোনো সম্পর্কই নেই। 'রুচিশীল' ডিরেক্টরির 'শিক্ষিত' সাবডিরেক্টরির 'রবীন্দ্র' ফাইলটা হার্ডডিস্ক প্ল্যাটারের সেক্টরের জ্যামিতির নিরিখে থাকতে পারে একেবারে 'জন্তু' ডিরেক্টরির 'বলদ' সাবডিরেক্টরির 'ন্যাজ' ফাইলের গায়ে হেলান দিয়ে। ওতে কিছু এসে যায়না ভাই, প্রতীকী পরিচয়গুলো ইনট্যাক্ট থাকলেই হল। যাইহোক, যা বলছিলাম, 'mv' কমান্ডের দ্বিতীয় কাজটা, মানে, একটা ফাইল বা ডিরেক্টরিকে একটা ডিরেক্টরি ঠিকানা থেকে অন্য ডিরেক্টরি ঠিকানায় নড়ানোর কাজটা এবারে আমরা আনতে পারব, আগে পারিনি। কারণ, ঠিকানা দিতে গেলেই তো চূড়ান্ত হোক আপেক্ষিক হোক, পথের ধারণাটা দরকার পড়বে, যা এতক্ষণ ছিলনা আমাদের।

এবার এই '/home/dd/multimedia' ডিরেক্টরিতে দাঁড়িয়ে আমরা 'mv' দিয়ে ফাইল ডিরেক্টরি নড়াতে পারব। এখানে আমার ওয়ার্কিং ডিরেক্টরিতে রয়েছে, ছবিতে দেখুন, 'audio', 'video', আর 'garbage' নামে তিনটে ডিরেক্টরি। এই 'audio' ডিরেক্টরির মধ্যে আছে সেই ঝামেলা পাকানো ফাইল 'audio.filetypes.list'। এবার তাকে আমরা নড়িয়ে 'garbage' ডিরেক্টরিতে নিয়ে যাব। 'audio' ডিরেক্টরি থেকে 'garbage' ডিরেক্টরিতে 'mv' করার জন্যে কমান্ড দেব, 'mv audio/audio.filetypes.list garbage/audio.filetypes.list'। এই আদেশে আমরা ব্যাবহার করলাম আপেক্ষিক বা রিলেটিভ পথ, কারণ এই ডিরেক্টরিতে 'audio' আর 'garbage' ডিরেক্টরিদুটো আমার সামনেই রয়েছে, তার একটার থেকে একটা ফাইল নিয়ে আমি অন্য ডিরেক্টরিতে রাখতে বললাম। এতক্ষণ আমরা 'ls' কমান্ড দিয়েছি যে ডিরেক্টরির ফাইল দেখছি সেই ডিরেক্টরিতে দাঁড়িয়ে, কারণ আমরা পথনির্দেশ জানতাম না। এখন পথ-চিনে-যাওয়া আমরা এখানে দাঁড়িয়েই কমান্ড দেব, 'ls garbage', এবং দেখতে পাব ফাইলটা ওই ডিরেক্টরিতে চলে গেছে। এই রিলেটিভ পথেই কাজ চলে গেল, কারণ 'garbage' ডিরেক্টরিটা আমার ওয়ার্কিং ডিরেক্টরিতেই আছে। '/home/dd' ডিরেক্টরিতে দাঁড়িয়ে থাকলে, একই 'ls' কমান্ডের একই কাজের জন্যে কমান্ড

দিতে হত, '`ls multimedia/garbage`'। এটাও আপেক্ষিক বা রিলেটিভ পথ, শুধু একটু বড়। কারণ, যে ডিরেক্টরিটা আমি দেখতে চাইছি, সেটা ওয়ার্কিং ডিরেক্টরিরই একটা সাবডিরেক্টরিতে আছে যার নাম 'multimedia', কিন্তু সরাসরি এই ডিরেক্টরিতে নেই। ধরুন, আমরা যদি এই পথের সম্পূর্ণ বাইরে, রুট ডিরেক্টরির নিচেই সবচেয়ে বাঁদিকের সাবডিরেক্টরিটায় থাকতাম, মানে '/bin' ডিরেক্টরিতে, তাহলে আমাদের ওই একই ফাইল দেখার জন্যে কমান্ড দিতে হত '`ls /home/dd/multimedia/garbage`'। এটা হত চূড়ান্ত বা অ্যাবসলিউট পথ।



৫।। ঐক্যবদ্ধ ফাইলসিস্টেম

আমাদের পথনির্দেশ জানা হল। জানলাম, কী করে একটা হায়েরার্কিকাল বা ক্রমানুসারী ডিরেক্টরি কাঠামোয় একটা ফাইলের বা ডিরেক্টরির ঠিকানা দিতে হয়। এবার আমরা যাব, এই ডিরেক্টরি কাঠামোর বাস্তব গ্নু-লিনাক্স আকার ঠিক কী হয় তার একটা পরিচিতিতে। তার আগে, এই ছবিটায় ঠিক রুট ডিরেক্টরি বা '/'-র নিচেই প্রথম স্তরের সাবডিরেক্টরিগুলো খেয়াল করুন, '/bin', '/boot', '/dev', '/etc', '/home', '/lib', '/sbin', '/root', '/opt', '/proc', '/mnt', '/tmp', '/usr', '/var'। এগুলো ইচ্ছেমত নয়, আমার সুজে সিস্টেমের সত্যিকারের ডিরেক্টরি কাঠামোটাই দিয়েছি। শুধু আঁকার সময় ডিরেক্টরির নামগুলো ভুল করে বর্ণানুক্রমে রাখা হয়নি, লেখাতেও তাই রাখলাম। ছবিটা মন দিয়ে দেখে রাখুন, বারবার এটার কথা আসবে। রুট ডিরেক্টরি বা স্ল্যাশ, '/', এবং তার পরেই তার থেকে প্রবাহিত এই ডিরেক্টরিগুলো — এই গোটাটা মিলিয়ে তৈরি হয় গ্নু-লিনাক্সের ঐক্যবদ্ধ ডিরেক্টরি সিস্টেম। খেয়াল করুন, এটা একটা সিঙ্গল হায়েরার্কিকাল স্ট্রাকচার বা এক-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। একটা উৎস, রুট ডিরেক্টরি '/', তার থেকে প্রবাহিত আর সমস্ত ডিরেক্টরি। যে ফাইল যে ডিরেক্টরিই ভাবুন, তা এই মূল উৎস থেকে প্রবাহিত কোনো এক ধারায় বা উপধারায় আছে, রুট ডিরেক্টরির পেটে কোনো এক ডিরেক্টরির কোনো সাবডিরেক্টরিতে।

উইন্ডোজ থেকে গ্নু-লিনাক্সে অভিবাসীরা প্রথম প্রথম তাদের হারিয়ে যাওয়া ড্রাইভ নিয়ে হাপিত্যেশ করেন, কোথায় গেলি 'c:\' মানে সি-ড্রাইভ, বা 'd:\' বা 'a:\' ইত্যাদি। এখানে তেমন কোনো ড্রাইভের গল্পই নেই। ওইরকম আলাদা আলাদা ড্রাইভ আছে বলে উইন্ডোজ হল একটা মাল্টিপল হায়েরার্কিকাল ডিরেক্টরি স্ট্রাকচার, বা বহু-উৎসের ক্রমানুসারী ডিরেক্টরি কাঠামো। গ্নু-লিনাক্সের কাঠামোর সঙ্গে পার্থক্যটা খেয়াল করুন। সি-ড্রাইভ, ডি-ড্রাইভ, এ-ড্রাইভ, এই সবগুলোরই নিজের নিজের এক একটা রুট বা মূল ডিরেক্টরি আছে। বহুসময়, কথার কথা হিশেবে, উইন্ডোজেও 'c:\' বা সি-ড্রাইভটাকে রুট ডিরেক্টরি বলে ডাকা হয়। যেহেতু, বুট করার সময় উইন্ডোজের একদম প্রাথমিকতম সিস্টেম ফাইলগুলো এই 'c:\' থেকেই পড়া হয়। এখানের সেই মূল ফাইলগুলোর সিস্টেম ফাইলগুলোর কিছু কিছু নাম আমরা আগেই উল্লেখ করেছি, 'msdos.sys', 'io.sys', 'command.com', 'drvspace.bin' ইত্যাদি, এর মধ্যে কিছু ফাইল আবার গোপন বা হিডন, তাই 'dir/a' করে দেখতে হয় তাদের, এদিকে যে কেউই আবার 'attrib' কমান্ড দিয়ে যে কোনো সময় তাদের গোপনতার বস্ত্রহরণ করে দিতে পারে,

যাকগে। উইন্ডোজে 'sys' বলে একটা কমান্ডও আছে যা দিয়ে একটা ড্রাইভে এই ফাইলগুলো কপি করে নেওয়া যায়। এই ফাইলগুলো থাকে বলেই ওই ড্রাইভটা উইন্ডোজ দিয়ে বুটেবল হয়। আবার উইন্ডোজ ওএস কাজ করার সময় যে ফাইলগুলো বাধ্যতামূলক ভাবে প্রয়োজন পড়ে সেই ফাইলগুলো থাকে এই 'c:\' ড্রাইভের এই রুট ডিরেক্টরির মধ্যে 'c:\windows' ডিরেক্টরিতে, বিশেষ করে তার মধ্যের 'c:\windows\system' ডিরেক্টরিতে। অথচ মজা দেখুন, অন্য হার্ডডিস্কে বা অন্য পার্টিশনে বা অন্য পার্টিশনে থাকা কোনো ফাইল আপনি এই 'c:\' ডিরেক্টরির ভিতর কোনো ডিরেক্টরিতেই পাবেন না। ডি-ড্রাইভ 'd:\' বা এ-ড্রাইভ 'a:\'-এ কোনো ফাইল পেতে হলে প্রথমে আপনাকে 'd:' বা 'a:' কমান্ড দিয়ে ড্রাইভ বদলাতে হবে। তখন দেখবেন, কমান্ড প্রম্পটে আপনার ঠিকানা দেখাচ্ছে 'd:' বা 'a:'। মানে আপনি ডি-ড্রাইভ বা এ-ড্রাইভের রুট ডিরেক্টরিতে আছেন (এমএসডস-প্রম্পটের বদলে যদি যদি উইন্ডোজ এক্সপ্লোরারে থাকেন তাহলে দেখবেন লোকেশন বারে, মানে একদম উপরের দিকে, মেনুবারের নিচেই সাদা জায়গাটায়,আপনার গোটা ঠিকানাটা ফুটে উঠছে, কোন ড্রাইভের কোন ডিরেক্টরির কোথায় আপনি আছেন)। তারপর তার ভিতরের ডিরেক্টরি কাঠামোর ঠিকানা মেনে আপনাকে ফাইলটায় পৌঁছতে হবে। অর্থাৎ, রুট ডিরেক্টরি ধারণাটাই এখানে অকেজো হয়ে পড়ছে, রুট হয়ে দাঁড়াচ্ছে একাধিক। । উইন্ডোজের প্রতি ড্রাইভেই সেই অর্থে রুট ডিরেক্টরি থাকে একটা করে, কিন্তু গ্নু-লিনাক্সে তথা যে কোনো ইউনিক্সে রুট ডিরেক্টরি হল এক এবং অদ্বিতীয় '/'। এইজন্যে গ্নু-লিনাক্স ফাইলসিস্টেমকে একটা ঐক্যবদ্ধ ইউনিফায়েড ফাইল সিস্টেম বলা হয়।

পরে আমরা দেখব গ্নু-লিনাক্সে যে কোনো হার্ডডিস্ক (বা অন্য ডিস্কের-ও) পার্টিশনকে নিজের ইচ্ছেমতন নিয়ে আসা যায় ঐক্যবদ্ধ ফাইল সিস্টেমের ভিতরে। একে টেকনিকাল পরিভাষায় বলে মাউন্ট করা। পছন্দের যেকোনো ডিরেক্টরিতে পছন্দের যেকোনো ডিস্ক বা পার্টিশন মাউন্ট করা যায়। যে পার্টিশন মাউন্ট করা হয় সেই পার্টিশনের নিজের ডিরেক্টরি কাঠামোর হায়েরার্কিটা এবার গোটা ফাইলসিস্টেম হায়েরার্কির একটা উপধারা হয়ে পড়ে। ধরুন, উইন্ডোজের সি-ড্রাইভ বা 'C:\' মানে যে পার্টিশনটা তাকে মাউন্ট করা হয়েছে '/mnt/windows/c' ডিরেক্টরিতে। এই '/mnt/windows/c' ডিরেক্টরিটা হল ওই পার্টিশনের মাউন্ট-পয়েন্ট। এবার, মাউন্ট করার পর, 'C:\' ড্রাইভের ভিতরে ডিরেক্টরি এবং ফাইলের কাঠামোটাকে হুবহু আমি পাব '/mnt/windows/c' ডিরেক্টরিতে। 'C:\' ড্রাইভের ভিতরের গোটা ডিরেক্টরি কাঠামোটা এখন রুট ডিরেক্টরি '/'-এর ভিতর একটা উপধারা বা সাবসেট হয়ে গেল। উইন্ডোজের ডি-ড্রাইভ যদি মাউন্ট করি '/mnt/windows/d' ডিরেক্টরিতে, এই ডিরেক্টরিতে তখন ডি-ড্রাইভের গোটা ডিরেক্টরি কাঠামোটা পাব। এবং এখানে দেখুন, দুটো ড্রাইভই থাকছে একই রুট ডিরেক্টরি '/'-এর ভিতর দুটো উপধারায়, '/mnt/windows/c' এবং '/mnt/windows/d'। ড্রাইভ বদলানোর গল্পই নেই। এই ভাবে যে কোনো পার্টিশনই মাউন্ট করা যায় গ্নু-লিনাক্সের এক ঐক্যবদ্ধ ফাইলসিস্টেমের ভিতরে কোনো ডিরেক্টরিতে, নেটওয়ার্কে যুক্ত অন্য মেশিনের পার্টিশনেও। মাউন্টের আলোচনার আগে অব্দি এর পুরোটা কিন্তু স্পষ্ট হবেনা।

এমনকি একটা পার্টিশনকে দুটো সিস্টেমও শেয়ার করতে পারে। মানে দুজনেই ব্যবহার করতে পারে। কিছু কিছু ডিরেক্টরির বেলায় পারেনা, যেমন '/boot', '/etc', বা '/var'। এই ডিরেক্টরিগুলো প্রতিটি সিস্টেমের নিজস্ব। '/etc' ডিরেক্টরিতে থাকে সিস্টেমের সমস্ত কনফিগারেশন ফাইল, পাঁচ নম্বর দিনেই দেখেছেন সিস্টেমের প্রতিটি পাসওয়ার্ড থাকে এই ডিরেক্টরির 'passwd' ফাইলে। সেটা দুটো সিস্টেম একই সঙ্গে শেয়ার করবে কী করে? প্রতি সিস্টেমের প্রতিটি ইউজারের পাসওয়ার্ড তো আলাদা। '/boot' ডিরেক্টরিতে থাকে আপনার সিস্টেমের কারনেল এবং তার খুঁটিনাটি, সেটাও সম্ভব না। আট আর নয় নম্বর দিনে, লিনাক্স ফাইলসিস্টেম হায়েরার্কির সর্বজনগ্রাহ্য নিরিখ বা স্ট্যান্ডার্ডের আলোচনায় আমরা এগুলো ভালো করে জানতে পারব।

এই ঐক্যবদ্ধ ফাইলসিস্টেমের আর একটা গুরুত্বপূর্ণ জায়গা হল গ্নু-লিনাক্সে অত্যুন্নত মেমরি ব্যবহার ব্যবস্থা, যার অন্য নাম ক্যাশে। গোটা মেমরি ব্যবহারের প্রক্রিয়াটাকেই যা অনেকটা দ্রুততর করে তোলে। শূন্য এক আর দুই নম্বর দিনে বারবার আমাদের আলোচনায় এসেছে, কম্পিউটারে মূলত দুই রকমের মেমরির কথা, র‍্যাম আর রম। এই র‍্যামকে আবার দুইভাবে দেখেছি আমরা, ভৌত এবং ভৌতিক, ফিজিকাল এবং ভারচুয়াল। এই ভারচুয়াল মেমরি, সোয়াপফাইল, ক্যাশে — গোটা ব্যবস্থাটাই চলে একটা বিপুল পরিমাণ তথ্য অস্থায়ী ভাবে ভৌত ডিস্কে তুলে রাখার ভিত্তিতে। এই ব্যবস্থাটা এত ধারালো বলে গ্নু-লিনাক্সে মেমরির ব্যবহার চমৎকার। উইনডোজ ব্যবস্থায় র‍্যাম খুব বেশি বাড়ালে যেমন আদতে সিস্টেমের গতি কমেও যেতে পারে। যা ঘটে সেটা বাংলায় বললে এই যে, সিপিইউ-র স্ক্রু

ঢিলে হয়ে যায় ওই স্তূপ স্তূপ মেমরির পাতার মধ্যে, কী খুঁজছে সেটাই হারিয়ে ফেলে। ফাইনালের আগের দিন রাত্তিরে ধরুন আপনাকে দেওয়া হল, কালকের কোশ্চেন আসতে পারে এই একুশটা মাত্র বই থেকে, পড়ে ফেলো বাবা। পরের দিন আপনি হলে গিয়ে, কুল বাংলা কোশ্চেনের উর্দূ উত্তর লিখে চলে এলেন।

গ্নু-লিনাক্স আসলে ঘুঘু জিনিষ, সে গোটাটা পড়েই না, টুকলি বানাতে থাকে। এবার ভাবুন, গোটা সিস্টেমটা যদি এককেন্দ্রিক ঐক্যবদ্ধ না-হয়, তাহলে, আবার সেই সমস্যা, কোথাকার টুকলি কোথায় যাবে, কোন মেমরি বাফার কোথায় লেখা হবে। এই জন্যে একটা গ্নু-লিনাক্স সিস্টেমে সঠিক শাট-ডাউনটা খুব জরুরি। হঠাৎ বিদ্যুৎ চলে গেলে বা অফ করে দিলে সিস্টেমের উপর প্রচণ্ড চাপ পড়ে। যদিও তেমন সমস্যা কিছু হতে দেখিনি কখনো, গ্নু-লিনাক্স ফাইলসিস্টেম এত জোরালো বা রোবাস্ট। তবু, না-করাই ভালো। অনেক উপায় আছে শাট-ডাউনের, ধীরে ধীরে আপনি সবগুলোই শিখে যাবেন। তবে একটা বেশ সহজ এবং নিরাপদ উপায় হল কন্ট্রোল-অল্ট-ডিলিট মানে রিবুটের নির্দেশ। এরপর আঙুল উঁচিয়ে বসে থাকা, আবার বুট করার আগেই অফ করে দেওয়া।

ঐক্যবদ্ধ ফাইলসিস্টেমের ছবিটায় '/home' ডিরেক্টরির মধ্যে দেখুন, চারটে ইউজার ডিরেক্টরি '/home/atithi', '/home/manu', '/home/piu' এবং '/home/dd'। পাঁচ নম্বর দিনে দেখার জন্যে তুলে দেওয়া '/etc/passwd' ফাইল থেকে মিলিয়ে নিন। আর, একটু মন দিয়ে খেয়াল করুন '/root' ডিরেক্টরিটা। এটা কিন্তু গোটা হায়েরার্কির চূড়ান্ত মা-ডিরেক্টরি মানে স্ল্যাশ বা '/' নয়। এটা হল সুপারইউজার বা রুট নামক ইউজারের হোম। অন্য সব ইউজারের হোম ডিরেক্টরি হয় '/home' ডিরেক্টরির ভিতর। শুধু রুটের হোম হয় অন্য জায়গায়। এই যে ডিরেক্টরি কাঠামোটা আমি এখানে দেখিয়েছি, এটা গ্নু-লিনাক্সের একটা বিশেষ ফ্লেভারের, যার নাম সুজে ৮.২, ডিস্ট্রিবিউশনটা সুজে, ৮.২ তার সংস্করণ। এই ডিরেক্টরি কাঠামোটা, এর খুঁটিনাটি চেহারায়, এমনকি এর উপরসা অবয়বেও, ফ্লেভার থেকে ফ্লেভারে, মানে ডিস্ট্রিবিউশন থেকে ডিস্ট্রিবিউশনে বেশ বদলায়। আবার একটা একদেশতাও থাকে, গ্নু-লিনাক্স বলে নিজেকে ডাকতে গেলে কিছু ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড তাকে মানতে হয়। কিন্তু সেই ন্যূনতম জায়গাটুকুকে বাদ দিয়ে অনেক কনফিগারেশন বা আকারায়ন একটা থেকে অন্যটায় বদলায়। তাই, এই কাঠামোটা মুখস্থ না রেখে এর ধারণাটা মাথায় রাখার চেষ্টা করুন। এর কোনটার কী মানে, কেন থাকে, এইসবে আসছি — তার আগে এখানে একটা তালিকা দিয়ে রাখি, কত আলাদা আলাদা ফ্লেভার আছে গ্নু-লিনাক্সের। একটা ভারি চমৎকার বই থেকে তুলে দিচ্ছি। বইটার নাম 'ফ্রি ফর অল — হাউ লিনাক্স অ্যান্ড দি ফ্রি সফটওয়ার মুভমেন্ট আন্ডারকাট দি হাইটেক টাইটানস', লেখকের নাম পিটার ওয়েইনার, 'http://www.wayner.org/books/ffa/'।

বইটা শুরু হচ্ছে নিরানববই-এর সেই কুখ্যাত গেটস-মামলা দিয়ে — মাইক্রোসফট গোটা সফটওয়ার ব্যবসায় মনোপলি তৈরি করে অন্য সবার ব্যবসা করার আয় করার উপায় বন্ধ করে দিচ্ছে কিনা। এই মামলায়, মাইক্রোসফট যে মনোপলি নয়, তার যথেষ্ট ভয়াল সব প্রতিদ্বন্দ্বী আছে এটা বোঝাতে গিয়ে গেটস-এর উকিল উল্লেখ করেছিল গ্নু-লিনাক্স এবং এফএসএফ-এর কথা। পিটার ওয়েইনার এই গোটা অশ্লীল রসিকতাটার অভ্যন্তরস্থ বিকটতাটাকে চমৎকার দেখিয়েছেন — প্রশ্নটা এখানে টাকা, ক্যাশ, অন্য সবার টাকা আয় বন্ধ করে দেওয়ার জন্যে নোংরামি তুমি করছ কি করছ না, আর তাতে উদাহরণ দিচ্ছ লিনাক্স আর এফএসএফের, যারা প্রথম থেকেই বিনা পয়সায় দিয়ে দিচ্ছে গোটাটা, এবং শুধু নিজেদের সফটওয়ারকে নিয়ন্ত্রণ করছে-না তাই নয়, সফটওয়ার-প্রকৌশল-কম্পিউটার-চিন্তার গোটা প্রবাহটা যাতে কিছুতেই নিয়ন্ত্রণ করা না-যায় সেই ব্যবস্থা করার চেষ্টা করছে তাদের জিপিএল লাইসেন্স দিয়ে, যাকে অনেকেই মজা করে ডাকে 'কপিলেফট'। বইটা পড়ুন, সত্যিই ভালো লাগবে। এই বইটা থেকেই তুলে দিই লিনাক্স ডিস্ট্রিবিউশনের তালিকাটা, এবং ইংরিজি অক্ষরেই দিচ্ছি, কারণ এর অনেকগুলোর নামই বাপের জন্মে শুনিনি, অনেকগুলোই হল ইউরোপিয় নাম। একদম ল্যাটিন বর্ণমালার অনুক্রমে,

Alzza Linux, Apokalypse, Armed Linux, Bad Penguin Linux, Bastille Linux, Best Linux (Finnish/Swedish), Bifrost, Black Cat Linux (Ukrainian/Russian), Caldera OpenLinux, CCLinux, Chinese Linux Extension, Complete Linux, Conectiva Linux (Brazilian), Debian GNU/Linux, Definite Linux, DemoLinux, DLD, DLite, DLX, DragonLinux, easyLinux, Enoch, Eridani Star System, Eonova Linux, e-smith server and gateway, Eurielec Linux (Spanish), eXecutive Linux, floppyfw, Floppix, Green Frog Linux, hal91, Hard Hat Linux, Immunix, Independence, Jurix, Kha0s Linux, KRUD, KSI Linux, Laetos, LEM, Linux Cyrillic Edition, LinuxGT, Linux-Kheops (French), Linux MLD (Japanese), LinuxOne OS, LinuxPPC, LinuxPPP (Mexican), Linux Pro Plus, Linux Router Project, LOAF, LSD, Mandrake, Mastodon, MicroLinux, MkLinux, muLinux,

nanoLinux II, NoMad Linux, OpenClassroom, Peanut Linux, Plamo Linux, PLD, Project Ballantain, PROSA, QuadLinux, Red Hat, Rock Linux, RunOnCD, ShareTheNet, Skygate, Slackware, Small Linux, Stampede, Stataboware, Storm Linux, SuSE, Tomsrtbt, Trinux, TurboLinux, uClinux, Vine Linux, WinLinux 2000, Xdenu, XTeamLinux, and Yellow Dog Linux

ওয়েইনারও লিখেছেন, এটা সমাপ্ত কিনা জানিনা। আর আমি তো জানিই-না। তবে যেকটার নাম জানি তার মধ্যে মাংকি লিনাক্স ছাড়া সবকটাই আছে। তবে মাংকিটা নিজেই কোনো ডিস্ট্রিবিউশান না একটু বদলে নেওয়া তা জানিনা, যেমন নপিক্স (Knoppix) হল ডেবিয়ান ডিস্ট্রিবিউশানটাকে একটু বদলে নেওয়া। তবে ওয়েইনারের বইটা ২০০০-এ বেরোনো আর গ্নু-লিনাক্সের মানচিত্রটা ভারি দ্রুত বদলায়। গ্রহজোড়া কমিউনিটির অগণ্য মানুষের জ্যান্ত মনগুলো বদলায়। মহাভারতের ধর্ম-যখের সেই প্রশ্নোত্তর মনে আছে? মন তো সবচেয়ে দ্রুতগতি, সার্কিটের মধ্যে বিচরমান তথ্যবিদ্যুতের চেয়েও। এবার ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড, এখানে যা আমি জাস্ট ছুঁয়ে যাব, গোটা বিষয়টা খুবই বড় এবং বেশ ইন্টারেস্টিং। এই প্রসঙ্গটায় আবার ফেরত আসব আমরা, নয় নম্বর দিনে।

৬।। ফাইলসিস্টেমের মধ্যে আলাদা আলাদা ফাইলসিস্টেম

দুয়েকটা ধারণা আগে স্পষ্ট করে নেওয়া যাক। ফাইলকাঠামো বা ফাইলসিস্টেম বলতে আমরা বোঝাচ্ছি, অপারেটিং সিস্টেম যে ডিরেক্টরি এবং ফাইলের ছক ব্যবহার করে তার গোটাটাকেই। মানে, এক কথায়, গোটা ডিস্কের প্রতিটি পার্টিশনের প্রতিটি ডিরেক্টরি আর ফাইলের সংগঠন। কার মধ্যে কোথায় কে আছে, কাকে কোথায় পাব। 'ফাইলসিস্টেম' শব্দটার একটা অর্থ এটা। এখন থেকে এটাকে ডাকব ঐক্যবদ্ধ বা ইউনিফায়েড ফাইলসিস্টেম। আবার একটা ডিস্কে বা পার্টিশনে কী ভাবে ফাইল রাখা আর লেখা হবে তারও আলাদা আলাদা উপায় আছে। পার্টিশনের ডিরেক্টরি এবং ফাইলের মোট তথ্য কী ভাবে লেখা, হিশেব-রাখা এবং নিয়ন্ত্রিত হবে, এর আলাদা আলাদা রকম — 'ফাইলসিস্টেম' শব্দটার দ্বিতীয় অর্থ এটা। শুধু 'ফাইলসিস্টেম' বললে আমরা এটাকেই বুঝব এখন থেকে — আলাদা আলাদা পার্টিশনে ফাইল এবং ডিরেক্টরির আধারে তথ্য লেখার এবং রাখার আলাদা আলাদা কায়দা। যেমন আমার মেশিনের দুটো হার্ড ডিস্কের সাতটা পার্টিশনে আছে চার রকমের ফাইলসিস্টেম। এক্সএফএস, রাইজারএফএস, সোয়াপ, আর উইন্ডোজ ফ্যাট৩২ — ডিস্কের ফাইলগুলোয় তথ্য লেখা এবং রাখার আলাদা আলাদা কায়দা। আরো আছে, ইএক্সটি২ ইএক্সটি৩ ইত্যাদি। এদের ডাকছি ফাইলসিস্টেম। যখন একটা মেশিনের গোটা ডিরেক্টরি আর ফাইলের হায়েরার্কিবদ্ধ কাঠামোটাকে ভাবছি, সেটাকে ডাকছি ঐক্যবদ্ধ ফাইলসিস্টেম। আমার মেশিনের ঐক্যবদ্ধ ফাইলসিস্টেমে ফ্যাট৩২ পার্টিশনদুটো মাউন্ট হয় '/mnt/windows/c' আর '/mnt/windows/d' ডিরেক্টরিতে, একটা রাইজারএফএস পার্টিশন মাউন্ট হয় '/mnt/slackware' ডিরেক্টরিতে, অন্যটা '/mnt/arkive' ডিরেক্টরিতে। আগের সেকশনের ঐক্যবদ্ধ ফাইলসিস্টেমের ছবির সঙ্গে মেলান। সুজে ওএস-এর গোটা ঐক্যবদ্ধ ফাইলসিস্টেমটা আছে তিনটে এক্সএফএস ফাইলসিস্টেম পার্টিশন এবং অন্য রাইজারএফএস ফ্যাট৩২ আর সোয়াপ ফাইলসিস্টেম পার্টিশনগুলোকে মিলিয়ে। এখন একটু ধোঁয়াটে লাগছে হয়ত, ধীরে ধীরে স্পষ্ট হয়ে যাবে।

একটা পার্টিশন বলতে সতত আমরা বুঝি কাঁচা ভৌত হার্ডডিস্কের একটা অংশকে। আর একটা ফাইলসিস্টেম বানিয়ে তোলা হয় এরকম এক বা একাধিক পার্টিশনকে মিলিয়ে। যেমন আমার সুজে অপারেটিং সিস্টেমের ঐক্যবদ্ধ ফাইলসিস্টেম গড়ে উঠেছে সব কটা পার্টিশনকে মিলিয়ে, যার মোট আয়তন আশি জিবি। আবার উইন্ডোজ ওএস গ্নু-লিনাক্স পার্টিশনগুলোকে দেখতে পায়না, তাই তার কাছে গোটা ঐক্যবদ্ধ ফাইলসিস্টেমটা হল শুধু দুটো পার্টিশনের, তাই উইন্ডোজ দেখায় আমার মেশিনের মোট হার্ডডিস্কভূমি মাত্র ১৮.৭ জিবি। গ্নু-লিনাক্সে এই পার্টিশনগুলোকে সরাসরি আমরা চিনি এদের ডিভাইস ফাইল দিয়ে। আর প্রতিটি পার্টিশনে তথ্য লেখার এবং রাখার ফাইলসিস্টেম আলাদা করে তৈরি করে নিতে হয়। তাই ঐক্যবদ্ধ ফাইলসিস্টেমের হায়েরার্কি স্ট্যান্ডার্ড বোঝার আগে একটু ডিভাইস এবং পার্টিশনগুলোর সম্পর্কে জানা দরকার। এই মুহূর্তে একটু হিব্রু লাগছে হয়ত, কিন্তু আজকের আলোচনা শেষ হতে হতেই আর লাগবেনা বোধহয়।

আমার মেশিনের হার্ড ডিস্কদুটোর পার্টিশনগুলোর একটা তালিকা এখানে দিচ্ছি। এই তালিকাটা একটু ভালো করে বোঝার আর মনে রাখার চেষ্টা করুন। এর পরের দিনগুলোয় বারবার এই ছকটাকে মাথায় রেখে কথা বলব আমরা। আমার মেশিনে দুটো হার্ডডিস্ক, '/dev/hda' আর '/dev/hdb'। এখন আমরা জানি এর মানে, '/dev' নামের

ডিরেক্টরিতে দুটো ডিভাইস ফাইল, যাদের নাম 'hda' আর 'hdb'। দুটোই চল্লিশ জিবির বা গিগাবাইটের হার্ডডিস্ক, কিন্তু ফরম্যাটিং-এর পর পাওয়া যায় মোটামুটি সাড়ে সাঁইত্রিশ জিবি করে, কারণ অন্যটুকু লেগে যায় তথ্য লেখার এবং রাখার কায়দা অর্থে ফাইলসিস্টেম বানাতে। হার্ডডিস্কের পার্টিশনগুলো আবার এই '/dev' ডিরেক্টরিতেই কিছু ডিভাইস ফাইল, '/dev/hda1', '/dev/hda5', '/dev/hdb2' ইত্যাদি, যাদের তালিকাটা আমরা এখনই দেখব। তালিকাটায় আছে ডিভাইস ফাইলের নাম, যা দিয়ে সিস্টেম ভৌত পার্টিশনগুলোকে চিনছে। তারপর তার সাইজ, জিবিতে বা গিগাবাইটে আর এমবিতে বা মেগাবাইটে। তারপর ফাইলসিস্টেমের নাম, চার রকম আছে মোট, আগেই বলেছি। এবং একদম ডানদিকে, মাউন্ট-পয়েন্ট। মাউন্ট-পয়েন্ট পরে আমরা আরো ভালো করে বুঝব, এখন এই সাতটা মাউন্ট-পয়েন্টকে দেখুন, একটা আন্দাজ পাবেন। আগের সেকশনের ঐক্যবদ্ধ ফাইলসিস্টেমের ছবির সঙ্গে মাউন্ট-পয়েন্টের ডিরেক্টরিগুলো মিলিয়ে নিন। এখন এইটুকু মাথায় রাখুন — মাউন্ট করা হয় একটা পার্টিশনের ভৌত ডিভাইসকে, এবং মাউন্ট করার ভূমিটা হয় একটা ডিরেক্টরি। শুধু দেখুন, সোয়াপ ফাইল বা ভারচুয়াল মেমরিতে ফাইল রাখার অস্থায়ী জায়গাটার কোনো মাউন্ট-পয়েন্ট দেওয়া হয়নি। সোয়াপ-ফাইলের পুরো ব্যাকরণটাই অন্যরকম, পরে আসব আমরা।

	পার্টিশন	সাইজ	ফাইলসিস্টেম টাইপ	মাউন্ট ভূমি
১ম হার্ডডিস্ক, চল্লিশ জিবি, '/dev/hda'	/dev/hda1 /dev/hda5 /dev/hda6	7.8 gb 10.9 gb 18.6 gb	Win95 Fat32 Win95 Fat32 Linux Reiserfs	/mnt/windows/c /mnt/windows/d /mnt/slackware
২য় হার্ডডিস্ক, চল্লিশ জিবি, '/dev/hdb'	/dev/hdb1 /dev/hdb2 /dev/hdb3 /dev/hdb5	102 mb 455 mb 11 gb 25.8 gb	Linux XFS Linux Swap Linux XFS Linux Reiserfs	/boot — / /mnt/arkive

যখন '/dev/hdb5' পার্টিশনটা মাউন্ট করা আছে, তখন আমরা '/mnt/arkive' ডিরেক্টরিতে ঢুকে 'ls' মারলে তার ফাইলগুলো দেখতে পাব — আমার সমস্ত ডাউনলোড, সমস্ত সফটওয়ার, সমস্ত এভিআই ফাইলগুলো ওখানে রাখা — বেশ লোভনীয় ডিরেক্টরি। জিএলটির গোটা সিডিটা বানানো একটা ওয়েবপেজের মত করে, সেই সিডির মোট জিনিষপত্তরও ওখানেই রাখা, 'linux.books' নামের একটা ডিরেক্টরিতে। কিন্তু। কিন্তু যখন মাউন্ট করা নেই, আপনি 'chdir' করে '/' থেকে '/mnt' থেকে '/mnt/arkive' ডিরেক্টরিতে যান। এবার 'ls' মারুন। ফক্কা। কিছুই নেই। কারণ, ওই ভৌত উপাদানটা, '/dev/hdb5' পার্টিশনটা, যাকে মাউন্ট করা হয় ওই '/mnt/arkive' ডিরেক্টরিতে, সেই পার্টিশনটা তখন ঐক্যবদ্ধ ফাইলসিস্টেমের বাইরে, তাকে মাউন্ট করা হয়নি।

দেখুন তো, এবার কি একটু শিকারের গন্ধ নাকে আসছে? একটু একটু চেনা লাগছে? অতটা আর হিব্রু লাগছে না? নাকি এখনো একটু অদ্ভুত লাগছে? অদ্ভুত লাগাটা একদমই অপরিচয়জনিত। চেনা কাউকে দিয়ে নিজের মেশিনে গ্নু-লিনাক্স ইনস্টল করে নিন। হাতের কাছে খোঁজাখুঁজি করে একদমই কাউকে না-পেলে সবচেয়ে কাছের লাগ বা জিএলটি-তে যোগাযোগ করুন। এমনকি চেষ্টা করে নিজেও করে ফেলতে পারবেন। আমাদের জিএলটি-র দুজন নিজে নিজেই করেছে প্রথমবারই। যদিও টেলিফোন করতে হয়েছে কয়েকবার, মাঝপথে নানা অসুবিধেয়। এবার ওই ইনস্টল করা সিস্টেমে সময় কাটাতে থাকুন। প্রথম প্রথম মনে হবে সবই একটা ল্যাবিরিন্থ-এর মত, গোলোকধাঁধা। তারপর, হঠাৎ একসময় সিস্টেমের যুক্তিগঠনটা আপনার মাথায় বসে যেতে শুরু করবে, দেখবেন, কুচি কুচি ম্যুরালের মত একটা ছকে বসে যেতে শুরু করেছে আপনা থেকেই। এই বসে যাওয়ার প্রক্রিয়াটা ত্বরান্বিত করার চেষ্টায় আসুন এবার একটু ডিভাইস, পার্টিশন, ডিস্ক, মাউন্ট এগুলোকে আর একটু ভালো করে জানার চেষ্টা করা যাক।

ফাইলসিস্টেমে তথ্য লেখার এবং রাখার একদম গোড়ার কয়েকটা উপাদান হল সুপারব্লক, আইনোড, ডেটা ব্লক, ডিরেক্টরি ব্লক, এবং ইনডিরেকশন ব্লক। যেগুলো এবার একটু ধরে ধরে বুঝতে হবে আমাদের। প্রাথমিক সংজ্ঞাটা জেনে রাখা যাক। সুপারব্লকে থাকে গোটা ফাইলসিস্টেম মানে ওই পার্টিশনের গোটা ডিরেক্টরি ব্যবস্থা এবং

ফাইলব্যবস্থা সম্পর্কে তথ্য। তার গঠন, তার আকার, এইসব। ঠিক কী কী তথ্য এই সুপারব্লকে থাকবে, এবং ঠিক কী ভাবে থাকবে এটা বদলায় ফাইলসিস্টেম থেকে ফাইলসিস্টেমে। ধরুন, পার্টিশনটা রাইজারএফএস হলে যেভাবে থাকবে এটা, এবং যা থাকবে সেই সুপারব্লকে, এক্সএফএস হলে সেভাবে থাকবেনা, এবং সেই তথ্যও থাকবেনা। উইন্ডোজের ফ্যাট৩২ হলে আবার আর এক রকম। ইত্যাদি।

আইনোডে থাকে একটা ফাইল সম্পর্কে যাবতীয় তথ্য, শুধু ফাইলটার নাম ছাড়া। কারণ, ফাইলসিস্টেমের কাছে নামটার কোনো মানে নেই, মানে আছে আপনার মত ব্যবহারকারীর কাছে। ডিরেক্টরি ফাইলে লেখা থাকে ফাইলের নাম এবং আইনোড নম্বর, আগেই বলেছি। ধরুন, আপনি যখন কোনো ফাইলকে টানাহেঁচড়া করছেন, কিছু পিন্ডি চটকাতে চাইছেন তার। আপনাকে শেলের তথা সিস্টেমের কাছে ফাইলটার নাম ধরে ডাকতে হচ্ছে। যেই তার নাম করছেন, ডিরেক্টরি ফাইলে সেই নামটার সঙ্গে তার আইনোড নম্বরটা পড়ে ফেলছে সিস্টেম, এবং জেনে যাচ্ছে কোন ফাইলটার কথা বলছেন। এবার এই আইনোড নম্বরটা যেই পেয়ে গেল, সেটা দিয়ে ফাইলসিস্টেমের আইনোড থেকে ফাইলটার সম্পর্কে যাবতীয় তথ্য জেনে নিচ্ছে সিস্টেম। ফাইলের তথ্যটা লেখা আছে কোন কোন ডেটা ব্লকে সেটা লেখা থাকে আইনোডে।

কিন্তু আইনোডের একটা সীমিত ভূমির সমস্যা আছে। আইনোডে মাত্র কয়েকটা ডেটা ব্লকের নম্বর লেখারই জায়গা থাকে। ফাইলটা যদি যথেষ্ট বড় হয়, ওই কটার চেয়ে বেশি ডেটা ব্লকের যদি দরকার পড়ে ফাইলের মোট তথ্যটা লিখতে, তাহলে আরো আরো ডেটা ব্লকের পয়েন্টার ফাইলের সঙ্গে যোগ করা হয় একটা গতিশীল বা ডায়নামিক রকমে। এই গতিশীল ভাবে কাজে লাগানো ডেটা ব্লকগুলোকেই বলে ইনডিরেক্ট ব্লক, যাদের নম্বর ঠিকানা ইত্যাদি লেখা থাকে ফাইলসিস্টেমের ইনডিরেকশন ব্লকে। একটু ভজোকটো লাগছে হয়ত, কেটে যাবে, বাস্তব একটা ফাইলসিস্টেমকে ধরে আলোচনা শুরু হলেই।

এই শেষ প্যারাগুলোয়, খেয়াল রাখবেন, গোটাটা জুড়েই আমরা 'ফাইলসিস্টেম' বলতে বোঝাচ্ছি একটা পার্টিশনে ফাইল আর ডিরেক্টরিতে তথ্য লেখার এবং রাখার ব্যবস্থা নিয়ে। এক বা একাধিক পার্টিশনকে মিলিয়ে তৈরি হয় একটা পূর্ণাঙ্গ ঐক্যবদ্ধ ফাইলসিস্টেমের ডিরেক্টরি কাঠামো। গ্নু-লিনাক্সে একটা পার্টিশনে গোটা ঐক্যবদ্ধ ফাইলসিস্টেম থাকেনা বললেই হয়। গ্নু-লিনাক্স নিরাপত্তা এবং কাজের ধারণার সঙ্গেই সেটা মেলেনা। আর যদি আমরা ফাইল লেখার বা রাখার একাধিক ব্যবস্থা ব্যবহার করতে চাই, এক্সএফএস রাইজার ফ্যাট৩২ গোছের, তাহলে তো সেটা ভৌত রকমেই সম্ভব না। একটা পার্টিশনে ফাইল লেখা এবং রাখার একটা কায়দাই কেবল সম্ভব। আবার একই মেশিনের পার্টিশনগুলো মিলিয়ে যদি আপনি একাধিক অপারেটিং সিস্টেম ব্যবহারের সুযোগ রাখতে চান, আমার মেশিনে যেমন, তাহলেও এক-পার্টিশনে সম্ভব নয়। স্ল্যাকওয়ারের, অন্তত বুট করার জন্যে এবং একান্ত নিজস্ব কনফিগারেশন ফাইলের জন্যে, একটা সম্পূর্ণ নিজস্ব পার্টিশন লাগবেই, ডেটার পার্টিশনগুলো সে যদি শেয়ারও করে সুজের সঙ্গে, যদিও সেটা যে অত্যন্ত কাঠালের আমস্বত্ত্ব গোছের ব্যাপার তাতে কোনো সন্দেহ নেই। আবার উইন্ডোজ তো গ্নু-লিনাক্স পার্টিশনকে দেখতেই পাবেনা, তার মানে, উইন্ডোজ থাকলে তার নিজের পার্টিশন লাগবেই। এমনকি যদি দেখতেও পেত, তাহলেও ওই বুট করার জন্যে এবং নিজের সিস্টেম কনফিগারেশন ফাইল রাখার জন্যে একটা অন্তত পার্টিশন লাগতই।

যখন আমার শুধু উইন্ডোজ ছিল তখনই, সেই আট জিবির হার্ড ডিস্কে ছিল তিনটে পার্টিশন, তার অবশ্য একটা রক্তমাংসের কারণ ছিল, যা গ্নু-লিনাক্সে অনুপস্থিত, সেটা হল ভাইরাস। মাঝে মাঝেই শুরু হত ভাইরাসের ভুতের নেত্য, তখন আর কি, বাপ বাপ বলে একটা পার্টিশন পুরো ফরম্যাট করে দাও, সমস্ত ফাইল অন্যটায় এনে। আবার যদি নিজের ভালোবাসা মাস্টার বুট রেকর্ড বা এমবিআর-এ ছড়িয়ে দেওয়ার মত রোমান্টিক ভাইরাস হয় তাহলে তো সবগুলোই ফরমাট করো। মানে কম্ম কাবাড়, গোটা হার্ডডিস্কের সমস্ত পার্টিশনের প্রতি কুচি তথ্য উপড়ে ফেলো। এরকম আবার একবারই হয়েছিল, তার আগের ফাইলগুলোর প্রিন্ট-আউট ছিল, মধ্যের তেইশদিনের কাজ চলে গেছিল ভাইরাসদের অস্ত্রে। মাঝে মাঝে মনে হয়, কোনো ক্রিয়াই তো অবশেষহীন হয়না, আর কম্পিউটারে লেখা মানেও তো একটা ভৌত ক্রিয়া — সেই ফাইলগুলো, তার তথ্যগুলো, গেল কোথায়, কোন শূন্যতায়? এটাই কি পরম ব্রহ্মে লীন হয়ে যাওয়া, যেখানে আছে, কিন্তু আর ফিরিয়ে আনা সম্ভব না?

এই যে দেখুন, দুবার বললাম 'ফরম্যাট' করার কথা, এখানে বলেছি তথ্যশূন্য করে দেওয়া অর্থে, আমরা সবাই করি, একটা কাঁচা হার্ডডিস্ক এনেই সেটাকে সরাসরি ব্যবহার করা যায়না। তাকে ফাইল রাখার, তথ্য রাখার, পরে সেই তথ্য ব্যবহার করার মত করে একটা আকার দিতে হয়, মানে একটা কঙ্কাল ফাইলসিস্টেম বানাতে হয়। কাঁচা ভৌত ভূমিটাকে শিখিয়ে পড়িয়ে নিতে হয়, সদ্য দলে আসা রাজনৈতিক কর্মীকে যেমন নেতা হওয়ার আগে, অনেক যত্নে কেয়ারে তিরষ্কারে, আগে সভ্য এবং চোর বানিয়ে তুলতে হয়। কোথায় কোন জিনিষের কী মানে এটা সে নিজেই যাতে বুঝে যেতে পারে। কাঁচা অবস্থায় যা বুঝত-না। গ্নু-লিনাক্সে এটা করা হয় মেক-ফাইলসিস্টেম (mkfs) দিয়ে। এর পুরোটারই একটা ছোট্ট অংশ হল আগে ভৌত হার্ডডিস্কটাকে পুরোনো ইলেকট্রনিক তথ্যের থেকে মুক্ত করা।

৭।। ডিভাইস, ডিভাইস-ফাইল, ডিভাইস-নাম

আমরা জানি গ্নু-লিনাক্সে সব ডিভাইসই এক একটা ফাইল যারা '/dev' ডিরেক্টরিতে বা '/dev' ডিরেক্টরির পেটের মধ্যে আরো আরো সাবডিরেক্টরিতে ছড়িয়ে থাকে। যত দিন বেড়েছে, এই '/dev' ডিরেক্টরির আকার এবং জটিলতাও বেড়েছে, কারণ প্রকৌশল বদলেছে, আরো আরো নতুন ধরনের ডিভাইস আবিষ্কার হয়েছে, তাদের নিয়ে আসা হয়েছে ব্যবহারে, অপারেটিং সিস্টেমের মধ্যে। একটা ডিভাইস খোলা, ব্যবহার করা, বন্ধ করা মানে, ওই ডিভাইসের ফাইলটাকে খোলা, পড়া, লেখা, বন্ধ করা। ডিভাইস পড়তে বললেই কী করে করতে হবে, এবং কী করতে হবে, কোথায় কোন ড্রাইভারকে কী ভাবে কাজে লাগাতে হবে, এটা কারনেল জানে, সেটা জানে বলেই সে কারনেল। দুই নম্বর দিনের আলোচনাগুলো মনে করুন। বিভিন্ন সিস্টেম ডিরেক্টরিগুলোর গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডের আলোচনায়, কোন ডিরেক্টরিতে কী থাকে এবং কেন থাকে গ্নু-লিনাক্সে তার চালু প্রথায় আসছি আমরা, তার আগে এখানে কয়েকটা ডিভাইস ফাইলের সূত্রে '/dev' ডিরেক্টরির কথা একটু বলে নেওয়া যাক। এখানে এই ডিভাইস ফাইলগুলো একটা গড় উদাহরণ, এক একটা সিস্টেমে এর থেকে একটু একটু আলাদা হতেই পারে, হয়েই থাকে। আমরা জাস্ট ছকটা বোঝার চেষ্টা করছি।

'/dev' ডিরেক্টরিতে যে ফাইলগুলো থাকে তাদের মধ্যে আছে হার্ডডিস্ক — '/dev/hd', আমরা আগেই দেখেছি। দুটো হার্ডডিস্ক থাকলে হয় '/dev/hda' এবং '/dev/hdb'। ধরুন, উইনডোজ হলে, যদি দুটো হার্ডডিস্কেই যদি গোটাটায় একটা করে পার্টিশন থাকে তাহলে হত 'C:\' এবং 'D:\'। আবার দুটো হার্ডডিস্কেই দুটো করে পার্টিশন থাকলে হত 'C:\', 'D:\', 'E:\', 'F:\'। গ্নু-লিনাক্সে সেগুলো দাঁড়াবে '/dev/hda1', '/dev/hda2', '/dev/hdb1','/dev/hdb2' ইত্যাদি। তবে, এখানে নম্বরগুলো বদলেও যেতে পারে, '/dev/hda2' না হয়ে '/dev/had5'-ও হতে পারে। কেন হতে পারে সেটা বোঝার একটু দেরি আছে, প্রাইমারি, লজিকাল এবং এক্সটেন্ডেড পার্টিশন তার আগে বুঝে নিতে হবে। '/dev/hda' এবং '/dev/hdb' দুটোই আইডিই বা ইন্টিগ্রেটেড-ডিভাইস-ইলেকট্রনিক্স হার্ডডিস্ক। আগেই বলেছি, আইডিই মানে এক ধরনের ডিস্ক-ড্রাইভ যাতে ডিভাইসের ড্রাইভারটা দেওয়া থাকে ড্রাইভের মধ্যেই, আলাদা করে অ্যাডাপ্টার কার্ড আর লাগাতে হয়না। এর থেকে উন্নত এবং তাই দামী হার্ডডিস্ক হল স্কাসি, স্মল-কম্পিউটার সিস্টেম-ইন্টারফেস। আইডিই হার্ডডিস্কের থেকে অনেক দ্রুত গতিতে তথ্য পড়া এবং লেখা যায় স্কাসি-ড্রাইভে। একটা বিশেষ ধরনের ব্যাবস্থা এই স্কাসি। এতে শুধু হার্ডডিস্ক নয়, অন্য পেরিফেরালও হয়। স্কাসি হার্ডডিস্ক প্রথমটার নাম হয় '/dev/sda', দ্বিতীয়টার নাম হয় '/dev/sdb'। আইডিই বা স্কাসি দুরকম হার্ডডিস্কেই একাধিক করে পার্টিশন থাকতে পারে। '/dev/hdb1' মানে দ্বিতীয় আইডিই হার্ড ডিস্কের এক নম্বর পার্টিশন, '/dev/sda3' মানে প্রথম স্কাসি হার্ড ডিস্কের তিন নম্বর পার্টিশন, ইত্যাদি।

এই যে বারবার বলছি না, প্রথম হার্ডডিস্কের নাম '/dev/hda' বা '/dev/sda' এবং দ্বিতীয়টার নাম '/dev/hdb' বা '/dev/sdb', এই 'প্রথম' আর 'দ্বিতীয়' মানে কী? আইডিই ড্রাইভের বেলায় এর মানে প্রাইমারি মাস্টার আর সেকেন্ডারি মাস্টার। তাদের ডিভাইস ফাইলের নাম হয় '/dev/hda' আর '/dev/hdb'। সেকেন্ডারি মাস্টার আর সেকেন্ডারি স্লেভ হলে '/dev/hdc' আর '/dev/hdd'। এই প্রাইমারি আর সেকেন্ডারি, মাস্টার আর স্লেভ — এগুলো গোটাটাই মেশিনের হার্ডওয়ার সংযোগে কারনেল কার পরে কাকে খুঁজবে তার একটা ক্রম। জিএলটির জন্যে অনেকের মেশিনে গ্নু-লিনাক্স ইনস্টল করতে গিয়ে এটায় সমস্যায় পড়েছি বেশ কয়েকবার। যদি হার্ডডিস্ক এবং সিডিড্রাইভ ঠিক ঠিক জায়গায় সংযুক্ত না-থাকে, তাহলে উইন্ডোজ চলে, কিন্তু গ্নু-লিনাক্স প্রচণ্ড সমস্যা করে। বড়

সামর্থকে কাজে লাগাতে গেলে আপনাকে বড় মনোযোগ দিতেই হবে। যেমন, প্রাইমারি মাস্টার বা '/dev/hda'-তে গ্নু-লিনাক্স যদি হার্ডডিস্ক না-পায়, তাহলে বুটলোডার বা 'lilo' ইনস্টল করতে পারেনা, তখন আপনাকে আলাদা করে বলে দিতে হবে, কোথায় 'lilo' ইনস্টল করতে হবে। কারণ মাস্টার-বুট-রেকর্ড (MBR) খোঁজার কথা প্রথম হার্ডডিস্কের প্রথম মানে বুট সেক্টরেই — আগে বলেছি, পরে আরো ভালো করে বুঝবেন এগুলো। এমনকি, ভুলভাল রকমে অ্যাসেম্বল করা হলে, ধরুন, সেকেন্ডারি মাস্টার স্লটটা ফাঁকা আছে, অথচ সিডিড্রাইভটা সেকেন্ডারি স্লেভে ঢোকানো, এই অবস্থাতেও খুব অশান্তি হয়। গ্নু-লিনাক্স যেমন চূড়ান্ত ভাবে কাজে লাগাতে পারে হার্ডওয়ারকে, তেমনি তার কনফিগারেশনও সঠিক হতে হয়। এই সমস্যাগুলো এমন কিছু নয়, দু-একবার ড্রাইভ খুলে তার গায়ের জাম্পার সেটিং দেখে নিয়ে আপনি নিজেই পারবেন, তারপর সেটা বায়োসে জানিয়ে নিতে হয়। তবে এই গোটা আলোচনাটাই করলাম আইডিই ড্রাইভের জন্যে। স্কাসি আমি নিজে কোনোদিন ব্যবহার করিনি, তাই বলতে পারব না। তবে নামের মিল দেখে মনে হচ্ছে, কাজেরও মিল থাকবে বোধহয়। এখানে একটা মজার তথ্য উল্লেখ না করে পারছিনা। লস-অ্যাঞ্জেলিসের কর্তারা ঘোষণা করেছেন, এই 'মাস্টার' আর 'স্লেভ' শব্দদুটো যন্ত্রাংশের বিষয়ে আর ব্যবহার করা যাবেনা। ন্যাম থেকে ইরাক অব্দি ইতিহাসের প্রতিটি স্তরে প্রভু-ভৃত্য সম্পর্ক বিলোপের ব্যাপক মার্কিন অবদানের পরেও যন্ত্রের জগতের লুপ্ত মেটাফরে তাদের এভাবে টেনে নিয়ে চলায় বাবুদের বড় বেদনা হল। পুরো মজাটা পেতে চাইলে সিএনএন-এর সাইটে গিয়ে 'মাস্টার স্লেভ' দিয়ে সার্চ দিন।

সিডিড্রাইভ একটা হলে সতত সেটা রাখা হয়সেকেন্ডারি মাস্টারে, দুটো হলে সেকেন্ডারি মাস্টার আর স্লেভ স্লটে। যা বললাম, যদি অন্য কিছু করেন, কনফিগারেশন সেই অনুযায়ী বদলে নিতে হয়। সিস্টেম ডিভাইসে এরা দাঁড়ায় '/dev/hdc' আর '/dev/hdd'। আইডিই ডিভাইস হলে সচরাচর এই ডিভাইস ফাইল দুটোকে লিংক করা থাকে '/dev/cdrom0' আর '/dev/cdrom1' নামের আরো দুটো ডিভাইস ফাইলে। শূন্য মানে প্রাইমারি, এক মানে স্লেভ। এটা করা হয় মূলত ব্যবহারযোগ্যতা বাড়াতে, আপনার চেনা নামে আপনি যাতে ডিভাইসগুলোকে পেয়ে যেতে পারেন। লিংক নিয়ে আলোচনার পর এটা স্পষ্ট হবে। স্কাসি সিডিড্রাইভ হলে তখন নাম হয় '/dev/scd0' এবং '/dev/scd1'। শুধু হার্ড ডিস্ক নয়, সিডি রম, মেমরি এরকম অনেক কিছুই স্কাসি হতে পারে।

মেশিনের বাইরের বাস্তবতার সঙ্গে মেশিনের তথ্য যোগাযোগ ঘটে পোর্ট দিয়ে, বলেছি আগেই। গ্নু-লিনাক্সে এই পোর্টগুলোও ডিভাইস ফাইল। উইনডোজে যাকে কমিউনিকেশন পোর্ট কম-ওয়ান (COM1) বলে, গ্নু-লিনাক্স তাকে দেখে '/dev/ttys0' ফাইল হিশেবে। মানে প্রথম সিরিয়াল পোর্ট। সচরাচর এতে মোডেম লাগানো থাকে। তখন '/dev/ttys0' আর '/dev/modem' নামের ডিভাইস ফাইল দুটো পরস্পর লিংক করা থাকে। মোদ্দা কথায় 'লিংক' মানে সিস্টেম একটাকে পেলেই অন্যটাকে বোঝে। '/dev/modem' থেকে আমরা তথ্য পড়ছি মানে '/dev/ttys0' পোর্টে যুক্ত মোডেম থেকে কোনো মেল বা ওয়েবপেজ নামাচ্ছি মেশিনে। ইউএসবি (USB — **U**niversal-**S**erial-**B**us) ডিভাইসগুলোর ফাইলগুলো থাকে '/dev' ডিরেক্টরির পেটে '/dev/usb' সাবডিরেক্টরিতে। ইউনিভার্সাল সিরিয়াল বাস হল একটা উন্নত বাস-ব্যবস্থা। বাস মানে তো আমরা জানি, পেরিফেরালগুলোর সঙ্গে যোগাযোগপথ। ইউএসবি '৯৮-এর পর থেকে বহুল ব্যবহৃত হচ্ছে। আবার, মাউসের জন্যে অনেকসময় যেটা ব্যবহার করি সেই পিএসবাইটু (PS/2) পোর্টের ডিভাইস ফাইল হয় '/dev/psaux'। এই 'পিএসটু' নামটা এসেছে আইবিএম-এর একটা পিসি মডেলের নাম থেকে, পিসিটা খুব একটা চলেনি, অন্য ধরনের একটা নির্মাণের কারণে, কিন্তু নামটা রয়ে গেছে। পিএসবাইটু পোর্টে মাউস, মোডেম ইত্যাদি ব্যবহার হয়। ফ্লপিড্রাইভের ডিভাইস ফাইল হয় সচরাচর '/dev/fd0', যাকে অনেকসময় লিংক করা থাকে '/dev/floppy' ফাইলে। '/dev/dsp' মানে তো বলেছি, সাউন্ড কার্ড বা প্রথম সাউন্ড ডিভাইস। '/dev/lp0' মানে প্রিন্টার পোর্ট বা উইন্ডোজের এলপিটি-ওয়ান (LPT1)। এই যে নামকরণগুলো লিখলাম, ডিরেক্টরিতে দেখবেন, ডিস্ট্রো থেকে ডিস্ট্রোতে বদলায়। তাই এদের মনে রাখার চেষ্টা করবেন না, বোঝার চেষ্টা করুন।

ভৌত ডিভাইসগুলোর দুটো ভাগ ব্লক ডিভাইস আর ক্যারেকটার ডিভাইসকে মেনে ডিভাইস ফাইলগুলোও ব্লক ডিভাইস ফাইল আর ক্যারেকটার ডিভাইস ফাইল এই দুই রকমের হয়। যেমন, ফ্লপি, হার্ডডিস্ক, সিডিড্রাইভ — এগুলো ব্লক ডিভাইস। আবার সিরিয়াল পোর্ট, মাউস, প্যারালাল প্রিন্টার পোর্ট — এগুলো ক্যারেকটার ডিভাইস। দুই নম্বর দিন থেকে মনে করুন, ব্লক ডিভাইসে তথ্য লেখা এবং পড়াটা হয় ব্লকের এককে। একটা প্রোগ্রাম যখন

একটা ফাইলকে লিখতে চায় একটা ব্লক ডিভাইসে, সে আবেদন জানায় কারনেলের কাছে, কারনেল তখন ডিভাইস ড্রাইভারকে কাজে লাগিয়ে এক এক তাল তথ্য, মানে এক এক ব্লক করে, লিখতে থাকে ডিস্কের শরীরে ওই ফাইলে। প্রতিটি ব্লকের একটা নিজের নম্বর আছে, মনে পড়ছে? আবার ডিস্ক থেকে একটা ফাইল পড়ছি যখন, প্রথমে খোঁজা হয় বাফার-ক্যাশে, মানে সেই তুলে রাখা অস্থায়ী স্মৃতি, সেখানে সাম্প্রতিকতম অতীতে ব্যবহৃত তথ্যের মধ্যে পাওয়া যায় কিনা। যদি মিলল তো গুড, আর ডিস্ক নাড়াচাড়া করতে হলনা, অনেকটা সময় বেঁচে গেল, একটু আগে গ্নু-লিনাক্স ব্যবস্থার মেমরি ব্যবহারের যে বাড়তি ওস্তাদিটার কথা বলছিলাম।

'/dev' ডিরেক্টরিতে গিয়ে যদি '`ls -l -R`' কমান্ড দিই (দুটো অপশানকে জুড়ে '`ls -lR`' কমান্ডও দেওয়া যেত), একটা বিরাট সময় ধরে অজস্র ফাইলের দীর্ঘ তালিকা স্ক্রিন জুড়ে দেখিয়ে চলে সিস্টেম। প্রতিটি ফাইলের খুঁটিনাটি। '`l`' অপশানটা বলে ফাইলের সমস্ত প্রয়োজনীয় খুঁটিনাটিকে দীর্ঘ আকারে ফুটিয়ে তুলতে, আর '`R`' অপশানটা বলে ওই ডিরেক্টরির মধ্যে নিম্নতম সাবডিরেক্টরির ফাইলগুলোও দেখাতে, মনে পড়ছে? এই শেষহীন তালিকাটাকে রিডাইরেক্ট করে বানানো ফাইল থেকে কয়েকটা লাইন তুলে দিই, মোট লাইন এগারো হাজার। আসলে '/dev' ডিরেক্টরিতে সব রকম সম্ভাবনার জন্যেই ব্যবহারযোগ্য ফাইল বানানো থাকে । আটখানা হার্ডডিস্কের প্রতিটায় যাতে একত্রিশটা অব্দি পার্টিশন বানানো যায়, 'hda' থেকে 'hdh', 'hda1' থেকে 'hda31', আরো বারোখানা হার্ডডিস্কে যাতে পনেরোটা অব্দি পার্টিশন বানানো যায়, 'hdi' থেকে 'hdt', 'hdi1' থেকে 'hdi15', ইত্যাদি। এরকম সব ডিভাইসের জন্যেই। এখানে আমার বানানো ফাইলও আছে, তাই নিজের সিস্টেমের সঙ্গে মেলাতে যাবেন না।

```
lrwxrwxrwx  1 root  root       10 2003-12-05 14:53 modem -> /dev/ttyS0
crw-rw----  1 root  uucp   4,  64 2003-03-14 18:37 ttyS0
crw-rw--w-  1 root  lp     6,   0 2003-03-14 18:37 lp0
brw-rw----  1 root  disk   3,   0 2003-03-14 18:37 hda
brw-rw----  1 root  disk   3,   1 2003-03-14 18:37 hda1
brw-rw----  1 root  disk   3,  64 2003-03-14 18:37 hdb
brw-rw----  1 root  disk   3,  65 2003-03-14 18:37 hdb1
lrwxrwxrwx  1 root  root       10 2003-12-05 07:09 mouse -> /dev/psaux
crw-rw----  1 root  root   10,  1 2003-12-17 11:17 psaux
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdrom -> /dev/sr1
lrwxrwxrwx  1 root  root        8 2003-12-05 07:38 cdwri -> /dev/sr0
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 sr0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 sr1
brw-r--r--  1 root  disk   11,  0 2003-03-14 18:37 scd0
brw-r--r--  1 root  disk   11,  1 2003-03-14 18:37 scd1
```

এবার দেখুন তো, কয়েকটা ফাইলের এই লং লিস্টিং থেকে আপনি কিছু আন্দাজ করতে পারছেন? ঠিক আগের আলোচনায় যা বলে এলাম? এই প্রতিটি ফাইলের নামের আগেই সিস্টেমের কাছে তার অ্যাবসলিউট বা চূড়ান্ত ঠিকানা '/dev/' যোগ করে নিন, কারণ, এই 'ls' কমান্ডটা তো আমরা দিয়েছি '/dev' ডিরেক্টরিতে দাঁড়িয়েই। শুধু একটা জিনিষ মনে রাখুন, লাইনের একদম গোড়ায় যে 'b' বা 'c' — তার মানে ব্লক ডিভাইস বা ক্যারেকটার ডিভাইস, আর 'l' মানে লিংক, ওই যে সংযোগের কথা বললাম, পরে আরো ভালো করে জানব। এবার একদম ডানদিকের ডিভাইসের নামের সঙ্গে মেলাতে পারছেন? চারটে ডিভাইসের নামের পরে দেখুন '->' চিহ্ন দিয়ে অন্য একটা ডিভাইস দেখানো, এর মানে লিংক। যেমন '/dev/psaux' ডিভাইসের মুখোশ হল '/dev/mouse'। কোনো প্রোগ্রাম '/dev/mouse' বললে সিস্টেম আসলে '/dev/psaux' বুঝবে। আমরা '`cat /dev/mouse`' কমান্ড দিয়ে মাউসের নড়াচড়ার টেক্সট সংকেত স্ক্রিনে দেখেছিলাম, মনে পড়ছে? এই মেশিনে মাউস ডিভাইসের ফাইলটা '/dev/psaux' হল কারণ এটা পিএসবাইটু মাউস। এই ডিভাইস ফাইলগুলো এবং বাস্তব ভৌত উপাদানগুলো সঠিক অর্থেই তুল্যমূল্য, আমরা আগেই, বিভিন্ন ডিভাইস ফাইলগুলো নিয়ে মজা করার সূত্রে দেখিয়েছি, কী ভাবে একদম ভৌত উপাদানগুলোর মতই এই ডিভাইসগুলোয় তথ্য লেখা যায়, বা এখান থেকে তথ্য পড়া যায় — মাউস-চলনের ফরমুলা বা কারনেলের কণ্ঠস্বর তারই প্রমাণ।

ডিভাইসদের বোঝার একটা উপায় মেজর আর মাইনর নাম্বার, মুখ্য আর গৌণ সংখ্যা। মেজর নাম্বার দিয়ে বোঝায় ওই ধরনের ডিভাইসের বর্গ, আর মাইনর নাম্বার ডিভাইসটার নিজস্ব। ডিভাইসটা মাস্টার না স্লেভ, বা কত নম্বর

পার্টিশন এগুলো বোঝায় এই গৌণ সংখ্যা দিয়ে। ফাইলের ওই দীর্ঘ তালিকাটায় দেখুন পাঁচ নম্বর স্তম্ভটা হল মেজর আর মাইনর নাম্বারের। যেমন ‘hda1’ ডিভাইসের সংখ্যাদুটো হল ‘3, 1’, আবার ‘hdb1’ ডিভাইসের সংখ্যাদুটো ‘3, 65’। একই ডিভাইস কন্ট্রোলার তাদের দুজনকেই নিয়ন্ত্রণ করে বলে এদের দুজনেরই মেজর নাম্বার সমান। এই জায়গাটা নিয়ে আমার আর না-এগোনোই ভালো, আমি নিজেও এইরকম ভাসাভাসা ভাবেই জানি।

পরের দিন আমরা শুরু করব ফাইলের অনুমতি মালিকানা নিরাপত্তা নিয়ে। তারপর যাব একটা ভৌত পার্টিশনে কী করে ফাইল সিস্টেম লেখা থাকে সেই কথায়। তারপর আসবে গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি। এখন লেখাটা এগোচ্ছে বিরক্তিকর শ্লথগতিতে। কলেজ চলছে। কতটুকু আর সময় পাচ্ছি লেখার? আজ যা যা লেখার প্ল্যান ছিল তার অনেকটাই হলনা। পরের দিন হবে। নিজেকেও অনেক পড়তে হচ্ছে। জিএলটি ইশকুলের ক্লাসের চেয়ে এখানে পরিশ্রম বেশি হচ্ছে। বলার সময়ে চট করে বলে চলে যাওয়া যায়, নিজেরও খেয়াল পড়েনা, দুটো চারটে করে ডিটেইলস-এর ফাঁক রয়ে যায়, নিজে যেভাবে ভাবি সেভাবেই বলি। কিন্তু এখানে লিখতে গিয়ে অনেক খুচরো পাপের প্রায়শ্চিত্য করতে হচ্ছে। আমার মাথায় নয় কৌতূহলটা আসেনি, অন্য আর একজনের তো আসতেই পারে। গ্নু-লিনাক্সে, এবং তার কমান্ড মোডেই, উইনডোজ-এর মত গুই দিয়ে নয়, আপনার সক্রিয়তা — সেটাই চায় এই পাঠমালা। কৌতূহল না-মিটলে তো সক্রিয়তা তৈরি হয়না। শেখানোটা বড় কথা নয়, সেই কাজে আমি খুব দড়ও নই, নিজেই তো শিখছি একটু একটু করে। বড় কথা আপনার চিন্তা আর কৌতূহলকে উত্তেজিত করা। সেটা পারছি কিনা কে জানে? লিখি তো সবসময়েই, কিন্তু এই লেখাটা আমার কাছেও খুব ভিন্ন — নিজের এলাকার বাইরে একটা টেকনিকাল আন্দোলনের গতিটাকে হাজির করা আমারই মত অটেকনিকাল মানুষদের কাছে। যারা ঠিকমত টেকনিকাল রকমে কম্পিউটার শেখেন তাদের জন্যে এই পাঠমালাটা নয়, আগেই বলেছি।

glt-mad@ilug-cal.org