জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

ছয় নম্বর দিনে ফাইল সম্পর্কে ধারণাটা একটু তৈরি হল। নানা রকম ফাইল। তাদের বৈশিষ্ট্য। এদের সম্মিলিত আকারই ফাইলব্যবস্থা। আজ এই ফাইলব্যবস্থা নিয়ে নানা জিনিষ আলোচনা করব। ফাইলের অনুমতি মালিকানার সম্পত্তি-সম্পর্ক নিয়ে কথা বলার পর যাব একটা ভৌত পার্টিশনে ফাইল সিস্টেম গড়ে ওঠার ব্যবস্থাগুলো নিয়ে। তারপর যাব ফাইলসিস্টেম নিয়ে আমাদের শেষ প্রসঙ্গ, গ্নু-লিনাক্স কাঠামোয় ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড।

।। দিন সাত।।

১।। ফাইলের সম্পত্তিসম্পর্ক

এক নম্বর দিনে যখন লিখছিলাম, তখন কিছুতেই বুঝতে পারছিলাম না অনভ্যস্ত একজন কাউকে আমি সুপারইউজার আর ইউজার-এর ধারণাটা কী করে বোঝাব। মনে আছে, কারনেল আর অ্যাপ্লিকেশন এই দুটো স্তরের উল্লেখ ছিল? কারনেল স্তরে খাপ খুলতে পারে কেবল রুট বা সুপারইউজার, আর ইউজার কাজ করে একদম উপরের প্রয়োগ বা অ্যাপ্লিকেশন সফটওয়ারের স্তরে। সেই অবস্থাটা আমরা পেরিয়ে এসেছি। পার্থক্যটার একটা প্রাথমিক ধারণা এসেছে। এখন আমরা জানি ইউজার বলতে কী বোঝায়। এই পাঠমালায় বারবার উদাহরণগুলো দিচ্ছি আমার নিজের মেশিনের সিস্টেম থেকে, কিছু অপ্রয়োজনীয় জটিলতা ছেঁটে। এই মেশিনের গ্নু-লিনাক্স ব্যবস্থায় যেমন ইউজার হল ‘atithi’, ‘manu’, ‘piu’, এবং ‘dd’। সিস্টেমের হোম ডিরেক্টরি ‘/home’-এর ভিতরে তাদের নিজের বাড়ি বা হোমগুলো যথাক্রমে ‘/home/atithi’, ‘/home/manu’, ‘/home/piu’, এবং ‘/home/dd’। পাঁচ নম্বর দিনের ‘/etc/passwd’ ফাইল থেকে মিলিয়ে নিন। আর সুপারইউজার ‘root’ , তার হোম ডিরেক্টরি ‘/root’। আগেই বলেছি, রুট শব্দটা দুটো অর্থেই ব্যবহার হয় গ্নু-লিনাক্সে। রুট ডিরেক্টরি বা ‘/’, আর রুট ইউজারের হোম ডিরেক্টরি মানে ‘/root’। এই রুট গ্নু-লিনাক্স সিস্টেমে সর্বশক্তিমান, যা কাজ করা যায় তার প্রত্যেকটাতেই তার অপ্রতিহত অধিকার।

কিন্তু সুপারইজার নয় অপ্রতিহতগতি, ইউজারেরও বেশ কিছু অধিকার আছে গোটা সিস্টেমে, নইলে সে কাজ করবে কী করে? যখনি সে কোনো প্রয়োগমূলক সফটওয়ার বা ইউজার অ্যাপ্লিকেশন চালাবে, তখনি তাকে রসদ ব্যবহার করতে হবে। একটা কুচোতম কাজ ভাবুন। একটা অ্যাপ্লিকেশন প্রোগ্রাম খুললেন, ধরুন একটা টেক্সট এডিটর। সে আগে লিখে রাখা একটা টেক্সট ফাইলকে অন্য এক জায়গায় কপি করল। এই কাজটার মধ্যে রসদ ব্যবহারের কতগুলো আলাদা ধাপ আছে সেটা খেয়াল করুন। প্রোগ্রামটা চালালেন। একটা প্রোগ্রামের একবার চলা মানে একটা প্রসেস। প্রোগ্রাম ফাইলটা যেখানে লেখা আছে, ডিস্কের সেই স্থির মেমরি থেকে প্রোগ্রাম ফাইলের একটা কপি তুলে ফেলা হল জ্যান্ত ভারচুয়াল মেমরিতে, প্রতিমুহূর্তে যেটা বদলাচ্ছে, সিস্টেম সরাসরি তার দেখভাল করে। তার মানে ইতিমধ্যেই ডিস্ক এবং র‍্যাম এবং ভারচুয়াল মেমরি বা ডিস্কের গায়ে অস্থায়ী ফাইলের রসদ ব্যবহার হয়ে গেল। এবার সেটাকে কপি করা হল, আবার মেমরি। গেল মেশিনের যাতায়াতপথ বা বাস বেয়ে। তাকে লেখা মানে আবার মেমরি এবং আবার ডিস্ক। আর সিপিইউ গোটা সময়টা ধরেই ব্যবহার হচ্ছে। কতবার কতগুলো রসদকে ব্যবহার করতে হচ্ছে এই সরলতম কাজটুকু করতে গিয়েও, সেটা খেয়াল করলেন।

কিন্তু প্রোগ্রামটা চালাল যে সে তো ইউজার। তার তো রসদের উপর কোনো অধিকার নেই। তার হয়ে প্রতিটি রসদ আসলে ব্যবহার করে দিল কারনেল। কিন্তু কারনেল তাকে করতে দিল কেন? কারণ, ওই প্রোগ্রাম চালানোর অধিকার তার আছে। সেই প্রোগ্রামটার আবার অধিকার আছে মেশিনের রসদগুলো ব্যবহারের। সেখানেও অনেকগুলো সীমা আছে। কোন ইউজার কোন প্রোগ্রাম ব্যবহার করতে পারবে, কোন কোন রসদ কতটা ব্যবহার করতে পারবে। যেমন, ওই প্রোগ্রামটা যদি সাধারণ বাইনারির ডিরেক্টরি ‘/bin’-এ না-থেকে সিস্টেম বাইনারির ডিরেক্টরি ‘/sbin’-এ থাকত, তাহলে একজন সাধারণ ইউজার প্রোগ্রামটা চালাতে পারত না, যদি না তাকে বিশেষ অনুমতি দেওয়া থাকে বিশেষ কোনো সিস্টেম প্রোগ্রাম চালানোর। আবার সাধারণ একজন ইউজার তার কপি করা টেক্সট ফাইলটাকে কখনোই তার

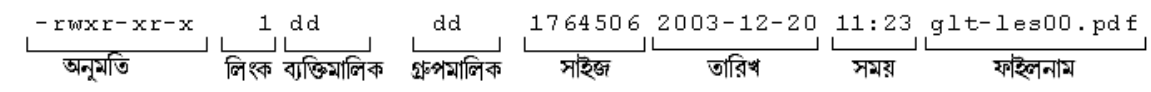
নিজের হোমের বাইরে সেভ করতে পারতনা। নিজের অধিকারের প্রোগ্রামটা চালানোর সময়ে, কোন রসদটা একজন সাধারণ ইউজার ব্যবহার করতে পারবে, এবং কতদূর পারবে, সেটাকেই নিয়ন্ত্রণ করে কারনেল। সাধারণ ইউজারদের এই সাধ্য এবং অধিকারের মধ্যে আবার নানারকম প্রকারভেদ আছে, নানা গ্রুপ আছে ইউজারদের। একজন ব্যবহারকারী যতগুলো খুশি গ্রুপেরই সভ্য হতে পারে, প্রাথমিকভাবে কোনো সিস্টেমে কোনো নতুন ইউজার তৈরি হওয়ামাত্র সে অন্তত গ্রুপের একটা মেম্বার হয়ই, তার নিজের নামে যে গ্রুপের নাম, এবং একজনই সভ্য সেই গ্রুপের। এটা সচরাচর সিস্টেমের ডিফল্ট সেটিং, তারপরে যা খুশি তো বদলে নেওয়াই যায়। কোন কোন ফাইলের উপর কী কী অধিকার থাকবে একজন ইউজারের, তা প্রাথমিকভাবে অনেকটাই নির্ভর করে, সেই ইউজার কোন কোন গ্রুপের সভ্য তার উপর। এই ফাইলের উপর অধিকারের একটা রকম হল চালানোর অধিকার, মানে কোন প্রোগ্রাম কে চালাতে পারবে, যা ঠিক করে দেয় কোন প্রোগ্রাম কোন ইউজার চালাতে পারবে। এখুনি ক্লিয়ার হয়ে যাবে গোটাটা।

১.১।। ফাইলের মালিকানা এবং অনুমতি

যখনই কোনো ফাইল তৈরি হয় একটা সিস্টেমে, সে যে ধরনের ফাইলই হোক, সেই ফাইলের একটা ব্যক্তি মালিক এবং একটা গ্রুপ মালিক ঘোষিত হয়ে যায়। এই ঘোষণাটা সিস্টেম দেগে দেয় সিস্টেমে তুলে রাখা ফাইলের খুঁটিনাটিতে। যেকোনো ফাইলের উপর '`ls -al`' মেরে যেটা দেখা যায়। যেমন '/home/dd' ডিরেক্টরির ভিতর একটা সাবডিরেক্টরিতে একটা '`ls -al`' কমান্ডের ফলাফল এখানে তুলে দেওয়া যাক, '/dev' ডিরেক্টরির উপর '`ls -al`' কমান্ডের ফলাফলের একটা অংশ আগের সেকশনেই আমরা দিয়েছিলাম।

```
-rwxr-xr-x    1 dd     dd      1764506 2003-12-20 11:23 glt-les00.pdf
-rwxr-xr-x    1 dd     dd       227703 2003-12-20 11:23 glt-les01.pdf
-rwxr-xr-x    1 dd     dd       202997 2003-12-20 11:23 glt-les02.pdf
-rw-r--r--    1 root   root        294 2003-12-20 11:24 ld.so.conf
-rw-------    1 root   root        637 2003-12-20 11:24 lilo.conf
-rw-r--r--    1 root   root      13227 2003-12-20 11:24 modules.conf
```

'`ls -al`' কমান্ডটা ব্যবহার করায় স্ক্রিনে ফাইলের তথ্যরাজি ফুটিয়ে তোলার যে ছকটা ব্যবহার করা হয়েছে, সেটা একটু খুঁটিয়ে বোঝা যাক। 'glt-les00.pdf' মানে তালিকার প্রথম ফাইলটাকে ধরুন। প্রথমে মালিকানা দেখানো হয়েছে '`-rwxr-xr-x`' অংশটায়। তারপর '`1`' অংশটায় দেখানো হয়েছে এর লিংক বা সংযোগের সংখ্যা। তারপর '`dd`' হল এর ব্যক্তি মালিক, মানে ইউজার 'dd'। তারপরের '`dd`' মানে, ফাইলটার গ্রুপ-মালিক হল 'dd' গ্রুপ। এরপর এর সাইজ, বাইটে, '`1764506`', একে মেগাবাইটে বদলে নিলে সাইজটা দাঁড়াচ্ছে ১.৬৮ এমবি। তারপর ফাইলটাকে শেষ বদলানোর তারিখ '`2003-12-20`', সময় '`11:23`'। শেষে ফাইলের নাম '`glt-les00.pdf`'। এর মধ্যে অন্যগুলো তো জলের মত বুঝতে পারছেন, শুধু অনুমতি আর লিংকে এখনো একটু ব্যথা আছে, সেটা সমাধান হয়ে যাবে এই সেকশনেই।



ফাইলের উপর বিভিন্ন ধরনের অনুমতির কাঠামোটা দেখানো হয়েছে দশটা হাইফেন দিয়ে, '`----------`'। দশটা হাইফেন মানে দশটা শূন্যস্থান। নানা অক্ষর দিয়ে পুরন করা হয় এদের। আর কোনো অক্ষরই না-থাকা মানে হল হাইফেন। দশটা শূন্যস্থানের মধ্যে আবার প্রকারভেদ আছে। আসলে দশটা হল ১ + ৩ + ৩ + ৩। একদম বাঁদিকের এক নম্বর হাইফেনটা ফাইলের চরিত্র জানায়। তালিকার ফাইলটা ডিরেক্টরি হলে '`d`', রেগুলার ফাইল হলে এই হাইফেনটা শূন্য থাকবে। যেমন এই তালিকার সবকটা ফাইলেই তাই আছে। এরা সকলেই রেগুলার ফাইল। নিজের সিস্টেমের কোনো ডিরেক্টরির মধ্যে '`ls -al`' মেরে দেখুন, মধ্যের ডিরেক্টরিগুলোয় '`d`' দেখাচ্ছে কিনা। ফাইলটা যদি ক্যারেকটার ডিভাইস ফাইল হয়, এখানে আসবে '`c`'। ব্লক ডিভাইস ফাইল হলে আসবে '`b`'। লিংক ফাইল হলে আসবে '`l`'। ছয় নম্বর দিনের মানে আগের দিনের ৭ নম্বর সেকশনের ডিভাইস ফাইলের তালিকা থেকে মিলিয়ে নিন এগুলো।

দশ হাইফেনের রইল বাকি তিন তিরিক্কে নয়। তিনটে ড্যাশের তিনটে গ্রুপ, '--- --- ---'। প্রথম তিনটে হাইফেন ইউজারের নিজের। পরের তিনটে তার গ্রুপের। শেষ তিনটে অন্যদের। এবং প্রতি তিনটে হাইফেনের প্রথমটায় আসতে পারে 'r', তার মানে পড়ার বা রিড করার অনুমতি আছে, আর শূন্য মানে হাইফেন আছে মানে অনুমতি নেই। দ্বিতীয় হাইফেনটায় 'w' থাকা মানে লেখার বা রাইট করার অনুমতি আছে, হাইফেন মানে লেখার অনুমতি নেই। শেষ হাইফেনটায় 'x' মানে চালানোর বা একজিকিউট করার অনুমতি আছে, হাইফেন মানে অনুমতি নেই। এই নয়টা হাইফেনের সবকটাই শূন্য মানে ফাইলটার উপর কারো কোনো অনুমতি নেই। না ইউজারের, না তার গ্রুপের, না অন্যদের। না পড়ার, না লেখার, না চালানোর। মানে, ফাইলটা লিটারালি শিবঠাকুরের আপনদেশের ফাইল, সর্বনেশে এর আইনকানুন, ফাইলটার যে মালিক সে নিজেও ফাইলটা পড়তে বদলাতে বা চালাতে পারবেনা, শুধু ফাইলমালিকের পূর্ণ অধিকারে স্ক্রিনে ফাইলটার নাম আর অনুমতির তালিকার দিকে নির্নিমেষ চোখে চেয়ে থাকতে পারবে, তার উপরে গ্নু-লিনাক্সে কোনো বারণ নেই। সদ্য বেরোনো কারনেল ভার্শন ২.৬ অব্দি অন্তত নেই। ওই চেয়েই থাকতে হবে তাকে, যতক্ষণ না সে 'chmod' বলে একটা কমান্ড দিয়ে ফাইলটার এইসব বেলেল্লাপনাকে শায়েস্তা করছে, সেই কথায় আসছি, এর পরের সেকশনেই। তবে সেই কমান্ড না-জেনে নিজে নিজে এইরকম একটা ফাইলের মালিক হওয়ার কোনো উপায় আমি জানিনা। ফাইলের এই বিচিত্র আচরণ ঘটানো যায় একমাত্র ওই কমান্ডটা দিয়েই। আর একটা ফাইলের অনুমতি তালিকাটা যদি এই রকমের হয়, 'rwxrwxrwx', তার মানে, এই ফাইলটার উপর ব্যবহারকারীর নিজের, তার গ্রুপের এবং অন্যদের প্রত্যেকের ফাইল পড়ার, লেখার এবং চালানোর প্রতিটি অনুমতিই আছে। সম্পূর্ণ অবাধ সামগ্রিক অপ্রশ্নেয় গনতন্ত্র।

এবার দেখুন তো, আপনি একটু আগের তালিকাটা থেকে, বা আগের সেকশনের তালিকাটা থেকে, এই অনুমতি ব্যাপারটা পরিষ্কার বুঝে যাবেন, হাইফেন থাকা মানে অনুমতিটা নেই, অক্ষর থাকা মানে আছে, প্রথম তিনটে হাইফেন নিজের, পরের তিনটে গ্রুপের, শেষ তিনটে অন্যের। এবার কাজটা এখানেই শেষ নয়, গোটা সিস্টেমে যেখানে যেখানে পারেন, 'ls -al' কমান্ডটা মারুন এবং অনুমতির চক্করটা বোঝার চেষ্টা করুন। শুধু একটা কথা আলাদা করে বলে দিই। এই যে ফাইল পড়া বা লেখা বা চালানো, ডিরেক্টরি ফাইলের বেলায় এদের স্বাভাবিক অর্থগুলো তো অর্থহীন হয়ে যাচ্ছে। ডিরেক্টরি ফাইলের বেলায়, 'r' বা 'w' বা 'x'-কে বুঝতে হবে আলাদা অর্থে। ডিরেক্টরি তো প্রোগ্রাম না, কোনো গাড়িও না যে চালালেই হল। ধরুন একটা ডিরেক্টরি 'a'। তার মধ্যে একটা ফাইল আছে 'b'। এই অবস্থায় একজন ইউজারের এই ডিরেক্টরিকে পড়ার অনুমতি আছে মানে, সে 'a' ডিরেক্টরির উপর 'ls' কমান্ড মেরে তালিকায় 'b' ফাইলটাকে দেখতে পাবে। আর পড়ার অনুমতি বা 'r' না থাকলে, এই ইউজার 'ls' কমান্ড দেওয়া মাত্র ব্যাশ তাকে সবিনয়ে জানিয়ে দেবে, তোমার এই আদেশ দেওয়ার অধিকার নেই খোকা। ধরুন, পড়ার অনুমতি না থাকলেও এই ইউজারের এই ডিরেক্টরিতে 'w' এবং 'x', মানে লেখার এবং একজিকিউট করার অনুমতি আছে। এই অবস্থায় এই ইউজার, 'ls' কমান্ড দিয়ে ফাইলের তালিকা দেখতে না-পেলেও, যদি সে জানে এই 'a' ডিরেক্টরিতে 'b' ফাইলটা আছে, তাহলে 'cd' কমান্ড দিয়ে ডিরেক্টরিটায় ঢুকতে পারবে এবং সেখানে কোনো প্রোগ্রাম দিয়ে 'b' ফাইলটাকে বদলাতেও পারবে, বা নতুন করে একটা 'c' ফাইল লিখতেও পারবে সেই 'a' ডিরেক্টরিতে। কিন্তু, রসিকতাটা এইখানে যে, লেখার পরেও সেই 'c' ফাইলটা 'ls' করে দেখতে পারবেনা। মানেটা বুঝতে পারলেন? ডিরেক্টরি ফাইলের বেলায় 'w' হল ভিতরের ফাইল বদলানোর অনুমতি, আর 'x' মানে ডিরেক্টরির ভিতরে ঢোকার অনুমতি। আবার ধরুন, ওই ইউজারের 'a' ডিরেক্টরিতে 'r' এবং 'x', মানে, পড়ার এবং এক্সিকিউট করার অনুমতি আছে, কিন্তু লেখার অনুমতি নেই। এই অবস্থায় এই ইউজার 'a' ডিরেক্টরিতে 'ls' কমান্ড দিয়ে ফাইলের তালিকা দেখতে পারবে, বা 'b' ফাইলটা পড়তেও পারবে, কিন্তু নতুন করে 'c' ফাইলটা লিখতে পারবে না। আর, তৃতীয় খ্যাঁচটা ঘটতে পারে, যদি এই ইউজারের 'a' ডিরেক্টরিতে 'r' এবং 'w', মানে পড়ার বা লেখার অনুমতি থাকে, কিন্তু এক্সিকিউট করার বা ঢোকার অনুমতি যদি না-থাকে। তখন এই ইউজার 'cd' কমান্ড দিয়ে এই করে 'a' ডিরেক্টরিতে ঢুকতে পারবে না, কিন্তু বাইরে থেকে পুরো ঠিকানা বা পথনির্দেশ-সহ কমান্ড দিয়ে ফাইল পড়তে বা বদলাতে বা নতুন ফাইল লিখতে পারবে। পথনির্দেশটা মনে পড়ছে তো? এটা বারবার মেলাতে মেলাতে এগোন। পুরোনো প্রসঙ্গগুলো মাথায় থাকছে কিনা।

১.২।। ফাইলের অনুমতি বদলানো

এবার কথা হল, যে অবস্থায় আছে, সেখান থেকে, দরকার মত আপনি একটা ফাইলের অনুমতি বদলাবেন কী করে। অনুমতি বদলানোর জন্যে দরকার ওই ফাইলটার উপর আপনার মালিকানা। আর সুপারইউজার বা রুট যে কোনো ফাইলের মালিকানা যে কোনো সময় বদলে দিতে পারে। এই বদলানোর দুটো প্রক্রিয়া আছে, একটা হল অক্ষর দিয়ে, অন্যটা হল সংখ্যা দিয়ে।

অক্ষর দিয়ে বদলানো মানে কোনো একটা ফাইলের মালিকের সাপেক্ষে তিন ধরনের ব্যবহারকারীর জন্যে তিনটে অক্ষর, 'u' মানে সে ইউজার নিজে, 'g' মানে তার গ্রুপের যে কোনো ব্যবহারকারী, আর 'o' মানে আদারস বা অন্য আর সবাই, আর 'a' মানে সবাই, যার মধ্যে 'u', 'g', বা 'o' সকলেই পড়ে। আর অনুমতি তো আমরা আগেই জানি — 'r' মানে রিড বা পড়ার অনুমতি, 'w' মানে রাইট বা লেখার অনুমতি, 'x' মানে একজিকিউট বা চালানোর অনুমতি। আর অনুমতি দেওয়া মানে '+', অনুমতি তুলে নেওয়া মানে '-'। এবার ধরুন, আমি 'essay.all.text' নামে ফাইলের উপর নানা ধরনের অনুমতি বদলে দেখতে চাইছি, তখন আমার কমান্ড হবে 'chmod'। আপনার কৌতূহল হচ্ছে না, কমান্ডটা সম্পর্কে এখনি কিছু জেনে নেওয়ার? হচ্ছে? গুড। ম্যান আছে কী করতে? ধরুন, আমি চাইছি প্রবন্ধ লেখালেখির একটা স্তালিনিস্ট অ্যাসেম্বলি লাইন, প্রবন্ধটা পড়ার এবং বদলানোর সাম্যবাদী গনতান্ত্রিক অধিকার, নিজের, নিজের গ্রুপের, এবং অন্য সবার। কমান্ড দেব, 'chmod a+rw essay.all.text'। ডিরেক্টরির বেলাতেও এই একই ভাবে কাজ করতে হবে। শুধু ডিরেক্টরির বেলায় একটা বাড়তি অপশান '-R'। এর মানে রিকার্সিভ বা পারম্পরিক। ধরুন, 'chmod -R g+x my.directory', মানে, শুধু 'my.directory' ডিরেক্টরিটা আমার গ্রুপের সকলের কাছে এক্সিকিউটেবল হয়ে গেল তাই নয়, এর মধ্যের যাবতীয় ফাইলরা, ছানা সাবডিরেক্টরিরা, তাদের ছানারা, বংশপরম্পরায় সকলেই প্রবেশযোগ্য হয়ে গেল আমার গ্রুপের কাছে।

এই অক্ষর দিয়ে অনুমতি বদলানোটা আমার বেশ জটিল লাগে, এত কিছু মনে রাখতে হয়। এর চেয়ে সহজ লাগে সংখ্যা দিয়ে বদলানোটা। তিনটে করে হাইফেনের তিনটে গুচ্ছের বদলে তিনটে সংখ্যা। প্রথম সংখ্যাটা ইউজারের নিজের। দ্বিতীয় সংখ্যাটা গ্রুপের। তৃতীয় সংখ্যাটা অন্য সকলের। সংখ্যাটা কী ভাবে পাব? সংখ্যাটার মান হতে পারে আটটা, শূন্য থেকে সাত। '0', '1', '2', '3', '4', '5', '6', এবং '7'। '0' মানে কোনো কিছুর অনুমতি নেই, '1' মানে এক্সিকিউশনের অনুমতি, '2' মানে লেখার, '4' মানে পড়ার অনুমতি। এবং অন্যগুলো পাচ্ছি এদের যোগ করে। চালানো এবং লেখার অনুমতি মানে '1' আর '2', মানে '3'। চালানো আর পড়ার অনুমতি মানে '1' আর '4', মানে '5'। লেখা আর পড়ার অনুমতি মানে '2' আর '4', মানে '6'। আর, চালানো লেখা এবং পড়া, সবকটার অনুমতি মানে '1' আর '2' আর '4', মানে '7'। এবার 'chmod' কমান্ডটার সঙ্গে আসবে একটা তিন অঙ্কের সংখ্যা, প্রথমটা কমান্ড যে দিচ্ছে সেই ইউজারের নিজের, পরেরটা তার গ্রুপের, শেষেরটা আদারস বা অন্যের। এবার ধরুন আমি সবাইকে সব অনুমতি দিতে চাইছি 'somefile' ফাইলটার উপর, তার মানে কমান্ডটা হবে, তিনটে ঘরের জন্যেই '7'। মানে, 'chmod 777 somefile'। আমি নিজের জন্যে পড়ার লেখার আর চালানোর অনুমতি চাইছি, গ্রুপের জন্যে চালানোর আর পড়ার, অন্যের জন্যে চালানোর, তার মানে কমান্ডটা দাঁড়াবে, 'chmod 751 somefile'। এবার করে করে দেখুন, কী হয়। যত করে দেখবেন, মাথায় বসিয়ে ফেলতে পারবেন ছকটা, তত আপনি গ্নু-লিনাক্স শেখার কাছাকাছি যাচ্ছেন।

খুব ঝঞ্ঝাটের লাগছে, না? লাগবে না, বরং কৃতজ্ঞ লাগবে, যখন মনে হবে, এটাই সেই নিরাপত্তা, যা দিয়ে আপনার গ্নু-লিনাক্স সিস্টেমের ভাইরাস-অনাক্রমণীয়তাটা গড়ে উঠেছে, ইমিউনিটিটা। ভাইরাস তো এক ধরনের প্রোগ্রাম, যারা সিস্টেমে ঢুকে, সিস্টেম ব্যবহারকারীদের ইচ্ছে এবং প্রয়োজনের বাইরে গিয়ে সিস্টেমের ফাইল বদলাতে থাকে। এবার ভাবুন তো, কেন বলছি, যে এই মালিকানা আর অনুমতির ছকটাই নিরাপত্তা ব্যবস্থা। ধরুন এটা এই জিএলটি ইশকুল পাঠমালার টাস্ক রইল, কোনো খুঁটিনাটি না, কিছু না, জাস্ট এই যতটুকু জানেন এর ভিত্তিতেই ভেবে ওঠার চেষ্টা করুন তো কেন এটাকেই নিরাপত্তা বলছি? ও, একটা কথা, এই যে কোনো মালিকানা এবং অনুমতির ছকে রুট বা সুপারইউজার কিন্তু পড়েনা, তাই রুট নিজে যদি সিস্টেমের বেগড়বাই ঘটাতে চায়, সে নিজেই যদি একটা ভাইরাসকে চালু করে, রুট ইউজার বা সুপারইউজার হয়ে, তাহলে কিন্তু সে ভাইরাস তখন ভগবান, হল বলরাম

স্কন্ধে। তবে, রুট হয়ে অত ভাইরাস চালানোর পরিশ্রমের দরকার কী, সিধে তো ফাইলসিস্টেমটাই উড়িয়ে দিতে পারে সে, একদম গোড়া থেকে, আগেই বলেছি।

১.৩।। সিস্টেম এবং ইউজার

প্রত্যেক ইউজারের আলাদা আলাদা অনুমতি এবং মালিকানার এই গোটা কাঠামোটা সিস্টেম নিজের মাথায় রাখে '/etc' ডিরেক্টরির কয়েকটা কনফিগারেশন ফাইল দিয়ে। তার কিছুটা কিন্তু আপনি ইতিমধ্যেই জানেন, পাঁচ নম্বর দিনের আলোচনায়, '/etc/passwd' আর '/etc/shadow' ফাইলদুটোর কথা মনে করুন। মনে করুন, মেশিনের অপারেটিং সিস্টেম জেগে ওঠার সময় গোটা প্রক্রিয়াটার শেষ ধাপে এসে 'login' কিভাবে এদের পড়ে। গোটা সিস্টেমের মধ্যে একমাত্র এই '/etc/passwd' ফাইলটাতেই থাকে প্রতিটি নথীবদ্ধ ইউজারের খুঁটিনাটি। কিছু কিছু ডিস্ট্রিবিউশনে, বিশেষ করে সাম্বা বলে একটা সফটওয়ারের কার্যধারায়, এর কিছু ব্যত্যয় ঘটে, তবে তা নিয়ে এখন মাথা না-ঘামালেও চলবে। সাতটা ফিল্ডে বাঁধা এর লাইন নিয়ে আমাদের আলোচনা মনে করুন। এর মধ্যে একটা লাইন আমরা তুলে নিয়েছিলাম।

```
piu:x:503:100:Smita Bhadra:/home/piu:/bin/bash
```

':' চিহ্ন দিয়ে আলাদা করা 'piu', 'x', '503', '100', 'Smita Bhadra', '/home/piu', '/bin/bash', এই সাতটা ফিল্ড। এর প্রথমটা, 'piu' তো অবশ্যই ইউজারের নাম, সেই নাম যা দিয়ে তাকে সিস্টেমের ইউজার করা হয়েছিল, 'useradd' কমান্ডে। দ্বিতীয় ফিল্ডটা, 'x' একটা প্রতীক, ওই ইউজারের পাসওয়ার্ডের, আগেই বলেছি, নিরাপত্তার কারণে পাসওয়ার্ডটা এখানে রাখা হয়না। পঞ্চম মানে 'Smita Bhadra' ওই ইউজারের পুরো নাম। ষষ্ঠ ফিল্ড '/home/piu' তার হোম-এর ঠিকানা। সপ্তম ফিল্ড '/bin/bash' তার শেল। এগুলোর কথা আমরা পাঁচ নম্বর দিনেই বলেছি। তৃতীয় আর চতুর্থ ফিল্ডটা তখন আমরা আলোচনা করিনি। এখন এর পরের সেকশনগুলোয় এদের আলোচনাটা আসবে। তৃতীয় আর চতুর্থ ফিল্ডের ওই সংখ্যা দুটো, '503' এবং '100', 'piu' নামের ওই ইউজারের ইউজার-আইডি বা ইউআইডি (uid) আর গ্রুপ-আইডি (gid), ব্যক্তি-পরিচিতি আর গ্রুপ পরিচিতি। এখন বুঝতে পারছেন? এরা হল দুটো সংখ্যা, যাদের দিয়ে সিস্টেম চেনে 'piu' নামক ইউজার আর তার গ্রুপকে, যাদের আমরা চিনি ইউজার 'piu' আর তার গ্রুপ 'Users'। এই সুজে ৮.২ ডিস্ট্রিবিউশনের প্রথায়, এখানে দেখুন নম্বরদুটো আলাদা। সুজেতে 'ইউজারস' বলে গ্রুপটা তৈরি করে সেখানে সাধারণ ব্যবহারকারীদের রাখা হয়। যদিও চাইলে সুজেতেও অন্যভাবে, মানে প্রত্যেকের জন্যেই একটা ব্যক্তিগত গ্রুপ তৈরি করে, ইউজারদের রাখাই যায়। গ্নু-লিনাক্সের অনেক ডিস্ট্রিবিউশনের সঙ্গেই মেলেনা সুজের আলাদা আলাদা ইউআইডি এবং জিআইডি-র এই প্রথাটা। যেমন দেখুন, এবার স্ল্যাকওয়ার থেকে, ওই একই '/etc/passwd' ফাইলের একই লাইন, এখানে কিন্তু নম্বর দুটো এক, দুটোই 'piu' বা '1004'। এইখানে দেখুন ইউআইডি আর জিআইডি দুটোই এক। এবং রেডহ্যাট ডিস্ট্রিবিউশনেও, ডিফল্টে এভাবেই হয়, আলাদা করে গ্রুপ বিষয়ে কোনো অপশান না দিয়ে দিলে।

```
piu:x:1004:1004:Smita Bhadra:/home/piu:/bin/bash
```

আগেই বলেছি, সিস্টেমে নতুন ইউজার বাড়ানোর অধিকার সাধারণ ইউজারের থাকেনা, একমাত্র রুটই পারে কোনো নতুন ইউজারকে সিস্টেমে নথীবদ্ধ বা রেজিস্টার করাতে, 'useradd' বলে একটা কমান্ড দিয়ে। এই 'useradd' কমান্ডটার ম্যানুয়াল পেজ বা ইনফো পেজ পড়ে দেখুন, '/etc/passwd' ফাইলের লাইনে লাইনে এক একজন ইউজারের ওই সাতটা ফিল্ডের প্রত্যেকটাতেই কী তথ্য যাবে সেটা সিস্টেমকে দিয়ে দেওয়া যায় 'useradd' কমান্ড দিয়ে নতুন ইউজার রেজিস্টার করানোর সময়েই। নতুন ইউজারের ইউজারনেম, পাসওয়ার্ড, গ্রুপ, পুরো নাম, হোম এবং শেল কী হবে। আলাদা আলাদা অপশান আছে 'useradd' কমান্ডের, যা দিয়ে এই প্রতিটি তথ্যই দেওয়া যায়। যেই একজন ইউজার রেজিস্টার্ড হয়ে গেল, তার নাম, ধাম, গোত্র, বাড়িতে পুকুর আছে কিনা, পুকুরে ঘটি ডোবে কিনা, ইত্যাদি লিখে নেওয়া হল চিত্রগুপ্তর খাতায়, মানে '/etc/passwd' ফাইলে। ভোটার লিস্টে তার নাম উঠে গেল। এর আগে অব্দি সিস্টেমের কাছে তার কোনো অস্তিত্ব নেই। একবার রেজিস্টার করার বা নথীবদ্ধ হয়ে যাওয়ার পরেও যদি একজন ব্যবহারকারীর খুঁটিনাটি বদলাতে চান, তার জন্যে কমান্ড আছে 'usermod'। এই কমান্ডটাও সাধারণ ইউজার হয়ে কমান্ড প্রম্পটে দেখাবে, কমান্ড নট ফাউন্ড — আদেশ খুঁজে পাওয়া গেল না গো।

কিন্তু রুট বা 'root' হয়ে নিলেই সব সিধে। কেন এরকম হয়, মনে করতে পারছেন? একবার খুঁজে পায়না, আর একবার পায়? এরকম হয় কেন? আর 'usermod' কমান্ডটাকে জানতে চান? আর একবার বলে দেব ম্যান পেজ পড়ার কথা? এখন থেকে, যখনই কোনো নতুন কমান্ড উল্লেখ হবে, ধরেই নেবেন তার সঙ্গে এই ধ্রুবপদটা লাগানোই আছে, 'ম্যান পড়ার নেশা করো ম্যান, খেজুরগাছে হাঁড়ি বাঁধো ম্যান' (কার্টসি দোহার)। আমাদের কলকাতা লাগে অরিজিত আর তথাগত দুজনেই ভালো ক্লাসিকাল গান গায়, ওদের একবার বলে দেখব ভাবছি, লাগের মিটিং-এ এটা গাইতে।

ইউনিক্স বা গ্নু-লিনাক্সের অনেক টেক্সটেই দেখেছি, রাশি রাশি কমান্ডের ব্যবহারবিধি পরপর দিয়ে যেতে। আমিও এখানে দিতেই পারতাম। কিন্তু নিজে শেখার সময় দেখছি, নিজেই তো শিখছি এখনো, বারবার মনে হয়, এত বোগাস আর বোরিং এইগুলো লিখে দেওয়া, রাশিরাশি গুষ্টির ওই রাশিরাশিতর পিন্ডি। ও ছাই মনেও রাখা যায়না। আবার না-পড়ে পাতা উল্টে যেতেও একটু অস্বস্তি হয়। কমান্ড মনে রাখার একটাই উপায়। কমান্ডটাকে ব্যবহার করা, ব্যবহার করে চলা, ম্যানপেজ পড়ে পড়ে, করে করে, দেখে দেখে। গ্নু-লিনাক্স সিস্টেম আপনাকে যা খুশি করার মজাটা যেমন দেয়, তেমনি তার জন্যে নিজেকে বানিয়ে নেওয়া লাগে। কেন এত বারবার বিরক্তিকর রকমে এই ম্যান পড়ার কথা বলছি এটা বুঝতে পারবেন যেদিন প্রথম নিজের গ্নু-লিনাক্স সিস্টেমে নিজের মজায় নিজের খুশিমত কাজ করতে চাইবেন। ইট টেকস এ ওরিড ম্যান টু সিং এ ওরিড সং। প্রচুর অবিশ্বাস্য রকমের প্রচুর কাজ করার সুযোগ এবং উপায় দেওয়া থাকে গ্নু-লিনাক্সে। কিন্তু তাকে ব্যবহার করার তরিকাগুলো প্রথমে শিখে নিতে হয়, এবং শেখার উপায় একমাত্র ওই রগড়ে রগড়ে, লার্ন ইট দি হার্ড ওয়ে।

এরপর আসে চিত্রদার দ্বিতীয় ফাইল, সেটাকেও চিনি আমরা, তার ছায়া-ছবি দেখেছি পাঁচ নম্বর দিনেই, সেই ছায়া-ফাইলের, '/etc/shadow' তার নাম। আগেই বলেছি, আগে প্রবেশ-সংকেত বা পাসওয়ার্ডটা '/etc/passwd' ফাইলেই থাকত, পরে এই '/etc/shadow' ফাইলটা সিস্টেমে চালু হল একটা বিশেষ প্রয়োজন থেকে। '/etc/passwd' ফাইলটা নিয়ে সমস্যা এই যে সেটা যে কারুকেই যে কোনো সময়ে পড়তে হবে, জনপাঠযোগ্য বা ওয়ার্ল্ড-রিডেবল হতে হবে। এই ফাইল থাকে সরাসরি রুটের বা সিস্টেমের এলাকায়, সাধারণ ইউজারদের কেউ '/etc' ডিরেক্টরির ফাইলের মালিক হয়না। '/etc/passwd' ফাইলের ফাইল অনুমতির তিনটে জায়গাতেই, রুট ইউজার, তার গ্রুপ, এবং অন্যদের তিনটেতেই '`r`' দিয়ে রাখতে হবে। তার কারণ, সে 'root' হোক, বা কোনো সাধারণ ইউজার, যখনি যে কেউ কোনো প্রোগ্রাম চালাচ্ছে, সেই প্রসেস বা চলমান প্রোগ্রাম বলুন, সিস্টেম বলুন, প্রত্যেকেই তার প্রয়োজনীয় তথ্যগুলো পড়ে নেয় ওই '/etc/passwd' ফাইল থেকে। পাঁচ নম্বর দিনে, শেলের গল্প মনে আছে? যে প্রসঙ্গটা ঠিক আগের আগের প্যারাগ্রাফেই আপনাকে মনে পড়ানোর চেষ্টা করে এলাম। সাধারণ ইউজার 'usermod' কমান্ড দিলে ব্যাশাধম দেখায়, কমান্ড নট ফাউন্ড, আর 'root' চালালে সঙ্গে সঙ্গে পালন করে। খুঁজে পাওয়ার এই পার্থক্যটা ঘটে কেন?

কারণ, কমান্ড প্রম্পটটা ফুটিয়ে তোলার আগে থেকেই গোটা পদ্ধতিটা মনে করুন। 'login' আপনাকে ঢুকতে দেওয়ার আগে '/etc/passwd' ফাইল থেকে মিলিয়ে নিয়েছে, এই সিস্টেমে আপনার প্রবেশাধিকার আছে কিনা, আপনার পাসওয়ার্ড মিলিয়ে নিয়েছে '/etc/shadow' ফাইল থেকে। তখনি সে দেখে নিয়েছে আপনার সম্পর্কে যাবতীয় তথ্য, আপনার গ্রুপ কী, আপনার হোম কী, আপনার শেল কী। মানে প্রাথমিকভাবে দেখে নিয়েছে আপনার কোন কোন অধিকার আছে এবং নেই। এর পরেই সে পরে নেয় আরো কিছু কনফিগারেশন ফাইল, যা আপনার একান্ত নিজস্ব, থাকে আপনারই হোম ডিরেক্টরিতে। '~/.bash_profile', '~/.bashrc', '~/.bash_logout', '~/.bash_history', '~/.bashrc', '~/.profile' ইত্যাদি। যাদের এমনিতে আপনি দেখতে পাননা, তার কারণটা তো বুঝতেই পারছেন, প্রত্যেকের নামের গোড়ায় ওই বিন্দু বা '.'-টা দেখুন। আর প্রত্যেকের ঠিকানার গোড়াতেই ওই '~/' অংশটা বোঝাচ্ছে যে, এরা প্রত্যেকেই আছে আপনার হোম ডিরেক্টরিতে। মানে এটা ইউজার 'piu' হলে ওই '~/' অংশটা বোঝাত '/home/piu/'। মানে প্রত্যেকটা ফাইলই তখন খুঁজে নেওয়া হত '/home/piu' ডিরেক্টরি থেকে। নিজের হোম ডিরেক্টরিতে '`ls -a`' কমান্ড দিয়ে দেখুন তো এই ফাইলগুলো আছে কিনা, বা কোন কোন ফাইল আছে, তাদের খুঁটিনাটিগুলো বোঝার চেষ্টা করুন '`ls -al`' কমান্ড দিয়ে। এবং যেগুলো আছে, তাদের '`less`' কমান্ড দিয়ে পড়ার চেষ্টা করে দেখুন। এই মুহূর্তে প্রায় কিছুই বুঝতে পারবেন না, কিন্তু ব্যবস্থার সঙ্গে চেনাশোনাটা বাড়ায় লাভ বই ক্ষতি কিছু নেই। শুধু কিন্তু ইউজারের হোম ডিরেক্টরি নয়, কিছু কনফিগারেশন ফাইল থাকে

সিস্টেমের '/etc' ডিরেক্টরিতেও। একবার 'ls -al /etc/passwd' মেরে দেখে নিন, আপনার সম্পর্কে যাবতীয় ডিটেইলস ঠিকঠাক রেখেছে কিনা সিস্টেম। মেলানোটা তো এখান থেকেই শুরু। দেখেছেন? গুড। এই মুহূর্তে খুব বেশি কিছু বোঝার চেষ্টা না-করাই ভালো। এই কনফিগারেশন ফাইলদের নিয়ে আলোচনা আসবে আমাদের পাঠমালার দশ নম্বর দিনে, ধরে ধরে। আপাতত এটুকু মাথায় রাখুন, কোনো একজন বিশেষ ইউজারের প্রতিটি অধিকার অনুমতি ইত্যাদির খুঁটিনাটি সিস্টেম জেনে নেয় এই ফাইলগুলো থেকেই। তাহলে গোটা ব্যবস্থাটা ভাবুন, যখনি কোনো প্রোগ্রাম চালু করছে কোনো ইউজার, সিস্টেম প্রোগ্রাম বা ইউজার প্রোগ্রাম যাই হোক, প্রথমে দেখা হচ্ছে, '/etc/passwd'। তারপরে, '/etc/passwd' ফাইলে দেওয়া হদিশ অনুযায়ী হোম ডিরেক্টরির এবং '/etc' ডিরেক্টরির কনফিগারেশন ফাইলগুলো। অর্থাৎ, '/etc/passwd' ফাইলটাকে প্রতি মুহূর্তে পড়তে হবে প্রোগ্রামগুলোর এবং সিস্টেমের। জাস্ট কোনো ইউজারের কাজ করাটা চালিয়ে নিয়ে যেতে হলেই ফাইলটা মুর্হুমুহূ পড়তে হবে সবাইকে। এবার বুঝতে পারছেন, কেন সাধারণ ইউজার 'usermod' কমান্ড দিলে, ব্যাশ জানায় খুঁজে পাইনি, আর 'root' চালালে পালন করে? কারণ, '/etc/passwd' ফাইল ঘুরে আপনার হোম ডিরেক্টরির কনফিগারেশন ফাইল থেকে ব্যাশ আপনার অধিকারের এলাকার পথনির্দেশটা পেয়ে যায়। এবার যদি আপনি 'root' হন, তাহলে আপনার পথনির্দেশের মধ্যে সে '/sbin' বা '/usr/sbin' ইত্যাদি সিস্টেম বাইনারির ডিরেক্টরিগুলো পেয়ে যায়, যেখানে থাকে 'usermod' কমান্ডটা। আর সাধারণ ইউজার হলে তার অধিকারের পথের মধ্যে এগুলো পড়েনা। তাই 'usermod' কমান্ডটাও সে খুঁজে পায়না। তার আর দোষ কী? পাঁচ নম্বর দিনের একদম শেষের শেলের কমান্ড খোঁজা আর পথনির্দেশের জায়গাটা আর একবার একটু দেখে নিন তো, আগের চেয়ে এখন একটু বেশি বুঝতে পারছেন কিনা? এই কমান্ড খুঁজে না-পাওয়ার খ্যাঁচটা সেই মুহূর্তের জন্যে সমাধান করার দুটো তরকিব এখানে দিয়ে রাখি।

তরকিব নাম্বার ওয়ান হল শেলের জায়গায় নিজেই খুঁজে নেওয়া, 'whereis' কমান্ড দিয়ে, তারপর গোটা ঠিকানাটা শেলকে দিয়ে দেওয়া। ধরুন, আপনি 'root' হয়ে কোনো একটা সিস্টেমের কাজ করছেন। এইসময় আপনার মনে হল, ব্যাকগ্রাউন্ডে একটু গান বাজলে মন্দ হয়না। আপনি কমান্ড দিলেন 'mplayer song.mp3'। কিন্তু শেল জানালো, কমান্ড খুঁজে পায়নি। যাত্তারা, এসব কী? আসলে, 'mplayer' প্রোগ্রামের বাইনারি ফাইলটা আছে '/usr/local/bin' ডিরেক্টরিতে, 'whereis mplayer' কমান্ড দিয়ে আপনি সেটা জেনে নিলেন। এই ঠিকানাটা 'root'-এর পথে থাকেনা। ওসব পাঁচপেঁচি প্রোগ্রাম তো সাধারণ ইউজাররা চালায়, রুটের কাজ শুধু শাসন করা। এবার রুট হয়েই আপনি যদি গোটা ঠিকানাসহ কমান্ড দেন, '/usr/local/bin/mplayer song.mp3', দিব্য দেখবেন গান চলছে। কিন্তু বারংবার এতটা সহ কমান্ড দেওয়াটা একটু আসুরিক লাগতে পারে। তখন তরকিব নাম্বার টু কাজে লাগাতে পারেন। একটা কমান্ড দিন, 'PATH=$PATH:/usr/local/bin'। মানে এই মুহূর্তে পথ যেটা আছে তার সঙ্গে '/usr/local/bin' ঠিকানাটুকুও আপনি যোগ করে দিলেন। এখন থেকে শেল এই ঠিকানাটাতেও খুঁজে নেবে। এখন থেকে শুধু 'mplayer song.mp3' কমান্ড দিয়েই রুট অবস্থাতেই আপনি গান চালাতে পারবেন। পরের বার লগ-ইন করে আর এটা পাবেন না, তখন আবার যোগ করে নিতে হবে। তরকিব এক আপনি এখনি বুঝতে পারছেন। তরকিব দুইটা আমি ইচ্ছে করেই এখানে দিলাম না। অনেক পরে আসবে এটা, দেখুন তো তার আগেই আপনি নিজেই বুঝে নিতে পারেন কিনা।

যা বলছিলাম, ফেরত আসা যাক, '/etc/passwd' থেকে '/etc/shadow' ফাইল চালু হওয়ার প্রসঙ্গে। শেল তথা অন্য প্রোগ্রামগুলো এই '/etc/passwd' ফাইল থেকেই সংগ্রহ করে নেয়, যে তাদের প্রোগ্রামটা চালাচ্ছে তার সংক্রান্ত খুঁটিনাটি, যখন যখন তাদের ইউজারের পুরো নাম, তার হোম ইত্যাদি তথ্য কাজে লাগছে। তাই তাকে ওয়ার্ল্ড-রিডেবল বা সর্বসাধারণের পাঠযোগ্য হতেই হবে। এবার ভাবুন, পুরোনো প্রথায়, যখন '/etc/shadow' ফাইল থাকত না, যে কেউ, যে কোনো প্রোগ্রাম '/etc/passwd' ফাইলটায় এনক্রিপ্টেড চেহারায় পাসওয়ার্ডটাকে দেখতে পাচ্ছে, প্রতিটি লাইনের দ্বিতীয় ফিল্ড থেকে। তার মানে অন্য যে কেউ যে কোনো ইউজারের এনক্রিপ্টেড পাসওয়ার্ড বা ধাঁধানো প্রবেশ-সংকেতটা তুলে নিতে পারছে, এবার সম্ভাব্য প্রতিটা সমাহারকে নিয়ে ট্রাই করে যাচ্ছে। আর এক একটা বড় সিস্টেমে শ-শ ইউজার, তাদের মধ্যে এমন কেবলরাম পাবলিক দু-এক পিস থাকবেই যাদের পাসওয়ার্ড আন্দাজ করা খুব সহজ। নিজের জন্মদিন, প্রেমিকার ডাকনাম, পছন্দের হিরো, এই গোছের। সেখানে লোকটাকে জানার চেনার আন্দাজ থেকেই পাসওয়ার্ড ভাঙার একটা সুবিধে মিলে যায়। আবার, কিছু পাসওয়ার্ড থাকে যারা

সরাসরি ডিকশনারি শব্দের সঙ্গে মেলে, এই ধরনের পাসওয়ার্ড ভাঙার জন্যে 'ডিকশনারি-অ্যাটাক' বা অভিধান-আক্রমণ করা হয় বহুসময়। সরাসরি একটা ডিকশনারি ধরে ইংরিজির চালু আশিহাজার শব্দকে পরপর মিলিয়ে যাচ্ছে, এটার চালু নামই হল 'ডিকশনারি অ্যাটাক'। এ ধরনের প্রোগ্রাম বানানোটাও এমন কিছু শক্ত নয়, যে নিজেই পরপর দেখে যাবে, একের পর এক। কেউ কেউ ডিকশনারি শব্দের আগে পরে একটা করে নম্বর লাগায়, সেটাও খুব সহজেই নিজের অ্যালগরিদমে নিয়ে আসতে পারে পাসওয়ার্ড ক্র্যাক করার সফটওয়ারগুলো। যেটা একজনের মাথায় আসে সেটা আর এক জনের মাথায় আসবেই।

যেমন স্প্যাম মেল-এর কথাই ভাবুন। ইলেকট্রনিক সংযোগের দুনিয়ায় একটা বৃহত্তম সমস্যা এখন এই স্প্যাম মেল। স্প্যাম মেল মানে অকাঙ্খিত অপ্রয়োজনীয় অনিমন্ত্রিত মেল, কখনো কখনো আফ্রিকান কোনো জেনেরালের বাচ্চার কাছ থেকে দশ মিলিয়ন ডলার দয়া করে গ্রহণ করার নেমন্তন্ন ইত্যাদি ভাট মেল, আর মূলত বিভিন্ন পর্নো সাইটের বিজ্ঞাপন। এই জাংক মেল বা স্প্যাম মেল আটকানোর জন্যে যেসব সফটওয়ার আছে, তারা মেলের সাবজেক্ট লাইনে কিছুর শব্দের ভিত্তিতে স্প্যাম আটকায়। কিছু যৌনশব্দের একটা তালিকা, তার থেকে কোনো শব্দ সাবজেক্ট লাইনে থাকলেই মেলটা আটকে দেবে জাংক মেল ফিল্টারে, জঞ্জালচিঠির চিরুনিতে। কিন্তু যা বললাম, টেকনোলজি বেরোলেই তার প্রতিটেকনোলজি বেরোয়। স্প্যামাররা এদের মেলগুলো বানায় কোনো প্রোগ্রাম দিয়ে। রাশি রাশি কাল্পনিক বা সংগৃহীত মেল আইডিতে রাশি রাশি মেল ছেড়ে যাওয়ার প্রোগ্রাম। মেলে আপনি সাড়া দিলেন কী মরলেন, এমনকি পাঠানোতে আপত্তি জানাতেও যদি যান তো, স্প্যামারের কাছে এই তথ্যটা চলে গেল যে, এই আইডিটা বাস্তব। যাই হোক, যেই স্প্যাম ফিল্টাররা ওই যৌনশব্দের তালিকা ব্যবহার করা শুরু করল, সঙ্গে সঙ্গে স্প্যামাররাও যৌনশব্দগুলোর একটা করে অক্ষর বদলে দিতে লাগলো, উচ্চারণের দিক দিয়ে কাছাকাছি বা আকৃতির দিক দিয়ে কাছাকাছি, 'o' বদলে '0', 'g' বদলে 'z', ইত্যাদি। স্প্যামারদের মত পাসওয়ার্ড ভাঙিয়ে তথা সিস্টেম ক্র্যাকাররা প্রায়শই বেশ বুদ্ধিমান অল্পবয়স্ক ছেলেপুলে হয়, দুনিয়াকো কুছ কর দিখানার মনোবৈজ্ঞানিক সমস্যায় ভোগে, নিজের যৌন-আত্মবিশ্বাসের অভাবকে বিকিরণ করে দুনিয়াকে তার দিকে তাকাতে বাধ্য করার মধ্যে দিয়ে। কখনো কখনো এর মধ্যে বাবা-অবস্থানের প্রতি একটা বিরূপতাও থাকে। ব্যক্তি পারিবারিক বা সামাজিক স্তরে। ভাইরাসও বানায় এরাই, পনেরো থেকে তিরিশ বছর বয়সের, মূলত, একটু সমাজবিমুখ একটা মানসিক রোগে ভোগা পুরুষ ছেলেপুলেরা। এখানে একটা জনপ্রিয় কিন্তু বিরক্তিকর ভুল ব্যবহারকে উল্লেখ করে রাখা ভালো। পৃথিবীর সব দেশেই সাংবাদিকতার মূল শর্ত প্রাথমিকভাবে একটা বাচাল অর্ধসাক্ষর আত্মবিশ্বাস, যাতে পৃথিবীর যে কোনো বিষয়েই যে কোনো কথা লিখে যাওয়া যায় জানার কোনো তোয়াক্কা না-রেখে। এই সাংবাদিকদের কল্যাণে মিডিয়ায় 'হ্যাক' (hack) তথা 'হ্যাকার' (hacker) শব্দদুটো দাঁড়িয়ে গেছে খারাপ। যারা স্প্যাম মেল লেখে, যারা ভাইরাস বানায়, যারা পাসওয়ার্ড ভেঙে অন্যের সিস্টেমে ঢুকে অসুবিধে সৃষ্টি করে — এদেরকে প্রায়ই দেখি মিডিয়ায় 'হ্যাকার' বলে ডাকা হয়। এটা শুধু ভুল না, ঠিক বিপরীত অর্থে ব্যবহার করা। 'হ্যাকার' শব্দটা গ্নু-লিনাক্স তথা ফ্রি সফটওয়ার জগতে একটা সম্মাণের শব্দ। পাঁচ নম্বর দিনে 'হবিস্ট' শব্দটা মনে আছে, বিল গেটস-এর চিঠি থেকে, এই শব্দটায় যে অর্থটা এসেছিল, তার খুব কাছাকাছি এই 'হ্যাকার' শব্দটা। যারা সিস্টেমকে বদলায়, ভাঙে, নতুন আকার দেয়। নতুন প্রোগ্রাম লেখে, নতুন অপারেটিং সিস্টেম তথা কম্পিউটার মননের জন্ম দেয় — এরাই হল হ্যাকার। মূল জায়গাটা এখানে নির্মাণ। আর অন্যের ইউজার অ্যাকাউন্টে অন্যের সিস্টেমে এই বুদ্ধিমান বোকা ছেলেমানুষি নোকঝোঁক — এর নাম ক্র্যাক (crack) করা ক্র্যাকিং (cracking)। যারা করে তারা ক্র্যাকার (cracker)।

যাইহোক, এই ক্র্যাকারদের হাত থেকে পাসওয়ার্ডগুলোকে বাঁচাতেই মূলত, সিস্টেমে '/etc/shadow' ফাইলটা রাখা চালু হল। '/etc/passwd' ফাইলটা ছিল জনপাঠযোগ্য বা ওয়ার্ল্ড-রিডেবল। দুটো ফাইলের উপরেই '`ls -l`' কমান্ড চালিয়ে দেখে নিন অনুমতির ফারাকগুলো। মিলিয়ে দেখুন, এই ফাইলটা একমাত্র পড়তে পারে 'root' বা রুটের অধিকারসম্পন্ন সিস্টেম। কোনো ইউজারকে 'login' করানোর সময়ে সিস্টেম ইউজারের টাইপ করে দেওয়া পাসওয়ার্ডটাকে এনক্রিপ্ট করে দেখে নেয়, '/etc/shadow' ফাইলে রাখা প্রতিরূপটার সঙ্গে মিলছে কিনা। মনে আছে, আমরা বলেছিলাম, এই '/etc/shadow' ফাইলটা কপি করে টেক্সট ফাইল বানাতে আমার খুব সমস্যা হয়েছিল, কারণ আমি সাধারণ ইউজার হয়ে সেটা করার চেষ্টা করছিলাম। '/etc/shadow' ফাইলের গঠন প্রকরণ, ফিল্ড সাজানো, যতিচিহ্ন সবই হুবহু '/etc/passwd' ফাইলের মত, পাঁচ নম্বর দিন থেকে মিলিয়ে নিন। এখানে আটটা

কোলোন ':' দিয়ে আলাদা করা নটা ফিল্ড বা অবস্থান আছে, তার খুঁটিনাটি অর্থগুলো 'man -5 shadow' করে দেখে নিন কনফিগারেশন ফাইলগুলোর বিবরণ থাকে ম্যানুয়ালের পাঁচ নম্বর সেকশনে। ম্যান আছে কেন?

১.৪।। '/etc' ডিরেক্টরির আরো দুচারটে ফাইল

'/etc/passwd' এবং '/etc/shadow' এই দুটো ফাইলের সঙ্গে এই '/etc' ডিরেক্টরিতেই থাকে আরো দুচারটে ফাইল, যারা গোটা অপারেটিং সিস্টেমের অনুমতি এবং কাঠামোর মালিকানাটা ধরে রাখার কাজে '/etc/passwd' আর '/etc/shadow' ফাইলদুটোকে সহায়তা করে। যেমন, একটা হল '/etc/group'। নাম দেখেই বুঝতে পারছেন, এর কাজ হল গোটা ব্যবস্থার গ্রুপের কাঠামোটাকে নিয়ন্ত্রণ করা। বড় সিস্টেমে ভারি কাজের ফাইল। স্ট্যান্ড-অ্যালোন ছোট ব্যক্তিগত সিস্টেমেও, যদি বেশ কয়েকজন ইউজার থাকে, মাঝে মাঝে ভারি কাজে লাগে, কখনো নিজের রেলাবাজি দেখাতে, কখনো কখনো সত্যিই নিরাপত্তার জন্যে। যে যে ইউজারকে যে যে অধিকার দিতে চান, তাদের সেই সেই গ্রুপে রেখে দিলেন, অন্যদের রাখলেন না। ধরুন, কেউ একটা বারবার বাইরের ফ্লপি ঢুকিয়ে সিস্টেমে ভাইরাস ঢোকাচ্ছে। তাকে বাদ দেওয়ার জন্যে 'floppy' নামের একটা গ্রুপ বানিয়ে নিন, যার সভ্যদেরই কেবল সুযোগ থাকবে ফ্লপি ব্যবহারের। মানে, ফ্লপি ডিভাইসের ডিভাইস ফাইলের উপর শুধু 'root' ইউজারের এবং এই 'floppy' গ্রুপের অধিকার এবং অনুমতি থাকবে। এবং ওই বিপজ্জনক ইউজারকে এই গ্রুপে রাখলেন না। ব্যাস, মিটে গেল।

'/etc/group' ফাইলে থাকে গ্রুপের খুঁটিনাটি বিবরণ, তাই এই ফাইলকে কাঁটছাঁট করে এই কাজগুলো সহজেই করা যায়। আবার সরাসরি কমান্ড দিয়েও করা যায়। যেমন ধরুন, অনেক সময়, সিডি বার্নারের উপর নিয়ন্ত্রণ করার দরকার পড়ে, যে কেউ যাতে সিডি পুড়িয়ে নিতে না-পারে। ধরুন আপনার মেশিনে একটা সিডি-বার্নার আছে, এবং আপনি চাননা যে যে কেউ সেটায় সিডি পোড়াক, তখন আপনি ওই 'floppy' গ্রুপের মত 'burner' নামে একটা গ্রুপ তৈরি করে নিন। নতুন গ্রুপ বানানোর কমান্ডটা হল 'groupadd'। ব্যবহারবিধি ঠিক ওই 'useradd' কমান্ডের মতই। এখানেও আলাদা আলাদা অপশান দিয়ে আলাদা আলাদা খুঁটিনাটি ভরে দেওয়া যায় সিস্টেমে। শুধু এর বেলায় চিত্রদার খাতাটা আর '/etc/passwd' নয়, এর খাতাটার নাম '/etc/group'। বার্নার গ্রুপ বানানোর কমান্ড দিন, 'groupadd burner'। তবে, মনে রাখুন, এই কমান্ড দেওয়ার আগে 'burner' নামের একটা ইউজার বানিয়ে নিতে হবে। এবার ব্যাপারটা খুব সহজ। ধরুন আপনার সিস্টেমে দুটো সিডি-ড্রাইভ আছে। তাদের ডিভাইস ফাইলদুটোর নাম দিয়েছেন '/dev/cdrom' এবং '/dev/cdwri'। এর প্রথমটা হল সিডি-রম ড্রাইভ, এবং দ্বিতীয়টা হল সিডি-রাইটার ড্রাইভ। এবার এই '/dev/cdwri' ডিভাইস ফাইলটায় 'burner' ইউজার এবং 'burner' গ্রুপের বাইরে অন্য কারো পড়ার বা লেখার অনুমতি তুলে নিলেন। 'chmod' কমান্ডটা তো জানেনই। বা, এই সিডি-বার্নারের ডিভাইস ফাইলটার মালিকানা আপনি দিয়ে দিলেন এই গ্রুপকে, 'chown' কমান্ড দিয়ে। এবং এর পরে, আপনি যাকে যাকে এই গ্রুপে আনতে চান তাদের নাম ঢুকিয়ে দিলেন এই গ্রুপে। বা, গ্রুপ সংক্রান্ত গোটা খুঁটিনাটি, কী কী গ্রুপ, কোন কোন ইউজার কোন কোন গ্রুপের মেম্বার, এই গোটাটাই যেখানে লেখা আছে সেই '/etc/group' ফাইলটা 'emacs' বা অন্যকিছু দিয়ে এডিট করে বদলে নিলেন। আগেই তো বলেছি, একটা গ্নু-লিনাক্স সিস্টেমে সবকিছু, সমস্ত কিছুকে নিজের মত করে বদলে নেওয়া যায় এই ফাইলগুলোকে বদলে। আরো একটা সমাধান আছে এই সমস্যাটার। ধরুন যদি চান, শুধু 'root' ছাড়া আর কেউ সিডি-বার্নার ব্যবহার করতে পারবে না, '/etc/fstab' বলে একটা ফাইল বদলে সেই ব্যবস্থা করা যায়। এই ফাইলটা এবং '/etc' ডিরেক্টরির এরকম আরো অনেক জরুরি কনফিগারেশন ফাইল নিয়ে পরে আমরা আলোচনা করব।

তবে বড় সিস্টেমে এত এত বেশি ইউজার এবং এত এত গ্রুপ থাকতে পারে যাদের এভাবে বদলানো খুব জটিল হয়ে পড়ে। তার জন্যে একটা কমান্ড আছে, 'groupmod', তার ব্যবহারবিধি ঠিক 'usermod' কমান্ডের মতই। কোনো একটা ইউজারকে কান ধরে নিজের সিস্টেম থেকে তুলে বাইরে ফেলে দেওয়ার জন্যে যেমন আছে 'userdel'। কোনো চালু গ্রুপকে উড়িয়ে দেওয়ার জন্যে আছে 'groupdel'। এবার ধরুন, খুব বেশি সৃষ্টিপরায়ন হয়ে আপনার সিস্টেমে গ্রুপ এবং ইউজারদের গোটাটাই ঘেঁটে ফেলেছেন। এবার আপনার সেই গ্রুপিণ্ডটা পাবেন কোথায়? ওই '/etc/group' ফাইলটা আছে। কিন্তু, খুব বড় গাবদা সিস্টেমে, প্রচুর যার ইউজার, তৃতীয় বিশ্বের মত জনবহুল, সেই

গাবদা ফাইল পড়ে ফাইলের মধ্যে থেকে কোন কোন ইউজার কোন কোন গ্রুপের মেম্বার সেটা খুঁজে খুঁজে বার করার কাজটা প্রায়ই অসম্ভব হয়ে পড়ে। সেরকম পরিস্থিতি সামলানোর জন্যেও একটা কমান্ড আছে, 'groups'। কী করে ব্যবহার করে? ... *খেজুরগাছে হাঁড়ি বাঁধো ম্যান, ম্যান পড়ার নেশা করো ম্যান* ...

'/etc' ডিরেক্টরিতে শুধু '/etc/passwd' বা '/etc/shadow' বা '/etc/group' ইত্যাদি কনফিগারেশন ফাইলই না, ইউজারের খুঁটিনাটির কাজে সমর্পিতপ্রাণ একটা আস্ত ডিরেক্টরিই থাকে। তার নাম '/etc/skel'। এই স্কেলিটন বা কংকালটা একটা টেমপ্লেট, একটা ছক, একটা ছাঁচ। 'টেমপ্লেট' (template) ধারণাটা গোটা কম্পিউটার মননেই বারবার ব্যবহৃত হয়। নানা কাজে। একটু বুঝে নেওয়া ভালো। অনেকটা সময় ধরে ভাবলাম একটা জবর প্রতিতুলনার জন্যে, তেমন কিছু মাথায় এলনা। সবচেয়ে কাছাকাছি আসছে যেটা সেটা এরকম। ধরুন একটা মেয়ে, তার একটা টেমপ্লেটের নাম 'কনে'। তখন তার পরনে লাল বেনারসি, কপালে চন্দন, গায়ে গয়না, হাতে কাজললতা, ইত্যাদি। এবং এই 'কনে' টেমপ্লেটটা মুহূর্তিক বা ইনস্ট্যানটেনিয়াসও নয়। কারণ কনে সাজার একটা দীর্ঘ শ্রমসাধ্য প্রক্রিয়া আছে, গায়ে হলুদ, চুলে হেনা, ঠোঁটে তাম্বুল-না-হলেও-রেভলন, ইত্যাদি। মানে, এই টেমপ্লেট জিনিষটা একটা প্রক্রিয়া, যার মধ্যে ঢুকবে একটা মেয়ে, বেরোবে কনে হয়ে। আবার ওই মেয়েকে যখন 'রাঁধুনি' টেমপ্লেটে দেখছি, তখন তার পরনে ছাপাশাড়ি গাছকোমর করা, চুল উড়োঝুরো, গালে একটু ময়দা, নাকে ঘাম, আঙুলে হলুদ, হাতে খুন্তি, ইত্যাদি। এখানেও একটা প্রক্রিয়া আছে। 'রাঁধুনি' টেমপ্লেটের ভিতরে সে এসেছে তরকারি কোটা, মাছে হলুদ-নুন মাখানো, ময়দা মাখা ইত্যাদি উপাদানে তৈরি একটা প্রক্রিয়ার ভিতর দিয়ে — ওই একই মেয়ে এখন রাঁধুনি। '/etc/skel' ডিরেক্টরিতে থাকে ঠিক এইরকম একটা টেমপ্লেট, একজন সদ্য যোগ-করা ইউজারের হোম ডিরেক্টরির যাবতীয় সম্ভাব্য কনফিগারেশন ফাইলের। একজন নতুন ইউজার যোগ করার সময় '/etc/skel' ডিরেক্টরির প্রতিটি কনফিগারেশন ফাইলকে কপি করে দেওয়া হয় সদ্য বানানো তার ইউজার ডিরেক্টরিতে। পরে ইউজার তার প্রয়োজন বা ইচ্ছে অনুযায়ী বদলে নেয় কনফিগারেশনগুলো। পরে দেখবেন সেসব, দশ নম্বর দিনে। নতুন ইউজার যোগ করার সময় গ্নু-লিনাক্স সিস্টেমই এই কাজগুলো করে দেয় — নতুন ইউজারের নতুন হোম ডিরেক্টরি বানানো বা সেখানে কনফিগারেশন ফাইল রেখে দেওয়া ইত্যাদি, 'useradd' কমান্ড দিয়েই আপনার দায়িত্ব শেষ। কোনো কোনো ডিস্ট্রিবিউশনে আবার একটু পার্থক্যও ঘটে। স্ল্যাকওয়ারে যেমন নতুন ইউজারের নতুন হোম ডিরেক্টরি নিজে হাতে করে বানাতে হয়, সিস্টেম নিজে বানায় না। এই কনফিগারেশন ফাইলগুলোর মধ্যেই দেওয়া থাকে বিভিন্ন সময়ে সিস্টেমের সঙ্গে এই নতুন ইউজারের সম্পর্ক কী হবে, তার লগ-ইনের পরপরই সিস্টেম কী করবে, ব্যাশ বা শেল কী করবে, তার কমান্ড প্রম্পট কী হবে, কোথায় কোথায় কোন রসদে তার অধিকার থাকবে, কোন কোন প্রোগ্রাম সে চালাতে পারবে, সে কোনো জরুরি ইউজার প্রোগ্রাম চালানোর সময়ে সিস্টেম কোন কোন ডিরেক্টরির কোন কোন ফাইল খুলবে, তার এক্স উইনডোজ চালানোর খুঁটিনাটি, ওই ইউজারের এনভায়র্নমেন্ট ভ্যারিয়েবলগুলো কী হবে, ইত্যাদি। এর একটু আধটু আমরা ইতিমধ্যেই জেনেছি। অনেকটা আমাদের জানতে হবে এই পাঠমালার আগামী অংশগুলোয়।

২।। লিংকের আগে নরম করে 'cat' আর 'touch'

ছয় নম্বর দিনের ৭ নম্বর সেকশনের ডিভাইস ফাইলগুলোকে আমাদের বুঝতে অসুবিধে হচ্ছিল বলে আমরা গিয়েছিলাম ফাইলের মালিকানা এবং অধিকারের আলোচনায়। সেটার প্রাথমিক একটা ধারণা এখন আমাদের হয়েছে। কিন্তু, এখনো, ওই ডিভাইস ফাইলগুলোকে বোঝার একটা বড় সমস্যা রয়ে গেছে, তার নাম লিংক। ছয় নম্বর দিনের ডিভাইস ফাইলের ওই দীর্ঘ তালিকায় মনে আছে, কতকগুলো বিষয় আমরা ছেড়ে এসেছিলাম? যে ফাইলগুলোর লং বা দীর্ঘ লিস্টিং শুরু হয়েছে '`l`' দিয়ে, তারা হল লিংক ফাইল। যেমন ব্লক ডিভাইসের '`b`', ক্যারেকটার ডিভাইসের '`c`', ডিরেক্টরির '`d`' এবং রেগুলার বা সাধারণ ফাইলের বেলায় নিছক '`-`'। এই লিংক ব্যাপারটাকে এবার বুঝে নেওয়ার সময় এসেছে।

লিংক ফাইলের মূল প্রয়োজনটা এইখানে যে, প্রায়ই একটা ফাইলের একই সঙ্গে দুই বা তারো বেশি জায়গায় থাকার দরকার পড়ে। নানা ধরনের প্রোগ্রাম নানা কাজে নানা জায়গায় তাদের ফাইল খোঁজে। কোন প্রোগ্রাম কোন ফাইল কোথায় খুঁজবে, গ্নু-লিনাক্সে তার একটা প্রথা বা ছক আছে, যার নাম ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড, আট এবং নয়

নম্বর দিনে সেই আলোচনায় যাব আমরা। এবার ভাবুন, কোনো একটা বিশেষ ফাইল যদি এরকম একাধিক প্রোগ্রামের সঙ্গে অন্বিত হয় যারা আলাদা আলাদা জায়গায় নিজের কাজের ফাইল খোঁজে তাহলে সেই ফাইলটাকেও একই সঙ্গে একাধিক জায়গায় উপস্থিত থাকতে হবে। এই গোটাটার মধ্যে অনেকসময় কিছু গ্নু-লিনাক্স বিবর্তনের ঐতিহাসিক কারণও থাকে। আগে একসময় প্রথা ছিল একটা বিশেষ ফাইলের একটা বিশেষ জায়গায় থাকা, পরে কাজের সুবিধের জন্যে জায়গাটা বদলানো হয়েছে। এবার, পুরোনো সময়ের প্রথায় তৈরি প্রোগ্রামগুলো সেই ফাইলটাকে পাবে কী করে? একই সঙ্গে কোনো ফাইল একের বেশি জায়গায় থাকতে পারেনা। ফাইল কেন, আমরা কেউই পারিনা। তাই, একটা জায়গায় সত্যিই নিজেকে রেখে, অন্য জায়গায় বা জায়গাগুলোয় নিজের বদলে নিজের একটা কান রেখে আসার দরকার পড়ে, পরে কান টানলেই যাতে মাথা আসতে পারে। এইসব কানবাজি না করে সরাসরি আর এক জায়গায় ফাইলটা কপি করে দেওয়ার সমস্যা এই যে কপি করা মাত্রই কপিটা তো নিজেই একটা স্বতন্ত্র ফাইল হয়ে গেল, স্বতন্ত্র হয়ে গেল তার নিজের অস্তিত্ব এবং ইতিহাস। তার মানে এখন থেকে সিস্টেমের হিশেব রাখতে হবে দুটোর আলাদা আলাদা বদলের ইতিহাসের। একটা বদলালেই অন্যটায় তার অনুরূপ বদল ঘটাতে হবে। শুধু সিস্টেম না, সিস্টেম যে চালায়, সেই সুপারভাইজর বা রুট, তার কথা ভাবুন। এবার এমন একটা ফাইল ভাবুন, যা একইসঙ্গে একাধিক ডিরেক্টরিতে আছে, আর, একটা জায়গায় একটা বদলালেই অন্যটাও বদলে যায়। মানে, একাধিক ফাইল, যাদের দুজনেরই তথ্যভাণ্ডারটা এক। একেই বলে লিংক।

কাজটা প্রথমেই একদম হাতেকলমে করে দেখে নেওয়া যাক। আমরা একাধিকবার 'cat' বা কনক্যাটেনেট কমান্ডটা ব্যবহার করেছি। মানে, যোগ করা বা সাজানো বা পেতে দেওয়া। কমান্ডটাকে ব্যবহার করেছি কিছুটা পরিমাণ টেক্সটকে পেতে দেওয়ার কাজে। কখনো পেতে দিয়েছে স্ক্রিনে, তখন কনসোলে পেতে দেওয়া ফাইলটা পড়েছি আমরা। যেমন '`cat /etc/lilo.conf`' কমান্ড দিয়ে আমরা '/etc/lilo.conf' ফাইলটাকে স্ক্রিনে পেতে নিয়ে পড়েছি। বা '`cat /etc/lilo.conf > lilo.text`' কমান্ড দিয়ে '/etc' ডিরেক্টরির 'lilo.conf' ফাইলের টেক্সটটাকে পেতে দিয়েছি 'lilo.text' ফাইলে। সেই ফাইলটা আগে ছিলনা, ফাইলটাকেই বানিয়ে দিয়েছে আমাদের 'cat' কমান্ড। এবার এই 'lilo.text' ফাইলকে আবার স্ক্রিনে পড়া যাবে, '`cat lilo.text`' কমান্ড দিয়ে। ঠিক এই টেক্সটকে ফাইলে পেতে দেওয়ার তথা ফাইল বানানোর কাজটাই করব এবার আমরা 'cat' কমান্ড দিয়ে। এবং সেই করে লিংক ফাইলের চরিত্রটা বোঝার চেষ্টা করব। গ্নু-লিনাক্সে মুহুর্মুহু ছোটখাটো দু-চার লাইনের ফাইল লিখবার কাজে 'cat' ব্যবহার করা হয়।

ফাইল বানানোর প্রক্রিয়াটা খুব সহজ। ধরুন, আপনি 'onefile' নামে একটা ফাইল বানাতে চান। আপনাকে কমান্ড দিতে হবে, '`cat > onefile`' কমান্ড লিখে এন্টার মারার পর, কমান্ড প্রম্পট আর ফেরত আসবে না। কমান্ড প্রম্পট ফেরত না-আসার মানে কী সেটা মনে আছে তো? কাজ চলছে। মানে, সিস্টেম এখনো আপনার কাজে নিয়োজিত, কাজটা চলছে, এখনো সমাপ্ত হয়নি, সমাপ্ত হলেই কমান্ড প্রম্পটটা ফেরত আসত। ফেরত আসেনি কমান্ড প্রম্পট, এবং আপনার শেষ দেওয়া এই 'cat > onefile' কমান্ডের লাইনের নিচেই কালো স্ক্রিনে দেখুন একটা কার্সর বা চিহ্নক দপদপ করছে। বোঝাচ্ছে যে, আপনি যা টাইপ করবেন সেটা এখানেই ফুটে উঠতে থাকবে। এবার, যা যা এখানে আপনি টাইপ করবেন সেটা 'onefile' ফাইলে যোগ হতে থাকবে। একটা লাইন লিখে নতুন লাইনে যাওয়ার জন্যে যেই এন্টার মারবেন, সেই নিউলাইন-চিহ্নটাও ঢুকে যাবে 'onefile' ফাইলে, পরে সে ওই ভাবে লাইন ভেঙেই দেখাবে। এই টাইপ করা টেক্সটকে ফাইলে তোলার প্রক্রিয়াটা চলতেই থাকবে যতক্ষণ না আপনি 'কন্ট্রোল-ডি' '<Ctrl><D>' , মারছেন, মানে 'কন্ট্রোল' সুইচটা টিপে রেখে 'ডি' সুইচটা টিপছেন। এবার 'কন্ট্রোল-ডি' মেরে বেরিয়ে এসে '`ls`' কমান্ড দিলেই দেখতে পাবেন 'onefile' ফাইলটাকে। এবার কমান্ড দিন '`cat onefile`'। দেখুন স্ক্রিনে ফুটে উঠেছে হুবহু যা আপনি টাইপ করেছিলেন। এখানে দেখুন, পরপর দুটো কমান্ডে দুবার আপনি 'cat' ব্যবহার করলেন। প্রথম 'cat' আপনার টাইপটাকে চালান বা রিডাইরেক্ট করে পৌছে দিল স্ক্রিন থেকে 'onefile' ফাইলে, দ্বিতীয়বার 'onefile' ফাইল থেকে স্ক্রিনে।

এই 'cat' কমান্ড দিয়েই আবার টেক্সটের সঙ্গে নতুন টেক্সট যোগও করা যায়। আগের বারের '`cat > onefile`' কমান্ডটার জায়গায় এবার নতুন কমান্ড দিন "`cat >> onefile`"। আবার টাইপ করে যান। একই ভাবে 'কন্ট্রোল-ডি' মেরে বেরিয়ে আসুন। আবার করুন '`cat onefile`', এবার দেখুন, আগের টেক্সট-এর সঙ্গে

পরেরটুকু যোগ হয়ে গেছে। এর নাম অ্যাপেন্ড বা যোগ করা। এর চিহ্ন হল '>>', যেরকম চালান করার বা রিডাইরেক্ট করার চিহ্ন হল '>'। শুধু স্ক্রিন থেকে রিডাইরেক্ট করে অ্যাপেন্ড করা নয়, সরাসরি একটা ফাইলকেও অ্যাপেন্ড করা যায়। এইমাত্র বলা প্রক্রিয়ায় আর '`cat > twofile`' কমান্ড লিখে একটা ফাইল লিখুন 'twofile'। ফাইল বানানো শেষ হওয়ার পরে, কমান্ড দিন '`cat twofile >> onefile`'। তারপর গোটাটাকে স্ক্রিনে ফুটিয়ে তোলার জন্যে কমান্ড দিন '`cat onefile`'। দেখুন, 'twofile' ফাইলের টেক্সট যোগ হয়ে গেছে 'onefile' ফাইলের মধ্যেই, পুরোনো টেক্সট-এর পরে। 'touch' দিয়ে আমরা যেমন ফাঁপা ফাইল বানানোর কায়দা শিখেছিলাম, তেমনি গ্নু-লিনাক্সে 'cat' দিয়ে টেক্সট ফাইল বানানোর সহজতম কায়দাটা আমরা এইমাত্র শিখলাম। এবার এই 'cat' আর 'touch' ব্যবহার করব লিংক বোঝার কাজে।

২.১।। সিম্বলিক বা সফট লিংক

প্রথমে টাচ করুন 'onefile' নামের একটা ফাঁপা ফাইলকে। কমান্ড দিন '`touch onefile`'। এবার কিছু আলাদা করে না বুঝেই পরের কমান্ডটা দিন, পরে আমরা এর মানেটা বুঝব, '`ln -s onefile twofile`'। কমান্ড প্রম্পট তো ফেরত এলো তার কাজ শেষ করে, কিন্তু এই কমান্ডে কাজটা কী হল সেটা তো বুঝতে হবে, ডিরেক্টরির হালহদিশ বোঝার কমান্ড তো আছেই, '`ls -l`'। দিন, স্ক্রিনে দেখুন, পরিষ্কার দুটো ফাইলকে দেখাচ্ছে। তার মানে আমাদের ওই না বোঝা কমান্ডটা আমাদের টাচ করা 'onefile' ফাইলটা ছাড়াও একটা নতুন ফাইল তৈরি করেছে 'twofile' নামে। এবার এদের দুজনের বাইট সাইজটা দেখুন তো? চমকালেন? এর মানে কী? 'onefile' দেখাচ্ছে ০, অথচ 'twofile' দেখাচ্ছে ৭? 'onefile' শূন্য, এর মানেটা বুঝতে পারছি, কিছুই রাখিনি সেখানে। শূন্য ফাঁপা নামসর্বস্ব, যাকে জরুরি অবস্থার পরের ফেজে আমরা ডাকতাম সিদ্ধার্থ। চায়ের দোকানে গিয়ে বলতাম, মেনোদা, দুটো চা একটা সিদ্ধার্থ। মানে, দুটোকে তিনটে করো। মধ্যের বছরগুলোয় কিশোর সিদ্ধার্থ ম্যাচিওর করেছে, পরিণত হয়েছে। এখন বোধহয় বলা যায়, দুটো চা একটা বুদ্ধদেব। তাই 'onefile' শূন্য ঠিক আছে, কিন্তু 'twofile' ফাইল সাত দেখাচ্ছে কেন? ফিরে আসব এটায়, একটু পরে, মাথায় রাখুন।

আমরা জানি 'cat' দিয়ে কী করে নতুন ফাইলে টেক্সট ভরতে হয়, ভরুন আপনার মনোমত টেক্সট মনোমত ফাইলে, যত খুশি, কিন্তু সাবধান, গ্নু-লিনাক্স বিষয়ে কোনো খারাপ কথা লিখবেন না। এটা গ্নু-লিনাক্স সিস্টেমের একটা বৈশিষ্ট্য যে, আপনি যদি প্রকৃত বিশ্বাস থেকে সিস্টেমের বিষয়ে কোনো খারাপ কথা লেখেন আপনার স্ক্রিনে, সঙ্গে সঙ্গে আপনার মনিটরটা সাউন্ডবক্স হয়ে যাবে আর মড়াকান্না কাঁদতে থাকবে, হার্ডডিস্ক পুরো ফরম্যাট করে ফেলার আগে অব্দি। এটা মাথায় রেখে '`cat >> onefile`' কমান্ড দিয়ে এইমাত্র বানানো 'onefile' ফাইলে টেক্সট ভরুন, আর ভরা শেষ হলে 'কন্ট্রোল-ডি' মেরে বেরিয়ে আসুন। আমরা জানি, এখন স্ক্রিনে '`cat onefile`' কমান্ড দিয়ে আমরা এইমাত্র টাইপ করা গোটা টেক্সটটা দেখতে পাব। কিন্তু, একই কাজ কবার করতে ভালো লাগে মানুষের? তার চেয়ে কমান্ড দিন, '`cat twofile`', দেখুন তো কী ঘটল? স্ক্রিনে আপনি দেখছেন হুবহু 'onefile' ফাইলে ঢোকানো টেক্সটটা। হ্যাঁ, ঠিক এটাই ঘটার কথা ছিল। দুটো আলাদা ফাইল, তাদের তথ্যভাণ্ডারটা এক। ঠিক এটাই করে '`ln -s`'। আসলে '`ln`' হল কমান্ড, আর '`-s`' তার অপশান। 'ln' কমান্ডটা একটা ফাইলের লিংক তৈরি করে, মানে, অন্য এমন একটা ফাইল তৈরি করে যা প্রথম ফাইলের তথ্যটাই বহন করে, তাতে কোনো বদল এলে লিংক ফাইলটাও বদলে যায়। এবং একবার '`ls -l`' মেরে দেখুন, দীর্ঘ তালিকায়, একদম বাঁদিকের জায়গাটায়, 'onefile' আর 'twofile' ফাইলদুটোর দুটো আলাদা অক্ষর। 'onefile'-এর পাশে '-', আর 'twofile'-এর পাশে '`l`'। এর মানেটা মনে করুন। 'twofile' হল একটা লিংক, সিম্বলিক লিংক, যার নিজের শরীরে কিছুই নেই, শুধু একটা প্রতীক। যখন আমরা এই প্রতীক বা লিংক ফাইলকে কিছু করছি, ক্যাট বা ডগ বা যা-খুশি, তখন আসলে তাকে কিছুই করছি-না, তার শরীরে পাওয়া লিংক বেয়ে পৌঁছে যাচ্ছি সেই আসল ফাইলে। যা ঘটাচ্ছি, সবই ঘটছে মূল ফাইলে। লিংক ফাইলের নিজের বলে কিছু নেই, তোমারি গরবে গরবিনী আমি, রূপসী তোমারি রূপে। 'ls -l' কমান্ডের কাজ আমাদের এখনো ফুরিয়ে যায়নি, এখনো একটা চমক বাকি আছে, ফাইলদুটোর বাইটসাইজ দেখুন তো? 'onefile' কত তা আমি বলতে পারব না, যতটা আপনি লিখেছেন। কিন্তু 'twofile' কত তা আমি নিশ্চিতভাবেই জানি, ৭। কী করে? আচ্ছা, আর একটা ফাঁপা ফাইল বানান তো। আপনার তো কোনো ব্যাপার না, টাচ করে করে টাচউড হয়ে

গেছেন, ছুঁলেই ফাইল। শুধু এবার তার নাম দিন 'threefile', এবার তাকে '`ln -s`' করে আর একটা ফাইল বানান, তার নাম দিন 'fourfile'। আবার '`ls -l`' করুন। 'threefile' বাইটসাইজ যথারীতি পাচ্ছেন ০, 'fourfile' কিন্তু পাচ্ছেন ৯, আগেরবারের মত ৭ নয়। এবার 'threefile' ফাইলে টেক্সট ভরুন, ভরে দেখুন। যত ভরবেন সেই অনুযায়ী তার বাইটসাইজ বাড়বে, কিন্তু 'fourfile' সেই ৯ থেকে যাবে চিরকাল। এই সাইজটা আমি বলছি কী করে? বুঝতে পারছেন? না-পারলে শূন্য নম্বর দিনের একদম শেষ দিকের টেক্সট-তথ্যের সাইজের জায়গাটা আর একবার পড়ে আসুন। আচ্ছা, আপনাকে একটু হেল্প করি, দেখুন তাতে পারেন কিনা। এবারে আরো দুটো ফাইল বানান, ওই একই ভাবে, প্রথমে টাচ, তারপর লিংক। এবারে মূলটার নাম দিন 'fivefile' আর লিংকটার 'sixfile'। এবারেও, 'fivefile' অবভিয়াসলি পাবেন ০, কিন্তু 'sixfile' পাবেন ৮। *কেন এমন হয়, দেব আপল্লন, কেন এমন হয়?*

শুধু ফাইলই না, সিম্বলিক লিংক তৈরি করা যায় ডিরেক্টরিরও। ঠিক একই ভাবে, তখন ওই সিম্বলিক লিংক ফাইলের উপর আমাদের যে কোনো কাজ আসলে ঘটবে সেই মূল ডিরেক্টরির উপর। ধরুন আপনার ডিরেক্টরি 'onedir'। এর সিম্বলিক লিংক বানালেন 'twodir'। এবার এই 'onedir', মূল ডিরেক্টরির মধ্যে একটা ফাইল বানান, ফাঁপা ফাইল হলে 'touch', আর দোকানে মাল রাখতে চাইলে 'cat'। এবার '`ls -l twodir`' করে দেখুন, 'onedir' ডিরেক্টরিতে এইমাত্র বানানো ফাইলের খুঁটিনাটি অবিকল দেখিয়ে দিচ্ছে তালিকায়। আমাদের কাছে 'twodir' আসছে মূল 'onedir' ডিরেক্টরির একটা হুবহু প্রতিলিপি হয়ে। কিন্তু আসলে এটা কোনো প্রতিলিপি না, আমরা 'onedir' ডিরেক্টরিকেই দেখছি, শুধু অন্য নামে।

এই নানা নামে ডাকা গোলাপের গল্প নিছক ফুল ভালোবাসার বৃত্তান্ত নয়, এর নানা জরুরি ব্যবহার আছে। দু-একটা আমরা দেখব। অন্যগুলো দেখতে থাকবেন, সিস্টেমে কাজ করতে করতে। যেমন ধরুন, আপনার দুটো সিডি-ড্রাইভ আছে, একটা বার্নার, আর একটা নিছক রিডার। আপনি বার্নারটাকে শুধু সিডি পোড়ানোর কাজে ব্যবহার করতে চান, আর এমনি রিডারটাতেই ভিউসিডি অডিয়োসিডি ইত্যাদি চালাতে চান। দুটো সিডিড্রাইভ, সেটা যদি স্কাসি হয় তাহলে, ছ নম্বর দিনে দেখেছি আমরা, তার নাম '/dev/scd0' আর '/dev/scd1'। সচরাচর এর প্রথমটাই হয় বার্নার ড্রাইভ, কারণ, মাদারবোর্ডের মাস্টার সংযোগে বার্নার আর স্লেভ সংযোগে রিডার না-লাগালে কিছু অসুবিধা হতে পারে, তার মানে '/dev/scd0' হল বার্নার ড্রাইভ, সিডি পোড়াতে ব্যবহার করেন, আবার চালানোও যায়। আর '/dev/scd1' হল সেই ড্রাইভ যেটা আপনি শুধু সিডি চালানোর জন্যে ব্যবহার করেন, ভিউসিডি, অডিয়ো-সিডি ইত্যাদি। এবার ধরুন, আপনি এমপ্লেয়ার বা যেকোনো ইউজার প্রোগ্রাম চালাচ্ছেন, যার ডিফল্ট সেটিং করা আছে '/dev/cdrom'। মানে, আপনি যদি আলাদা করে কোথা থেকে ভিউ-সিডি চালাতে হবে সেটা বলে না-দেন, তাহলে 'mplayer' নিজে নিজে যে ডিভাইস ফাইলটা খুঁজবে সেটা হল '/dev/cdrom'। সেখানে ভিউ-সিডি পেলে তাকে চালাবে। তার মানে। প্রতিবার আপনাকে আলাদা করে 'mplayer'-কে বলে দিতে হবে আপনার উদ্দীষ্ট সিডি-ড্রাইভের ঠিকানা। তার চেয়ে একটা সিম্বলিক লিংক বানিয়ে দিন। কমান্ড দিন 'ln -s /dev/scd1 /dev/cdrom'। এখন থেকে সিস্টেম যখনই '/dev/cdrom' খুঁজবে, পাবে '/dev/scd1'। এবার ছ নম্বর দিনের ডিভাইস ফাইলের দীর্ঘ-তালিকা বা লং-লিস্টিং থেকে দেখুন তো এই কায়দাটা আপনি বুঝতে পারছেন কিনা, তখন যেটা আমরা ছেড়ে এসেছিলাম? লিংক ফাইলের মানে বুঝতে পারছেন? পরে দেখবেন, যখন আমরা মাউন্ট ব্যাপারটা ভালো করে বুঝব, মাউন্টের ব্যাপারেও খুব কাজে লাগে সিম্বলিক লিংক। এবং সিস্টেমে গোড়া থেকেই কোথায় কোথায় কোন ফাইলের বা ডিরেক্টরির সিম্বলিক লিংক থাকবে তার ছকটা করা থাকে গ্নু-লিনাক্সে ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডের মধ্যেই।

এখানে একটা কথা বলে রাখি, কোন ডিভাইস ফাইলটা আপনার কোন সিডি-ড্রাইভ সেটা প্রথম প্রথম বুঝতে অসুবিধে হয় খুব। এর জন্যে একটা বাংলা উপায় আছে, ব্রুট ফোর্স, '/dev' ডিরেক্টরিতে একের পর এক ডিভাইস ফাইলের নাম দিয়ে 'eject' কমান্ড দেওয়া। '`eject /dev/scd0`', '`eject /dev/sr1`', ইত্যাদি, কোনটায় শেষ অব্দি আপনার আজি সিডিরো দুয়ার খোলা হচ্ছে, সেটাই আপনার ডিভাইস ফাইল। আরো উপায় আছে, সেগুলো আপনি এখনি বুঝবেন না পুরোটা, তবু বলে রাখছি। ইচ্ছে করেই করছি এটা এই পাঠমালায়, প্রায়ই, যাতে সবসময়েই বোঝা আর না-বোঝার টানাপোড়েনটা রয়ে যায়, শোনা যায় শেখার নাকি একটা ভালো প্রথা সেটাই। একটা উপায় খুব ব্যবহার হয়, বিশেষত সিডি বার্নার ড্রাইভ থাকলে। পোড়ানো শুরু করার আগে সেটা করে নিতেই

হয়, সেটা হল ড্রাইভ খোঁজার কমান্ড, 'cdrecord -scanbus'। 'cdrecord' হল কমান্ড, সিডি পোড়ানোর একটা চমৎকার সফটওয়ার। বিশেষত, আমার মত যারা আগে উইনডোজের জন্যে তৈরি নেরো ব্যবহার করে অভ্যেশ গড়ে উঠেছিল, তাদের আরো অবাক লাগে, এতকিছু করা যায়, নিজের খুশিমত? এখানে 'cdrecord' কমান্ডের সঙ্গে 'scanbus' অপশানটা আমরা ব্যবহার করছি, যাতে বাস স্ক্যান করে ও স্কাসি ড্রাইভগুলো খুঁজে নিতে পারে। আগেই বলেছি, সিডি বার্নার ড্রাইভ এমনকি স্কাসি না-হলেও তাকে স্কাসি এমুলেট করা হয়, মানে, যেন সেটা স্কাসি ড্রাইভ এইরকম ভাবে সিস্টেমে ব্যবস্থা করে নিতে হয়, '/etc/lilo.conf' ফাইলে 'hdc=ide-scsi' বা 'hdd=ide-scsi' লাইন যোগ করে, যেন যথাক্রমে '/dev/hdc' বা '/dev/hdd' একটা স্কাসি ড্রাইভ। তবে এটা প্রয়োজন হয় গ্নু-লিনাক্স কারনেল ভারশন ২.৬-এর আগে অব্দি। এই মজাটা দেখুন, আমি এই বই লিখতে লিখতেই ক্রমে কারনেল ২.৬ ব্যবহার হতে শুরু করল। এতদিন এই স্কাসি এমুলেট করতে হত, এখন আর হবেনা। তবে 'cdrecord' খুঁজে দেয় শুধু স্কাসি ড্রাইভ বা স্কাসি এমুলেটেড ড্রাইভকে। যে কোনো সিডি-ড্রাইভ, সে স্কাসি হোক, আইডিই হোক, যাই হোক, তাকে খোঁজার কাজ খুব ভালো করে দেয় আর একটা প্রোগ্রাম, তার আসল কাজ অডিও সিডি থেকে গান পড়ে সেটাকে ডিস্ক-ফাইল বানানো, সচরাচর ওয়াভ ফাইল বা '*.wav' ফরম্যাটে। প্রোগ্রামটা হল 'cdparanoia'। সিডি-ড্রাইভটা খোঁজার অপশান সহ গোটা কমান্ডটা দাঁড়ায়, 'cdparanoia -vsQ'। পরপর '/dev' ডিরেক্টরির যাবতীয় ডিভাইস ফাইল খুঁজে খুঁজে আপনাকে দেখিয়ে যেতে থাকবে। তবে, দুটো কমান্ডই চালাবেন 'root' হয়ে। এবং একটু আগে সিডি পোড়ানোর নিয়ন্ত্রিত অধিকারের সূত্রে যে 'burner' নামের ইউজার এবং গ্রুপ বানানোর কথা বলেছিলাম, তার জায়গায় আর একটা সহজ জিনিষ করা যায়। সেটা হল, যে ডিভাইস ফাইলটা আপনার বার্নার ড্রাইভটার, তার উপরে অধিকারটা নিয়ন্ত্রণ করা। চটজলদি আমি বহুবার এটা কাজে লাগিয়েছি। 'v', 's', 'Q' ইত্যাদি অপশনগুলোর অর্থ? ম্যান পড়ুন, ম্যান। আর একটা ভালো হাউ-টু আছে, গ্নু-লিনাক্স সিস্টেমেই দেওয়া, সিডি-বার্নিং নিয়ে, সেটও পড়ে দেখতে পারেন, বোঝার জায়গাটাও বাড়বে, এমনকি সিডি পোড়ানোর দরকার যদি নাও থাকে।

২.২।। সফট লিংক আর হার্ড লিংক

গ্নু-লিনাক্সে একই ফাইল একটা ফাইলসিস্টেমের একাধিক জায়গায় একাধিক নামে থাকতে পারে, যেখানে তথ্যের ভাণ্ডারটা একই শুধু নানা নাম তাদের, একে বলে হার্ড লিংক। সিম্বলিক লিংকের সঙ্গে এর মিল আছে, কিন্তু পার্থক্যও আছে। দুই রকম লিংক-কে একসঙ্গে পেড়ে ফেলা যাক। শুধু, সরি, এটার জন্যে একটু হুবহু টাইপ করতে হবে, যারা টাইপ করতে খুব একটা অভ্যস্ত না, তাদের টাইপ করার সময় একটা কেলো হয় এই যে, বর্ণমালাগুলো ঠিকঠাক আসে, কিন্তু যতিচিহ্ন আর স্পেসগুলো ঠিকঠাক খেয়াল থাকেনা। এখানে একটা বাড়তি যতিচিহ্নও খেয়াল রাখবেন, নিউলাইন মানে নতুন-লাইন, সচরাচর টেক্সট-এর ভিতর আমরা এন্টার বা ক্যারেজ-রিটার্ন মেরে যা করি — যেখানে লাইন শেষ করে এটা মারতে হবে সেখানে আমি বলে দেব।

প্রথমে টাচ করে 'onefile' বানান, 'touch onefile'। এবার দুরকমের দুটো লিংক বানান, প্রথমটা সিম্বলিক বা সফট লিংক 'twofile', কমান্ডটা তো ইতিমধ্যেই জানেন, 'ln -s onefile twofile'। এবার দ্বিতীয় একটা লিংক বানান, 'threefile'। এটাকে করুন হার্ড-লিংক, তার জন্যে কমান্ড দিন 'ln onefile threefile'। এবার একবার 'ls -il' মারুন। এর আগে নানা কাজে আমরা '-al' এবং '-l' এই দুটো অপশান সমাহার ব্যবহার করেছি। এর মধ্যে 'l' হল লং-লিস্টিং বা দীর্ঘ-তালিকার জন্যে, আর 'a' হল সব বা অল ফাইল দেখানোর অপশান, আপাতগোপন ডটানন ফাইলগুলোকেও যাতে দেখায়। এখানে নতুন আসা 'i' অপশানটা কী জন্যে দেওয়া সেটা আমরা এখুনি দেখতে পাব। যে তালিকাটা ফুটে উঠবে সেটা এইরকম, শুধু বোঝার সুবিধের জন্যে পরের দুটো লাইন আগে পরে করে দিয়েছি, নামের বর্ণমালা অনুযায়ী এলএস তো থ্রিফাইলকে টুফাইলের আগে এনেছিল।

```
12623010 -rw-r--r-- 2 dd users 0 2003-12-24 10:49 onefile
12623012 lrwxrwxrwx 1 dd users 7 2003-12-24 10:50 twofile -> onefile
12623010 -rw-r--r-- 2 dd users 0 2003-12-24 10:49 threefile
```

এরমধ্যে একদম বাঁদিকের ওই বাড়তি গাবদা নম্বরটা, আইনোড নম্বর, সেটা দেখানোর জন্যেই দেওয়া ওই 'i' অপশানটা। আসছি আমরা আইনোডের কথায়। এখানে দেখুন, সফট লিংক বা 'twofile'-এর গল্পটা আমাদের চেনা,

ঠিক এরকমই এসেছিল আগেও, বাইটসাইজ ৭ , আর বাঁদিকে লেখা '`1`', মানে, এটা একটা লিংক। কিন্তু, হার্ড-লিংক বা 'threefile' দেখুন, কিছু মেলাতে পারছেন? সবকিছুই 'onefile'-এর মত, যেন নিজেই একটা স্বতন্ত্র স্বাধীন ফাইল, সব কিছুই এক। মানে, প্রায় এক। শুধু একটা জায়গা বদলে গেছে, খেয়াল করতে পারছেন? দেখুন, একটু আগেই যেটা বললাম, লিংকের সংখ্যা, সেটা বদলে গেছে। 'onefile' আর 'threefile' দুটোর বেলাতেই এই সংখ্যাটা আগে ছিল ১, এখন এসেছে ২। আরো একটা জিনিষ দেখুন, সমস্ত দিক থেকে সমস্ত অর্থে, এমনকি সময়চিহ্নের নিরিখেও, 'onefile' আর 'threefile' হুবহু এক। মজাটা খেয়াল করুন। আমরা প্রথমে বানালাম 'onefile', তার পরেই 'twofile', সবার শেষে 'threefile'। সময়ের স্ট্যাম্পটা দেখুন, 'twofile'-এর বেলায় '`10:50`', 'onefile'-এর '`10:49`' থেকে একমিনিট পরে, ঠিকই আছে, এরকমটাই হওয়ার কথা।। কিন্তু গোলমালটা আছে 'threefile'-এ। সে 'onefile'-এর লিংক, কিন্তু সে তার নিজের বাপের মানে 'onefile'-এর সঙ্গে একই ডেট-অফ-বার্থ টাইম-অফ-বার্থ শেয়ার করছে। কেলেংকারির একশেষ। আসলে বাপ-ব্যাটার সম্পর্ক নয়, সম্পর্কটা একদম হুবহু প্রতিরূপের। এবার আসুন একদম বাঁদিকের আইনোড নম্বরে, সেটা কী বস্তু মনে আছে? ছয় নম্বর দিনে উল্লেখ আছে, আজ এবং আট নম্বর দিনে আরো আসবে। আমরা একটা ফাইলকে চিনি তার নাম দিয়ে, কিন্তু ওএস বা অপারেটিং সিস্টেম তাকে চেনে তার আইনোড নম্বর দিয়ে। খেয়াল করুন, 'onefile' আর 'threefile'-এর আইনোড নম্বরটাও এক। 'twofile'-এর বেলায় তা নয়। ঠিক একই জিনিষ দেখুন অনুমতির কাঠামোয়। 'chmod' দিয়ে বদলে বদলে দেখুন, 'onefile' আর 'threefile' এই দুই ফাইলের সবকিছুই এক মানে এক মানে এক।

এবার দেখা যাক 'onefile'-এ কিছু তথ্য যোগ করে। প্রথমে কমান্ড দিন, '`cat > onefile`', চেনা কমান্ড, দিয়ে এন্টার মারুন। দেখুন কমান্ড প্রম্পট হাওয়া হয়ে গেছে, নিচের লাইনে কার্সরের চৌকো আলোটা দপদপ করে আপনাকে ডাকছে, এসো এসো টাইপ করো। সত্যিকারের লিনাক্সী হয়ে ওঠার প্রমাণ হল যখন আপনি ডাকটা কানেও শুনতে পাবেন, তার জন্যে দরকার রোজ জিটাইপিস্ট (Gtypist) নিয়ে প্রাকটিস করা। গ্নু-র একটা ফ্রি সফটওয়ার। টাইপ শেখার ইশকুলে যা শেখায় তার চেয়ে তেরো হাজার পাঁচশো ছেচল্লিশ গুণ ভালো করে শেখায়। আর বকে দেওয়াটা যা কিউট, সে আর কহতব্য নয়। যাকগে, 'cat'-এর কথায় আসি। ওই চৌকো আলোর জায়গাটায় আপনি হুবহু টাইপ করুন, স্পেস মানে স্পেস, অক্ষর মানে অক্ষর, একদম ঠিক এটাই, এর একটা কারণ আছে, এখনি বুঝতে পারবেন, '`onefile is one file`'। এবার এন্টার মারুন। মানে নিউলাইন, নতুন লাইনে চলে গেছে আমন্ত্রণী কার্সর। এবার আপনি জানেন কী করে ফাইলটা গোটাতে হবে, 'কন্ট্রোল-ডি' মেরে। মানে, 'Ctrl' সুইচটা টিপে রেখে 'D' সুইচটা মেরে, তারপর দুটোই ছেড়ে দিয়ে। এবার একবার '`cat onefile`' করুন, হুবহু এই টেক্সটটাই স্ক্রিনে দেখাচ্ছে। আপনি নিজের মত, আপন টাইপমাধুরী মিশায়ে অন্য কিছু ভরতেই পারতেন ফাইলটায়, কিন্তু তাহলে টেক্সট-এর চিহ্নের হিশেবটা আবার বদলে যেত। হিশেবটা এখুনি আমাদের দরকার পড়বে। তার আগে একটু কুচো করে শিখে আসা যাক, ব্যাশ কী ভাবে আপনার ইতিহাসটাকে লিখে রাখে আপনারই অগোচরে। সেই ইতিহাসকে কী করে প্রয়োজন মাফিক ফেরত পাওয়া যায়। আপনার ইতিহাস মানে আপনার আদেশের ইতিহাস।

২.৩।। আদেশের ইতিহাস

কিবোর্ডের তিনখাবলা চাবিসংস্থানের মধ্যে একদম ডানদিকেরটা নিউমেরিক কিপ্যাড। অব্যবহারের ধুলো জমে জমে এমন কালো হয়ে যায় যে মাসআটেক বাদে বাদে যখন সুইচগুলো খুলে সাবানজলে ভেজাই, ওগুলোকে আলাদা করে স্কচব্রাউট দিয়ে মাজতে হয়। একদম বাঁদিকের মূল অংশটা হল সেই জায়গাটা আমরা প্রায় সব টাইপই যেখানে করি। উপরে সংখ্যার সারি, নিচে বর্ণমালা আর নানা চিহ্ন। আর এই দুইয়ের মাঝে এক খাবলা সুইচ। উপরে ছটা, নিচে চারটে। নিচের চারটে হল অ্যারো সুইচ, উপরে নিচে ডানে বাঁয়ে চার পিস তীরচিহ্ন। '→', '←', '↑', '↓'। এবার, এই চারটে অ্যারো সুইচের মধ্যে চাতক সুইচটিকে, মানে যে অ্যারোটা উপরপানে মুখ করে আছে, '↑', ক্রিচ করে টিপতে হবে একবার করে, থেমে স্ক্রিন দেখতে হবে। একবার সুইচ টেপাটা সবসময় খুব সহজ হয়না। রিপিট-রেট (repeat-rate) এবং ডিলে-বিফোর-ফার্স্ট-রিপিট (delay-before-first-repeat) কত করা আছে, মানে, সুইচ টিপে ধরে রাখলে একটা নির্দিষ্ট সময়ে সিস্টেম সেটাকে কতবার টেপা বলে ভাববে, এবং টিপে রাখলে প্রথমবার টেপাটাকে নেওয়ার কত সময় বাদে পরের টেপাগুলোকে গুনতে শুরু করবে, তার উপরে নির্ভর করে, অনেকসময়

বেশ ঝামেলা পাকায়। একবার টিপছি ভাবলাম, এদিকে চলে গেল হয়ত গোটা পাঁচেক। একবার কিবোর্ডে চা পড়ে গিয়ে সে কী কেলেঙ্কারি। 'G' আর 'T' সুইচটা একবার টিপলেই কয়েকখানা করে হচ্ছে। পরে খুলে রোদ্দুরে শুকিয়ে তবে ঝামেলা মিটল।

আপনি যদি এখন ঠিক একবার ক্রিচ করে থাকেন, মানে, একবার টিপে থাকেন, তাহলে দেখুন, এখন স্ক্রিনে আছে, 'cat onefile'। ক্রিচ ক্রিচ মানে দুবার করে থাকলে, স্ক্রিনে আছে 'cat > onefile'। তিন ক্রিচ ক্রিচ ক্রিচে 'ls -il'। আরো ক্রিচী মানুষ হতে চাইলে পরপর পেতে থাকবেন, 'ln onefile threefile', প্রথমবার। তারপর আসবে 'ln -s onefile twofile', তারপর 'touch onefile'। এইভাবে চলতেই থাকবে। আজ যা করেছেন, তার আগে যা করেছেন, তার আগে, তার আগে, গতজন্মেও চলে যেতে পারে, যদি আপনি তখন থেকে গ্নু-লিনাক্স শুরু করে থাকেন, এবং গতজন্ম থেকে এই অব্দি এক হাজারের বেশি কমান্ড না-দিয়ে থাকেন। কারন কমান্ড মনে রাখার এই প্যাঁচটা হল এক হাজারি মনসবদার, মানে ডিফল্টে একটা গ্নু-লিনাক্স সিস্টেমে প্রতি ইউজারের শেষ এক হাজার কমান্ড তুলে রাখা হয়। এর জমিদারি আরো বাড়ানো যায়, কী করে সেটা পরে শিখবেন। আমি নিজে যেমন, বয়েস হয়ে গেছে, খুব বেশি কমান্ড মনে রাখতে পারিনা, খোঁজার সুবিধের জন্যে তাই এটাকে দশহাজার করে রেখেছি।

আগে দেওয়া পুরোনো কমান্ড ফিরে দেখার এই দুঃসাধ্য ক্রিচকৌশলের আরো একটা সহজতর উপায় আছে ব্যবহৃত কোনো একটা বিশেষ কমান্ড খুঁজে পাওয়ার। 'কন্ট্রোল-আর', মানে 'Ctrl' সুইচটা টিপে রেখে 'R' সুইচটা মেরে, তারপর দুটোই ছেড়ে দেওয়া। এবার যে কমান্ডটা আপনি খুঁজছেন, ধরুন 'ls', সেটাকে টাইপ করে দিন, 'l' মারা মাত্র শেষ যে কমান্ডে আপনি 'l' দিয়ে শুরু করেছেন সেটা চলে আসবে স্ক্রিনে, তারপর 's' মারা মাত্র শেষ যে কমান্ডে আপনি 'ls' দিয়ে শুরু করেছেন সেটা চলে এসেছে। এবার আপনি ধরুন দেখলেন যে এটা নয়, এরও আগে অন্য নানা অপশানের সঙ্গে আপনি যে 'ls' কমান্ডটা ব্যবহার করেছিলেন, সেইটা পেতে চান, কোই বাত নেহি, আর একবার 'কন্ট্রোল-আর' (Ctrl-R)মারুন, তার আগেরটায় যেতে চাইলে আরো একবার। এই করতে করতে যেই ঠিক কাঙ্খিত কমান্ডটায় পৌঁছলেন তখন একবার ক্রিচ করে দক্ষিণপন্থী অ্যারো সুইচটা, '→', টিপুন। দেখবেন, ওই কমান্ডটা চলে এসেছে স্বাভাবিক কমান্ড প্রম্পটে। এখন এন্টার মারলেই কমান্ডটা কাজ করবে, যেন এইমাত্র আপনি টাইপ করে দিয়েছেন। এবার ধরুন দেখলেন যে কমান্ডটা মোটামুটি ঠিক আছে, একটু বদলাতে হবে। বামপন্থী বা দক্ষিণপন্থী তীরেরা আছে কি করতে। ওই দিয়ে কার্সর এগিয়ে পিছিয়ে ঠিক জায়গায় নিয়ে আসুন, টাইপ করুন যা যা বদল আনতে চান পুরোনো কমান্ডটায়। আর মোছার দরকার পড়লে, ডিলিট(Delete) মারলে কার্সরে থাকা অক্ষরটা মুছে যায়, ব্যাকস্পেস (Backspace) মারলে কার্সরের বাঁদিকের অক্ষরটা মুছে যায়। কখনো কখনো একটা কমান্ড হতে পারে বেধড়ক লম্বা, গ্নু-লিনাক্সে কমান্ড প্রম্পটে তিন চার লাইনের কমান্ড হরহামেশাই ঘটে, তখন ডান বা বাঁ তীর মেরে মেরে পৌছনোটা ঝামেলার। কমান্ডটার একদম গোড়ায় চলে যাওয়া যায় 'কন্ট্রোল-এ' (Ctrl-A) মেরে, একদম শেষে চলে যাওয়া যায় 'কন্ট্রোল-ই' (Ctrl-E) মেরে। কমান্ড লাইনের এরকম আরো অনেকগুলো তরকিব আছে, খুবই সুবিধেজনক, তবে সেগুলো কাজে লাগানোর আগে আর একটু কাজের জটিলতা বেড়ে ওঠা দরকার। টাইপ করা একটা কমান্ড স্মৃতিতে তুলে রাখতে চান তো তারও উপায় আছে। 'কন্ট্রোল-কে' (Ctrl-K) মারলে কার্সরের ডানদিকের কমান্ডের গোটা অংশটা কেটে স্মৃতিতে তুলে নেওয়া হয়, আর 'কন্ট্রোল-ইউ' (Ctrl-U) মারলে কার্সরের বাঁদিকের গোটা অংশটা। এবার যেখানেই দরকার পড়বে, 'কন্ট্রোল-ওয়াই' (Ctrl-Y) মারুন, কেঁটে নেওয়া অংশটা সেঁটে দেবে ব্যাশ। এই সেশন থেকে বেরিয়ে যাওয়ার আগে অব্দি ব্যাশ আপনার শেষ কেটে নেওয়া অংশটা মনে রেখে দেবে। এরকম আরো অনেক আছে। ভালো করে জানতে হলে ম্যান ব্যাশ, 'man bash' করুন। আর এগুলো সব অভ্যেশ করে নিন, এখুনি, নইলে হারগিস মনে থাকবে না।

আপনার আদেশের ইতিহাসের দলিলটা একটু চিনে নিন। খুঁটিনাটি পরে লড়ানো যাবে, যখন আমরা কাস্টমাইজেশন নিয়ে কথা বলব, দশ নম্বর দিনে গিয়ে। কাস্টমাইজেশন মানে আপনার কাস্টমের বা প্রথার সঙ্গে মিলিয়ে সিস্টেমকে নিজের মর্জিমাফিক করে নেওয়া। মানে, আপনার পাঁঠা আপনি কোথায় কাটবেন, লেজে না শিঙে, লেজ ক-পিস, শিঙ কী সাইজের চোকলায়, মাংস কোথায় যাবে, বদরক্ত কোথায়, ইত্যাদি। এর মূল জায়গাটা হল কনফিগারেশন ফাইল আর এনভায়র্নমেন্ট ভ্যারিয়েবলগুলো। আপনার হোম সুইট হোম ডিরেক্টরিতে দাঁড়িয়ে একবার 'ls' মারুন। একটু

জোরে মারুন। না না, কী-বোর্ডে জোরে মারবেন না, একটু জোরালো 'ls' মারুন, '`ls -a`', যাতে আপাতগোপন ডটানন ফাইলগুলোকেও দেখায়, যাদের নামের গোড়াতেই একটা ডট মানে বিন্দু মানে '.' থাকে। শুধু ফাইল না, ডটানন ডিরেক্টরিও হয়। ফলাফলটা হবে অনেকটা এরকম। এটা আমার মেশিনের সিস্টেমে আমার নিজের, মানে 'dd'-র হোমে পাওয়া। তালিকাটা একটু বদলে নিলাম, দেখানোর সুবিধের জন্যে, আর, '`ls -a`' সব ফাইল আর ডিরেক্টরিগুলোকেই দেখিয়েছিল, তার থেকে আমি শুধু ডটানন ফাইল আর ডিরেক্টরিগুলোকে তুলে নিয়েছি। স্বাভাবিক গুলো বাদ দিয়ে দিলাম, যেগুলো আপনি এমনিতেই '`ls`' মেরে দেখতে পেতেন। আর এর মধ্যে ডিরেক্টরিগুলো ব্লক করে দিলাম বোঝার সুবিধের জন্যে।

.AbiSuite	**.ICEauthority**	.X.err	.Xauthority	.Xdefaults
.Xmodmap	.Xresources	**.adobe**	.bash_history	.bashrc
.blackboxrc	.cdrdao	.dvipsrc	.emacs	**.emacs.d**
.esd_auth	.exrc	**.fltk**	**.fonts**	.fonts.cache-1
.fonts.conf	**.gconf**	**.gconfd**	**.gnome**	**.gnome2**
.gnu-emacs	.gtkrc-kde	**.kde**	.kderc	.kermrc
.links	.mailcap	**.mcop**	.mcoprc	.mime.types
.mozilla	**.mplayer**	.muttrc	.padminrc	.profile
.qt	.recently-used	.skel	.sversionrc	**.thumbnails**
.urlview	.viminfo	**.wine**	.wmrc	.xcoralrc
.xemacs	.xim	.xinitrc	**.xmms**	.xserverrc.secure
.xsession	.xsession-errors	.xtalkrc	.y2log	**.yast2**

এবার প্রশ্ন, এই এতাবৎ আপনার আদেশের ইতিহাস ধরে রাখার গল্পটা থাকছে কোথায়? সেটা একটু আগে অন্য একটা প্রসঙ্গে আজকের আলোচনাতেই বলে এসেছি, আপনি কি মনে করতে পারছেন? সহজে সেই হদিশটা পাওয়ার জন্যে একটা কমান্ড ব্যবহার করা যায়, যার নাম 'set'। এই '`set`' একবার মেরে দেখুন, আবার সেই মিষ্টি করে মাত্র শচারেক লাইন স্ক্রিন বেয়ে গড়িয়ে যাওয়ার গল্প। আমি যেমন 'set' মেরে দেখলাম, মাত্র তিনশো বিরাশি লাইন। আপনার 'set' কত লাইনের? আপনি লাইন গুনতে জানেন তো? না জানলে, এই পাঠমালাটা আর একবার গোড়া থেকে পড়তে পড়তে আসুন, 'wc' বলে কোনো কমান্ড কোথাও পান কিনা। তবে এত বাজে বকেছি কনস্ট্যান্টলি, গোটাটা একটা শব্দকল্পদ্রুমের মত হয়ে গেছে, কী আছে আর কী নেই, অনেকটা আমার লেখাপড়ার টেবিলের মত, যাতে নাকি কু-লোকে বলে, হাতি হারালেও খুঁজে পাওয়া যাবেনা। একদমই বাজে কথা, আজ পর্যন্ত এত বছরে কোনোদিন কোনো হাতি ওখানে হারায়নি। এখানে সেই 'set' মেরে পাওয়া শশ-লাইনের কয়েকটা মাত্র তুলে দেওয়া যাক, দেখুন তো বুঝতে পারেন কিনা। মনে রাখবেন, এটা কিন্তু আমি, ইউজার 'dd', 'set' মেরেছিলাম আমার সুজে অপারেটিং সিস্টেমে। অন্য কোনো ইউজার 'set' মারলে কিন্তু অন্য একটা টেক্সট পাওয়া যেত।

```
BASH=/bin/bash
HISTFILE=/home/dd/.bash_history
HISTSIZE=1000
HOME=/home/dd
PATH=/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:
      /opt/gnome2/bin:/opt/gnome/bin:/opt/kde3/bin:/usr/lib/java/jre/bin:
      /opt/gnome/bin
INFOPATH=/usr/local/info:/usr/share/info:/usr/info
MANPATH=/usr/local/man:/usr/share/man:/usr/X11R6/man:/opt/gnome2/man:
      /opt/gnome/man
LANG=en_US
LOGNAME=dd
OSTYPE=linux
PS1='\u@\h:\w> '
PWD=/home/dd
SHELL=/bin/bash
```

এর দ্বিতীয় লাইন দেখুন, 'HISTFILE=/home/dd/.bash_history', এর মানে, আদেশের ইতিহাসটা রয়ে যায় ওই '/home/dd/.bash_history' ফাইলে। আর এর এক হাজারি মনসবদারির আয়তনটা লেখা আছে এখানে তৃতীয় লাইনে 'HISTSIZE=1000'। এবার একটা জিনিষ খেয়াল করুন, প্রতিটা লাইনের একটা নির্দিষ্ট গড়ন আছে। একদম বাঁদিকে ক্যাপিটাল কেস বা বড় হাতের ল্যাটিন বর্ণমালায় একটা কথা লেখা, তার পর একটা সমান-চিহ্ন (=), তার ডানে একটা কথা লেখা ছোট হাতের বর্ণমালায়। সমানচিহ্নের বাঁদিকে বড়হাতের শব্দগুলোই সিস্টেম ভ্যারিয়েবল। যেমন, এখানে 'BASH', 'HISTSIZE', 'HOME', 'PATH', 'LOGNAME' ইত্যাদি। ভ্যারিয়েবল বলতে বোঝায় চলরাশি, যা চলে বেড়ায়, বদলে যায় এক অবস্থা থেকে আর এক অবস্থায়, এক জন থেকে আর এক জনে, এক সিস্টেম থেকে আর এক সিস্টেমে। এর উল্টো হল কন্সটান্ট বা স্থিররাশি। যা কখনো বদলায় না। এই শব্দগুলো পাওয়া যায় গণিতে, অর্থনীতিতে, পদার্থবিদ্যায়, যে কোনো বিজ্ঞানে। আমাদের বাস্তব পৃথিবীতেও চলরাশি আর স্থিররাশি আছে, যেমন বৃত্তের ক্ষেত্রফল বার করার সময় যে 'π' বা 'পাই'-কে পাই আমরা, সেটা একটা স্থিররাশি বা কন্সটান্ট। কোনো অবস্থায় কোনো জায়গায় এর মান বদলাবে না। ইচ্ছে হলে একদিন বিকেল বিকেল বৃহস্পতিতে বা সকাল সকাল শনিতে গিয়ে একটা বৃত্ত মেপে দেখুন, সেখানেও এর মান সেই একই হবে। কিন্তু অন্য অনেককিছু আবার ভ্যারিয়েবল বা চলরাশি। আমাদের কার কত টাকা আছে এটা মানুষ থেকে মানুষে বদলায়, তাই ভ্যারিয়েবল। একটা বইয়ের চ্যাপ্টারে চ্যাপ্টারে কত করে শব্দ আছে সেটা একটা চলরাশি। আবার কটা আদেশ অব্দি সিস্টেম মনে রেখে দেবে, এটাও একটা ভ্যারিয়েবল। এখানে বলা আছে যে এর সাইজ 'HISTSIZE' হবে একহাজার। শেলের খোঁজার পাথ বা পথনির্দেশ ব্যাপারটা মনে পড়ছে? ইউজার 'dd' কোনো আদেশ দেওয়ামাত্র যেখানে যেখানে শেল খুঁজে দেখবে, সেই পথনির্দেশটা দেখুন দেওয়া আছে, 'PATH' নামের ভ্যারিয়েবল। 'dd'-র শেল কী হবে, সেটাও বলে দেওয়া আছে, 'SHELL' নামের ভ্যারিয়েবল। 'dd'-র শেল ব্যাশকে কোথায় পাওয়া যাবে তা দেওয়া আছে 'BASH' ভ্যারিয়েবলে। 'PS1' নামের ভ্যারিয়েবলটা আসলে আমার কমান্ড প্রম্পট কেমন দেখতে হবে সেটা বলে দিচ্ছে। ডানদিকেরটা একটা ছক বা টেমপ্লেট, যেখানে মানগুলো শেল নিজেই ভরে নেবে, শুধু একটা জিনিষ মনে রাখুন, '\' চিহ্নটা আসলে বলে দেয় শেলকে যে এর পরের কথাটাকে তুমি আক্ষরিকভাবে নাও। পরে বুঝে যাবেন পুরোটা। এখানে তোলা লাইনগুলোর দ্বিতীয় লাইনে লেখা '/home/dd/.bash_history' হল আমার বা 'dd' নামের ইউজারের আদেশের ইতিহাস লিখে রাখার জায়গা। '`less /home/dd/.bash_history`' কমান্ড দিয়ে আমি পাব আমার ব্যবহার করা আদেশের তালিকা। আপনি পাবেন আপনার হোমের '.bash_history' ফাইলে। আর একটা কথা এখানে আপাতত ছোট করে জানিয়ে রাখি, '$' চিহ্নটা বোঝায় কোনো ভ্যারিয়েবলের মান। অর্থাৎ, কোনো ভ্যারিয়েবলের নামের আগে এই চিহ্নটা লাগালেই সিস্টেম সেটাকে বুঝবে ওই ভ্যারিয়েবলের মান। ধরুন, কমান্ড দিলাম, '`echo $PATH`'। এর মানে, ব্যাশকে বললাম, আমার 'PATH' ভ্যারিয়েবলের মানটা স্ক্রিনে ফুটিয়ে তোলো। একটু আগে 'mplayer' চালানোর সূত্রে, রুটের পথনির্দেশ বদলানোর যে কমান্ডটা আমি ইচ্ছে করে ব্যাখ্যা করিনি, '`PATH=$PATH:/usr/local/bin`', সেটার মানে এবার বুঝতে পারছেন? সিস্টেম প্রথমে 'PATH' ভ্যারিয়েবলের চালু মানটা পড়ে নিচ্ছে, তার সঙ্গে লাগাচ্ছে একটা যতিচিহ্ন ':', তার পরে আমাদের দরকারি ঠিকানাটুকু '/usr/local/bin' যোগ করে নিচ্ছে।

এবার প্রশ্ন হল, ভ্যারিয়েবলগুলোর মান যে এটাই এটা শেলকে বলে দিল কে? নিজেই খুঁজে দেখুন না? একটু মনে করিয়ে দিই। এই ফলাফলটা পাওয়া গেছিল 'set' কমান্ড দিয়ে। তার মানে আপনি '`man -k set`' কমান্ড লাগাবেন, 'set' সংক্রান্ত কী কী ম্যানুয়াল সিস্টেমে আছে, সেটা খোঁজার জন্যে। মনে পড়ছে? বা সরাসরি 'set' কমান্ডের ম্যানুয়ালেও যেতে পারেন, '`man set`' দিয়ে। মজার কথা 'set'-এর কোনো ম্যানুয়াল নেই, তাই, এই কমান্ড দিয়ে যা পাবেন সেটা হল 'bash'-এর ম্যানুয়াল পেজ। মানে, '`man bash`' দিয়ে যা পেতেন, কারণ, এই 'set' হল সরাসরি 'bash'-এর কাজ। ম্যান পেজ পেলেন, এবার পড়ে ফেলুন, একাধিকবার। এই মুহূর্তে আপনি খুঁজছেন কনফিগারেশন ফাইলগুলোর নাম। গুড, একটু চালিয়াতি খেলার চেষ্টা করা যাক। আপনি জানেন ম্যান পেজে বা 'less' কমান্ড দিয়ে খোলা পাতায় কী করে একটা বিশেষ শব্দবন্ধ খুঁজতে হয়। '\' টাইপ করুন। দেখুন, পাতার একদম নিচে সেটা ফুটে উঠেছে। এবার, সেখানে টাইপ করে দিন '`file`' শব্দটা। এন্টার মারুন। ম্যান পাতায় প্রথম যেখানে 'file' শব্দটা আছে সেখানে আপনাকে নিয়ে যাবে। ঠিক জিনিষটা না-পেলে আবার একবার '\' মারুন।

এবার আর কিছু টাইপ না-করেই এন্টার মারুন। ও আপনাকে দ্বিতীয় যেখানে 'file' সেখানে নিয়ে যাবে। না-পেলে আবার। আর একটু হেল্প করব? পঞ্চমবার যেখানে 'file' শব্দটা পাবেন, সেখানেই পেয়ে যাবেন প্রয়োজনীয় তথ্যটা। জানতে পারবেন যে, '`set`' মেরে আমি যে তালিকাটা পেয়েছিলাম, যার কয়েক লাইন তুলে দিলাম, সেই তালিকাটা 'bash' পায় চারটে ফাইল থেকে। '/etc/profile', '~/.bash_profile', '~/.bash_login' এবং '~/.profile'। একটু আগে যে ডটানন ফাইল আর ডিরেক্টরির একটা টেবিল আমরা বানিয়েছিলাম, তাতে কি এদের কারোর নাম আপনি পেয়েছিলেন? এই তেরাবেঁকা '~' চিহ্নটা কী তা তো আগেই বললাম। যে যেরকম ঘরেই থাকি-না কেন, দশ বাই দশ, চোদ্দ বাই চোদ্দ, যাই হোক, তার দেওয়ালগুলো যতই সোজাসোজা হোক, এই তেরাবেঁকা ত্রিভঙ্গমুরারি টিল্ড-ই হল আপনার হোম, আমার হোম, সবার হোম। যখনই কেউ '~' টাইপ করে, আপনার গ্নু-লিনাক্স সিস্টেম জানে কোন ইউজার সেটা করল, তখনই বুঝে নেয় যে, এই '~' চিহ্নটা সেই ইউজারের হোম-কে বোঝাচ্ছে। যেমন, 'dd' নামের ইউজারের বেলায় এই '~' মানে '/home/dd/'। একটু আগের তালিকাটায় 'HOME' ভ্যারিয়েবলের মানটা দেখে নিন। 'atithi' ইউজার হলে '~' মানে '/home/ atithi/'। আদেশের ইতিহাসের সূত্রে একটু সিস্টেম কনফিগারেশন বুড়ি-ছোঁয়া হল আমাদের, পরে আবার আসব আমরা এইসব কথায়, এখন ফিরে যাওয়া যাক আমাদের লিংক নিয়ে আলোচনায়।

২.৪।। হার্ড লিংক

এখন আমরা জানি কী করে আগের আদেশ ফেরত আনে সিস্টেম, এবার, সেই ফেরত আনা আদেশ দিয়ে আর একবার আমরা '`ls -l`' কমান্ডটা মারব এই 'onefile', 'twofile', আর 'threefile'-এর উপর। এবার আর আমরা '`ls -il`' মারবনা, কারণ এই তিনটে ফাইলের আইনোড নম্বরের চক্করটা তো আমরা ইতিমধ্যেই জানি। এখন আমাদের 'onefile' কিন্তু বদলে গেছে। একটু আগে আমরা তাতে টেক্সট ভরেছি '`onefile is one file`'। — এই সমস্ত চিহ্নগুলো পরপর, যার মধ্যে ষোলটা অক্ষর আছে, ফাঁকা জায়গা বা স্পেস আছে তিনটে, এবং তারপর একটা অদৃশ্য চিহ্ন, যার নাম নিউলাইন বা নতুন-লাইন। '`ls -l`' কমান্ড দিয়ে দেখা যাক।

```
-rw-r--r--   2  dd  users    20 2003-12-24 10:51 onefile
lrwxrwxrwx   1  dd  users     7 2003-12-24 10:50 twofile -> onefile
-rw-r--r--   2  dd  users    20 2003-12-24 10:51 threefile
```

এবার দেখুন 'onefile' আর 'threefile' দুটোরই বাইটসাইজ দেখাচ্ছে ২০ করে, আর বেচারা 'twofile', তার কোনো বদল হলনা গা, এটা কি একটা বিচার হল? যার সঙ্গে সে লিংকিত, নয় একটু কোমলতার সঙ্গেই, সফট লিংক না, সেই 'onefile' ফনফন করে বেড়ে উঠল ০ থেকে ২০, ফুটপাথ ফাটিয়ে মাথা-গজানো দুর্বোঘাসের মত যারা বারবার প্রমাণ করে, স্টিল স্প্রিং ইজ স্প্রিং ইভন ইন দি টাউন। তার সঙ্গে একই ভাবে বেড়ে গেল ওই আর একটা লিংক 'threefile', নয় একটু হার্ড। আর সে 'twofile' সেই সাত এক্কে সাতের নামতা? দেখুন তো ২০ সাইজটা কি চেনা লাগছে? ষোল আর তিন আর এক যোগ করে কত হয় যেন? এবার দেখুন তো, ওই সাতের রহস্যটার কোনো মানে বুঝছেন? ওই সংখ্যাটাতেই, সফট লিংকের বাইটসাইজেই, লেখা নেই তো গরবিনী নরম-লিংকিনীর প্রাণের কানুর নাম? '`onefile`' নামে কটা অক্ষর? একটু আগে ৯.১ সেকশনে আমি আপনার বানানো সফট-লিংক ফাইলের সাইজ বলে দিচ্ছিলাম মূল ফাইলের নামটার অক্ষর গুনে। প্রত্যেকটা উদাহরণেই গুনে দেখুন। সফট লিংকের বুকে শুধু লেখা থাকে তার নাম যার সে লিংক।

আর 'threefile' কেন বাইটসাইজ কুড়ি সেটা তো আপনারা আগেই জানেন, এটা হল 'onefile'-এর একটা হুবহু প্রতিরূপ। যে কারণে, লিংকসংখ্যার জায়গায়, 'onefile' আর 'threefile' দুজনেরই লেখা আছে দুই। কোনো দিক দিয়েই 'onefile' আর 'threefile'-এর ভিতর কোনোরকম কোনো পার্থক্য করা যায়না, এরা দুজনে তো আসলে একটাই ফাইল, শুধু দুটো আলাদা নাম দিয়ে আমি বোঝাচ্ছি একটাই ফাইলকে। যে কারণে তাদের আইনোড নম্বর এক, সেই আইনোড নম্বর যা দিয়ে সিস্টেম একটা ফাইলকে বোঝে, যাকে আমরা বুঝি তার নাম দিয়ে। তাই আমরা যাকে 'onefile' আর 'threefile' এই দুই নাম দিয়ে বুঝছি সে আসলে একটাই আইনোড নম্বর, সিস্টেমের কাছে একটাই ফাইল। দুই আলাদা নামে তাকে রাখার ফলে ঘটেছে এটাই যে, একটা ফাইলকে উড়িয়ে দিলেও অন্য নামটা

রয়ে যাচ্ছে। তাই রয়ে যাচ্ছে আইনোড নম্বরটাও, তার মানে, রয়ে যাচ্ছে ফাইলটাও। ফাইলটাকে সিস্টেম সত্যি সত্যিই ওড়াতে পারবেনা, যতক্ষণ না তার শেষ হার্ড লিংকটাও উড়িয়ে দিচ্ছি আমরা। মূল ফাইল 'onefile'-কে উড়িয়ে দিয়ে দেখুন। এখন আর 'onefile'-এর সফটলিংক 'twofile' ফাইলকে ক্যাট করে আপনি দেখতে পাবেন না, কানু না থাকলে রাই থাকে কেমন করে? সিস্টেম আপনাকে জানাবে, মূল ফাইল 'onefile' আর নেই। কিন্তু 'onefile' উড়িয়ে দিয়েও '`cat threefile`' করুন, দেখবেন অবিকল মূল ফাইলটা আপনি পেয়ে যাচ্ছেন।

যেসব পরিস্থিতিতে আমাদের লিংক ফাইল ব্যবহার করতে হয় তার কোথাও কোথাও সফট লিংক ব্যবহার করা যায়না, হার্ড লিংক করতেই হয়, তার কারণ, অনেক প্রোগ্রামই আছে যারা সিম্বলিক লিংক বোঝেনা। ধরুন কপি করার আমাদের যে কমান্ড 'cp'। এই 'cp' কোনো সিম্বলিক লিংকের উপর ব্যবহার করলে সে সিম্বলিক লিংকের লিংকত্বটাই বোঝেনা, নিট ফাঁপা নামাবলীটাই কপি করে আনে, মূল ফাইল কপি হয়না। আপনি মূল সফট লিংকটার একটা কপি পান, যে একই ভাবে প্রথম মূল ফাইলটার আর একটা সফট লিংক। এটা হার্ড লিংকের বেলায় হওয়ার কোনো চান্সই নেই। হার্ড লিংক হল জাস্ট আর একটা নামে মূল বাস্তব ফাইলটাই। আর নাম যাই হোক, সে তো বাস্তব একটা ফাইল, আর বাস্তব তথ্য নিয়ে বাস্তব একটা ফাইল তো কপি হবেই। তবে 'cp' কমান্ডটাকেও সিম্বলিক লিংক নিয়ে কাজ করানো যায়, সেই রকম অপশানও আছে। . . . *খেজুরগাছে . . .*

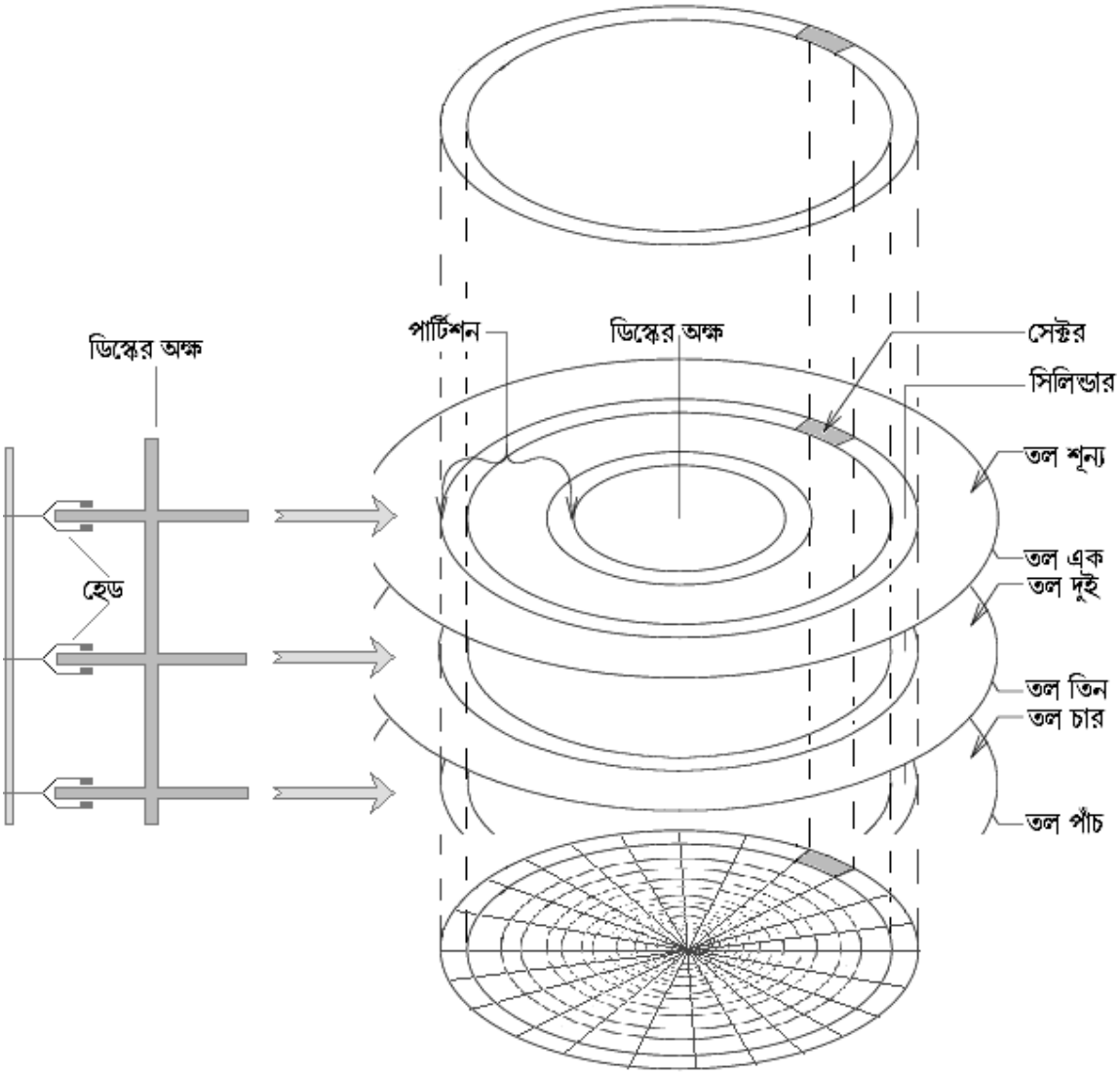
কিন্তু হার্ড লিংকের কঠোরতায় কখনো কখনো ব্যথাও লাগে যখন জানা যায় এই কঠিন যোগাযোগ কখনো দুটো আলাদা ফাইলসিস্টেমের মধ্যে ঘটতে পারেনা। মূল ফাইল আর লিংক দুটোকেই একই ফাইলসিস্টেমে থাকতে হবে। যেমন আমার মেশিনের গ্নু-লিনাক্স সিস্টেমে আমি কোনো হার্ড লিংক বানাতে পারিনা আমার উইনডোজ ফাইলসিস্টেমের পার্টিশনের কোনো ফাইলের। সেটা ফ্যাট-থার্টিটু ফাইলসিস্টেম। সেখানের কোনো হার্ড লিংক বানানো যায়না সুজে সিস্টেমের কোনো পার্টিশনে, সেগুলো রাইজারএফএস ফাইলসিস্টেম। এগুলো কী? দেখেছেন, গরু তার শিঙে যা নিয়েই ঘুরুক, পথ শেষ হবে শ্মশানেই, আমরা ফের ফেরত এসেছি ফাইলসিস্টেমে, যেখান থেকে, আমরা হার্ড ডিস্ক পার্টিশন আর ব্লকের কাহিনীতে যাওয়ার আগে, আগের ছ নম্বর দিনে, ডিভাইস, ডিভাইস ফাইল আর নাম নিয়ে আলোচনা শেষ করার পর, অন্য কিছু কথা বলে নেওয়ার দরকারে আজ শুরু থেকেই চলে গেছিলাম ফাইলের অনুমতি মালিকানায়, তারপর লিংকে। মধ্যে একটু কমান্ডের ইতিহাস তথা ব্যাশ-এর কনফিগারেশন ফাইল নিয়েও কথা বলে নিয়েছি। এবার ফেরত আসা যাক আমাদের মূল আলোচনায়, মানে ফাইলসিস্টেমে। প্রথমে তার প্রথম ধাপ, হার্ডডিস্ক।

৩।। হার্ডডিস্ক, পার্টিশন

শূন্য নম্বর দিনে হার্ডডিস্কের ভিতরে প্ল্যাটার আর হেডগুলোর কাজ করার কায়দাটা মনে আছে? একটা ডিস্ক মানেই কয়েকটা করে প্ল্যাটার। থালার সোজা আর উল্টো আছে, কিন্তু প্ল্যাটারের দুটো দিকই সোজা দিক। তাই প্ল্যাটার পিছু দুটো করে তল, যেখানে চৌম্বক পদার্থের উপর তথ্য লেখা যায় উপযুক্ত প্রকৌশল এবং কায়দা-সম্পন্ন ঘুরতে এবং নড়তে থাকা হেড দিয়ে। মনে পড়ছে? এবারে সেটা একটু ভালো করে বুঝব। সেখান থেকে পার্টিশন, তারপর সেখান থেকে ফাইলসিস্টেম। ছয় নম্বর দিন থেকে যে প্রসঙ্গটা শুরু হয়েছে এবার আমরা তার ফিনিশের দিকে যাচ্ছি, ছয়ের আলোচনাগুলো একটু মাথায় এনে নিন। ফাইল, ফাইলনাম, ফাইলের রকম। ডিরেক্টরি-সিস্টেম বা ডিরেক্টরি-ব্যবস্থা। মানে সিঁড়িভাঙার মত করে স্তরে ডিরেক্টরি এবং ফাইল রাখার বন্দোবস্ত। বহুসময় চালুকথায় একেও আমরা 'ফাইলসিস্টেম' বলি। তবে এটা শব্দটার স্বীকৃত ব্যবহার নয়। 'ফাইলসিস্টেম' শব্দটার স্বীকৃত এক নম্বর ব্যবহার হল গোটা ঐক্যবদ্ধ ফাইল এবং ডিরেক্টরির সমগ্রতা। 'ফাইলসিস্টেম' শব্দটার স্বীকৃত দু-নম্বর ব্যবহার হল ফাইল বানানো পড়া লেখা এবং রাখার নানা রকম। সেই আলোচনায় আমরা এখনো আসিনি, শুধু এদের দুচারটে নামের উদাহরণ দেওয়া ছাড়া, এক্সএফএস, রাইজারএফএস, ইএক্সটিটু, ফ্যাটথার্টিটু, ইত্যাদি। এক নম্বর অর্থে ফাইলসিস্টেম নিয়ে কথা শুরু করেছিলাম আমরা। তার সূত্রে পার্টিশন এবং ডিভাইস, ডিভাইস ফাইল, ফাইলের অনুমতি এবং অধিকার, ফাইল লিংক, তারপর মাইল্ডলি একটু কুপথ, মানে ব্যাশ এবং তার কনফিগারেশন। এবার আবার আমরা ফেরত আসছি ফাইলসিস্টেমের কথায়। পার্টিশন ব্যাপারটা বুঝে নিয়ে আমরা যাব ফাইলসিস্টেমের দু-নম্বর অর্থের আলোচনায়, মানে ডিরেক্টরি এবং ফাইল তথা তথ্য লেখার এবং রাখার কায়দা।

৩.১।। হার্ডডিস্কের ভৌত জ্যামিতি — সেক্টর, ট্র্যাক, সিলিন্ডার, পার্টিশন

এখানে এই ছবিতে আমরা তিনটে প্ল্যাটার দেখিয়েছি, তার মানে তিন দুগুনে ছ-খানা তথ্য লেখার তল, প্রতি তলে তথ্য লেখার জন্যে একখানা করে হেড, তার মানে ছ-খানা হেড। শুধু তলগুলোকে আমরা এক থেকে ছয় না-গুনে, গুনেছি শূন্য থেকে পাঁচ, সেটাই প্রথা। দেখুন দেখানো আছে পরপর। ছটা তলে মোট ছটা হেড একই তালে নড়ে, একই সঙ্গে। একা একা আলাদা করে নাড়ানো যায়না। তার মানে, খেয়াল করুন, যে কোনো একটা নড়াচড়া মানেই ছ-খানা হেডের একই সাথে নড়া, একই ভাবে, একই গতিতে, ছটা আলাদা আলাদা তলের উপর। এদের এই গতির ঐক্য থেকেই গড়ে উঠেছে হার্ডডিস্কের ভৌত ভূমিকে বোঝার আমাদের একক, যার নাম সিলিন্ডার।



থ্রি-অ্যান্ড-হাফ-ইয়ার্ডস-ই তো নাম, সেই গল্পটার কথা মনে করুন, যতটা জমি একসঙ্গে তুমি দৌড়তে পারবে, ততটা জমিই তোমার। লোকটা তাতে এমন দৌড় দৌড়লো যে শেষ অব্দি দৌড়তে দৌড়তে জমিতেই পড়ে গেল, অত অত জমির কিছুই লাগল না, লাগল মাত্র থ্রি-অ্যান্ড-হাফ-ইয়ার্ডস, কত জমি আর লাগবে? যত বড় দৌড়বাজেরই হোক, লাশ তো। হার্ড ডিস্কের বেলায় শুধু এই একা একলা একটা দৌড়বাজকে রিপ্লেস করে দিন ছ-খানা দৌড়বাজ হেড দিয়ে। এরা দৌড়চ্ছে একই তালে, এই ছ-খানা হেডের কমিউন। প্রত্যেকে যতটা করে দৌড়চ্ছে তার পরিমাণ সমান — সেই পরিমাণের ছয়গুণ হল কমিউনের মোট জমি। একটা হেডের একটা দৌড়ের ছ-খানা প্রতিরূপ তৈরি হচ্ছে তিনটে প্ল্যাটারের ছটা তলে ছখানা হেডের ছটা দৌড়ে। আমরা এখানে তিনটে প্ল্যাটার দেখিয়েছি বলে ছজনের

কমিউন, প্ল্যাটার বাড়া মানেই দৌড়বীর বাড়া, সেই একই হারে বেড়ে যাওয়া কমিউনের জমি। দেখুন, এই যে কমিউনের জমির ধারণাটা তৈরি হয়ে গেল, এটাই আমাদের কাজে লাগবে। এই জমিটা ঠিক কীরকম হবে? প্রতিটা প্ল্যাটারের প্রতিটা তলে একই ভৌগোলিক সংস্থানে থাকবে এই জমি। যেমন দেখুন, একটা তলে একটা হেডের এক সেক্টর জমির ঠিক ছ-খানা প্রতিরূপ থাকবে ছখানা তলে। এর ছটা টুকরো জমিকে মিলিয়ে তৈরি হবে কমিউনের জমির একটা সেক্টর। মাইরি, ছবিটা দেখে বোঝার চেষ্টা করুন, প্লিজ, হাউ-টু ডকুমেন্টের একটা ছবির একটু ব্যবহার করে, অনেক পরিশ্রমে, কাল দুহাজারতিন সালের বড়দিনের রাত্তিরে পৌনে বারোটা অব্দি, আজ আবার ভোরে সোয়া চারটে থেকে প্রায় তিনঘন্টা গাবিয়ে ছবিটা পাকানো।

এক একটা তল মানে একগুচ্ছ কনসেন্ট্রিক বা সমকেন্দ্রিক বৃত্ত। ছবির একদম নিচে দেখুন, একটা তলকে আমরা আলাদা করে দেখিয়েছি। এই সমকেন্দ্রিক বৃত্তগুলো হল ট্র্যাক, পরপর ক্রমানুসারী সংখ্যা দিয়ে চিহ্নিত। শূন্য নম্বর তলে ঠিক যেখানে আছে এক নম্বর ট্র্যাক, তার ঠিক খাড়াখাড়ি উপরে বা নিচে রয়েছে অন্য প্রতিটা তলেরই এক নম্বর ট্র্যাক। প্রতিটি নম্বরের ট্র্যাকের বেলাতেই এটা সত্যি। তার মানে, প্রতি নম্বরের ট্র্যাক আছে ছ-খানা, ঠিক খাড়াখাড়ি উপরে-নিচে, তিনটে প্ল্যাটারের ছ-টা তলে। যদি প্ল্যাটারের সংখ্যা বেশি হয়, প্রতি কমিউনে একই নম্বরের ট্র্যাকও বাড়বে। প্ল্যাটার চারটে হলে একই নম্বরের আটটা ট্র্যাক থাকবে এক একটা কমিউনে, পাঁচটা প্ল্যাটার থাকলে দশটা — এই ভাবে সংখ্যাটা বাড়বে। এবার খাড়াখাড়ি প্ল্যাটারগুলোকে পরপর একই সঙ্গে ভাবুন, যেভাবে আমরা ছবিতে ভাঙা দাগ দিয়ে দেখিয়েছি, প্রত্যেকটা প্ল্যাটারের প্রতিটি একই নম্বরের ট্র্যাককে একসঙ্গে নিয়ে এক একটা কমিউনকে আমরা একটা খাড়া সিলিন্ডার বা চোঙ আকারে ভাবতে পারি। এক একটা সিলিন্ডার তৈরি হচ্ছে ঠিক যেকটা প্ল্যাটার আছে তার দ্বিগুণ সেক্টর নিয়ে, আমাদের ছবির বেলায় ছটা। একটা সেক্টর বেয়ে একটা হেড নড়ছে মানে গোটা সিলিন্ডার বেয়ে ছটা হেড নড়ছে। মাথায় রাখবেন, এটা কিন্তু একটা ভৌতিক বা ভারচুয়াল চোঙ। আমরা আমাদের চিন্তায়, বুঝবার সুবিধের জন্যে যাকে বানিয়ে তুলছি। কাজের সুবিধের জন্যে যে ধারণাটাকে তারপর আমরা সফটওয়ারেও ব্যবহার করি। বাস্তব হার্ডডিস্কটা কিন্তু আদৌ এরকম চোঙদার নয়।

এবার, সিলিন্ডারের ধারণাটা যেই এসে গেল, এবার আমরা গোটা হার্ডডিস্কটাকে ভেবে ফেলতে পারি এইরকম কিছু সিলিন্ডারের সমাহার হিশেবে। মোট কটা সিলিন্ডার তাহলে হতে পারে একটা হার্ডডিস্কে? ঠিক যেকটা ট্র্যাক বা সমকেন্দ্রিক বৃত্ত আছে এক একটা তলে, সেই কটা সিলিন্ডারের সমাহার বলে ভাবা যাবে গোটা হার্ডডিস্কটাকে। গোটা হার্ডডিস্কটা কিন্তু ঘুরছে একই সঙ্গে। আমার মেশিনের স্যামসাং হার্ডডিস্কের বেলায় মিনিটে পাঁচ হাজার চারশো বার করে। আরো দামি ম্যাক্সটর বা ব্যারাকুডা হার্ডডিস্কের বেলায় গতিটা আরো বেশি। হার্ডডিস্কটা ঘুরছে মানে ঘুরছে তার মধ্যে ছখানা তল। আর এই ঘুরতে থাকা ছটা তলের উপর দিয়ে নড়ছে ছখানা হেড। একই সঙ্গে ছজন হেড ছটা তলে ছটা আলাদা আলাদা ট্র্যাক বেয়ে দৌড়চ্ছে। এই প্রত্যেকটা ট্র্যাকেরই তার নিজের নিজের তলে একই ক্রমিক নম্বর। প্রতিটা ট্র্যাককে আবার ভাবা হয় ছোট ছোট টুকরোয় যাদের নাম সেক্টর। ছবিতে দেখুন আমরা একটা সেক্টরকে দেখিয়েছি। সচরাচর এক সেক্টর মানে পাঁচশো বারো বাইট তথ্য। একে বলে ব্লক (ব্লক, ফিজিকাল ব্লক, লজিকাল ব্লক এসব নিয়ে কিছু জটিলতা আছে, সেগুলোয় আমরা পরে আসছি)। এবার এই গোটা হিশেবটা ভাবুন, এর মানে, গোটা হার্ডডিস্কের তথ্য রাখার ক্ষমতা দাঁড়ালো — (পাঁচশো বারো) × (প্রতি ট্র্যাকে সেক্টরের সংখ্যা) × (প্রতি তলে ট্র্যাকের সংখ্যা) × (তলের সংখ্যা) = (পাঁচশো বারো) × (প্রতি ট্র্যাকে সেক্টরের সংখ্যা) × (প্রতি তলে ট্র্যাকের সংখ্যা) × (প্ল্যাটারের সংখ্যা × দুই)। যদিও এই হিশেবটা এখানে দিলাম বোঝার স্বার্থে। সচরাচর আমরা ডিস্ক নিয়ে কথা বলি যখন, এই সিলিন্ডার আর সেক্টরের নিরিখেই বলি। তল, ট্র্যাক এইসব রোজকার কথায় আসেনা। আর তথ্য রাখার মোট সামর্থটাকে বুঝি মেগাবাইট গিগাবাইট ইত্যাদি দিয়ে।

এবার দেখা যাক, হার্ডডিস্কের এই ভৌত জ্যামিতি দিয়ে পার্টিশন ব্যাপারটাকে কতদূর বোঝা যায়। ছয় নম্বর দিনে এদের নিয়ে আলোচনা শুরু করেছিলাম, এখন একটানা অনেকটা আলোচনা চলতে থাকবে। পার্টিশন না-বুঝে ফাইলসিস্টেম ব্যাপারটা বোঝাই সম্ভব নয়। এইটুকু আমরা জানি হার্ডডিস্কটাকে কাজের সুবিধের জন্যে ছোট ছোট অংশে ভাগ করে নেওয়া হয়। এই ভাগ করাটা হয় সফটওয়ার দিয়ে, সত্যি সত্যি তো হার্ডডিস্কটাকে হ্যাকস দিয়ে কেটে টুকরো করা হয়না। বড়তর হার্ডডিস্কের এই ছোটতর অংশগুলোকেই আমরা বলি পার্টিশন। এইমাত্র সিলিন্ডার কাকে বলে বুঝলাম আমরা। হার্ডডিস্কের পার্টিশনগুলোকে আবার ভাগ করা হয় এই সিলিন্ডারের সীমানা বরাবর।

প্রথম দিকে, ইনস্টলেশনের সময়, বা অন্য কখনো পার্টিশন বানানোর সময়ে, খুব চটে যেতাম। আমি ঠিক যে সাইজের চাইছি, হুবহু সেই সাইজের না-করে, তার কাছাকাছি একটা সাইজের করছে? কেন রে বাপু? তুই কে? আমি যা বলছি তা তুই মানবি না কেন? কেন ঠিক সেই সাইজের পার্টিশন হবেনা? হয়না এই সিলিন্ডারের কারণে। আমি ঠিক যত মেগাবাইটের পার্টিশন চাইছি, হুবহু সেটা করতে গেলে হয়ত একটা সিলিন্ডারকে ভেঙে ফেলতে হবে। সেটা করার উপায় নেই। এর একটা সমাধান হল, একটা সিলিন্ডারে ঠিক যতটা তথ্য আসতে পারে, হিশেব করে বার করে নিয়ে, এবার তার কোনো সঠিক মাল্টিপল বা গুণিতকে কাঙ্খিত সাইজটা দিলে, একদম খাপে খাপে সেটাই করবে সিস্টেম। অর্থাৎ, প্রতিটি পার্টিশন শুরু হবে হুবহু একটা সিলিন্ডারে, আবার শেষ হবে হুবহু একটা সিলিন্ডারে। পার্টিশনটার ভিতরকার আর বাইরেকার পরিধি এই দুটোই হল দুটো সিলিন্ডার। ছবিটার সঙ্গে মিলিয়ে বোঝার চেষ্টা করুন। একটা নির্দিষ্ট পূর্ণাঙ্গ সিলিন্ডারের বৃত্তে শুরু হবে, গোটাটাকে নিয়ে, আবার শেষ হবে আর একটা নির্দিষ্ট পূর্ণাঙ্গ সিলিন্ডারের বৃত্তে, গোটাটাকে নিয়ে।

কিন্তু এখানে মৃদু ঘাপলা আছে একটা, পার্টিশনের সিলিন্ডারগুলোর মাপজোক রাখার জন্যে যে পার্টিশন টেবিল, তাতে এই কাজের জন্যে জায়গা থাকে মাত্র দশ বিট। দশ বিট জায়গা মানে কতটা তথ্য রাখা যায়, হিশেব করে বলুন তো? শূন্য নম্বর দিনে গাণিতিক তথ্যের মাপজোকের সূত্রে আমরা আলোচনা করেছি, না-পারলে দেখে নিন। প্রতি বিটে রাখা তথ্যের সর্বোচ্চ সীমা হল ২, তার মানে দশ বিটে মোট তথ্য রাখা যাবে, ২ এর দশতম পাওয়ার বা সূচক, মানে ১০২৪। মোট তথ্য রাখার সর্বোচ্চ সীমা ১০২৪। মানে ১০২৪-টা অব্দি আলাদা আলাদা সম্ভাবনাকে হাজির করা যায়। তাই, একটা পার্টিশনেরও সিলিন্ডার সংখ্যার চূড়ান্ত সীমা ওই ১০২৪। এই ঝামেলার জন্যে আগেকার প্রথায় হার্ডডিস্কের সিলিন্ডারের চূড়ান্ত সীমা হত ১০২৪। এই সমস্যার সমাধান হয়েছে লজিকাল ব্লক অ্যাড্রেসিং বা এলবিএ (LBA —**L**ogical/**L**arge/**L**inear-**B**lock-**A**ddressing/**A**rray) এসে। সেখানে এই ভৌত সিলিন্ডারের ধারণাটাকে বদলে নেওয়া হয় একটা যুক্তিনির্মিত বা লজিকাল ছকে। একটা একমাত্রিক ঋজুরেখ ব্লকের সারি বলে ভাবা হয় গোটা হার্ডডিস্কটাকে। ঠিক একটা সরলরেখায় যেমন পরপর বিন্দু আসতে থাকে, পরপর, একের পর এক। গতিটা ঘটতে পারে কেবল একটা মাত্রাতেই, রেখা বরাবর। গোটা হার্ডডিস্কের ভৌত ধারণক্ষমতাটাকে ভেবে নেওয়া হয় পরপর ব্লকের একটা সারি হিশেবে, ডিস্কের ভৌত জ্যামিতিতে আর যাওয়াই হয়না। হার্ডডিস্কের মোট তথ্য রাখার সামর্থটাকে ভাবার এককই হয়ে গেছে সিলিন্ডার। তাই, এলবিএ ছকে সাজিয়ে নেওয়া ব্লকটেবিলও সিলিন্ডার দিয়েই দেখায়। কিন্তু, এই এলবিএ আকারে ছকে ফেলা একটা হার্ডডিস্কের একটা সিলিন্ডারের শুরু বা শেষের সঙ্গে আদত হার্ডডিস্কের ভৌত জ্যামিতিতে একটা ভৌত সিলিন্ডারের বাস্তব শুরু বা শেষের কোনো সম্পর্কই নেই। বায়োস রম চিপে যে আদেশগুলো ভরা থাকে, যার কথা আমরা আলোচনা করেছি এক আর দুই নম্বর দিনে, সেই আদেশগুলোর মধ্যে একটা বিশেষ আদেশস্তর বাড়িয়ে নেওয়া হয়, এলবিএ-র জন্যে। যার নাম 'INT13H', যাতে সরাসরি বায়োস হার্ডডিস্কটাকে আর ভৌত ডিস্ক হিশেবে না-ভেবে, একটা যৌক্তিক ছকে আবদ্ধ ব্লকের সারি হিশেবে নিতে পারে। এতে হার্ডডিস্কের একটা ভৌত পার্টিশনের ১০২৪ সিলিন্ডারের সীমা, এবং স্মৃতি ধরে রাখার ৫২৮ মেগাবাইটের সীমা অতিক্রম করতে পারা যায়।

৩.২।। পার্টিশন — প্রাথমিক ধারণা

আমরা জানি, একটা হার্ডডিস্ককে ফর্মাট করা মানে কী — তাকে কাঁচা ভৌত চৌম্বক পদার্থের একটা সমাহার থেকে ফাইল তথা ডিরেক্টরি-সিস্টেম বানানোর মত জায়গায় নিয়ে আসা। যেকোনো অপারেটিং সিস্টেমের মত গ্নু-লিনাক্স সিস্টেমেও একটা হার্ডডিস্ক ব্যবহার করার আগে তাকে ব্যবহারযোগ্যতায় আনতে হয়, মানে ফর্মাটিং করতে হয়। এই ফর্মাটিং করেই অপারেটিং সিস্টেম কাঁচা হার্ডডিস্কটাকে নিজের তথ্য রাখার এবং পড়ার উপযোগী করে তোলে, কোনো ট্র্যাকে ত্রুটি থাকলে সেটাকে চিহ্নিত করে, যাতে পরবর্তী সময়ে তথ্য লেখার বা পড়ার সময়ে তাদের বাদ দিয়ে রাখা যায়। এর প্রাথমিকতম ফর্মাটিং-টা করা অবস্থাতেই আইডিই বা স্কাসি হার্ডডিস্কগুলো আমাদের কাছে আসে। আমরা ফর্মাটিং বলতে সচরাচর যে ক্রিয়াটা করি তার মানে মূলত ফাইলসিস্টেম বানানো। এর প্রথম স্টেপ হল পার্টিশন বানানো বা গোটা হার্ডডিস্কটাকে ছোট ছোট টুকরোয় ভাগ করা। পার্টিশনটা তৈরি হয়ে যাওয়ার পর থেকে সেটাকে অন্য পার্টিশনগুলো থেকে স্বতন্ত্র এবং স্বাধীনভাবে ব্যবহার করা যাবে, যেন সেটা নিজেই আলাদা

একটা হার্ডডিস্ক। ছ-নম্বর দিনের ৬ নম্বর সেকশনে হার্ডডিস্ক পার্টিশন আর ফাইলসিস্টেমের টেবিলটা আর একবার দেখে নিন। প্রথম মাস্টার হার্ডডিস্ক '/dev/hda'-তে মোট পার্টিশন তিনটে। '/dev/hda1', '/dev/hda5', '/dev/hda6'। দুটো পার্টিশন '/dev/hda1' আর '/dev/hda5', উইনডোজের, সেখানে ফাইলসিস্টেম উইন্ডোজ ফ্যাটথার্টিটু। যখন উইনডোজ ওএস থেকে এই ফ্যাটথার্টিটু পার্টিশনদুটোকে দেখি, তাদের উইন্ডোজ নাম হয় 'c:' আর 'd:'। আর '/dev/hda6' হল প্রথম হার্ডডিস্কের তিন নম্বর পার্টিশন যেখানে ফাইলসিস্টেম রাইজারএফএস। এখানে ওএস আছে দুটো। প্রথম দুটোয় ডিরেক্টরি এবং ফাইলকাঠামো তৈরি করেছে উইন্ডোজ আর তৃতীয়টায় করেছে স্ল্যাকওয়ার। উইন্ডোজ যখন বুট করে, সে তার বুট ফাইল কনফিগারেশন ফাইলগুলো তোলে প্রথম হার্ডডিস্কের প্রথম পার্টিশন '/dev/hda1' থেকে, আর স্ল্যাকওয়ার তোলে '/dev/hda6' থেকে। এই গোটাটার পরপর তিনটে আলাদা স্টেপ।

এক। কাঁচা ডিস্ক থেকে ফাইলসিস্টেম বানানোর অবস্থায় আনা হচ্ছে, পার্টিশন বানিয়ে। এটা গোটা হার্ডডিস্ককে ব্যবহার করেও করা যেতে পারে, আবার এই উদাহরণের মত তার এক একটা অংশকেও করা যেতে পারে। দুই। একটা বিশেষ ওএস তার জন্যে বরাদ্দ করা পার্টিশনে নিজের ফাইলসিস্টেম গড়ে তুলছে, মানে, তার নিজের মত করে ডিরেক্টরি এবং ফাইলগুলো লেখার আর রাখার বন্দোবস্ত। এমনকি একই ওএস আলাদা আলাদা ফাইলসিস্টেম বা তথ্য লেখা/রাখার কায়দা ব্যবহার করতে পারে, যেমন রাইজারএফএস, এক্সএফএস, ইএক্সটিথ্রি, উইন্ডোজ-ফ্যাটথার্টিটু, ডস-ফ্যাটসিক্সটিন, ইত্যাদি। তিন। এক বা একাধিক পার্টিশনকে মিলিয়ে তৈরি হচ্ছে আলাদা আলাদা ওএস-এর আলাদা আলাদা ইউনিফায়েড ফাইলসিস্টেম। এতে কখনো, কোনো একটা ঐক্যবদ্ধ সিস্টেমের ক্রমানুসারী বা হায়েরার্কিকাল ফাইলব্যবস্থায় সবগুলো হার্ডডিস্কের সবগুলো পার্টিশনকেই নিয়ে আসা হতে পারে। যেমনটা করা হয় আমার মেশিনের সুজে ওএস-এ। কিন্তু অন্য কোনোটাই তা নয়। উইন্ডোজ তো গ্নু-লিনাক্স পার্টিশনদের দেখতেই পায়না। আর স্ল্যাকওয়ার ওএস-এ সুজের মূল পার্টিশনটাকে মাউন্ট করা হয়না, সেটায় এক্সএফএস ফাইলসিস্টেম। এক্সএফএস-এর জন্যে একটা ড্রাইভার কারনেলে রাখার দরকার পড়ে। স্ল্যাকওয়ারেও চাইলে করা যায়, আমি ইচ্ছে করেই করিনি। আমার আসুস-এর মাদারবোর্ড, ইন্টিগ্রেটেড ভিডিও কার্ড সাউন্ড কার্ড সহ, 'A7N266VM'। এর এনভিডিয়ার ভিডিও ড্রাইভারে কিঞ্চিৎ বেদনা আছে, ঠিকভাবে কারনেলে কম্পাইল হতে চায়না, অনেক অ্যাডজাস্ট করতে হয়। সুজে নিজেই করে দেয়, কিন্তু স্ল্যাকওয়ারে সবকিছুই নিজে হাতে করে নিতে হয়, প্রতিটি কনফিগারেশন আর অ্যাডজাস্টমেন্ট একদম নিজে হাতে ফাইল লিখে লিখে। তাই, একটা ইউনিফায়েড ফাইলসিস্টেমের সমগ্রে কোন কোন পার্টিশন কী ভাবে ঢুকবে, এটা একদমই নিজের ইচ্ছে আর প্রয়োজন অনুযায়ী। পরে মাউন্ট প্রসঙ্গে ভালো করে এটা বুঝব আমরা।

পার্টিশন কী বারবার তার উদাহরণ দেখছি আমরা, এবার এর সংজ্ঞাটা ভাবা যাক। একটা ভৌত হার্ডডিস্ককে যখন আমরা একাধিক লজিকাল টুকরোয় ভেঙে নিচ্ছি, তার এক একটাকে ডাকছি পার্টিশন বলে। লজিকাল বলতে যৌক্তিক, এই অর্থে যে, ওএস-এর মধ্যে একটা আস্ত অটুট হার্ডডিস্ক ঠিক যে যুক্তি বা লজিকে কাজ করে, একটা পার্টিশনও ঠিক তাই করবে। মানে শারীরিক ভাবে এক হওয়া সত্ত্বেও, কাজের নিরিখে, কাজের যুক্তিতে আর লজিকে, তারা এখন অনেক হার্ডডিস্ক হয়ে উঠল। একটা পার্টিশন তৈরি হয় কন্টিগুয়াস বা প্রতিবেশী ব্লকদের নিয়ে। ব্লক মানে যে একক দিয়ে একটা হার্ডডিস্কের তথ্য রাখার সামর্থ্যকে মাপি। এই ব্লকগুলোকে হতে হবে কন্টিগুয়াস বা একত্রে সম্বদ্ধ, মানে একই জায়গায় পরস্পর সন্নিহিত। একটা হার্ডডিস্কের এখানে কিছু ব্লক, ওখানে কিছু ব্লক, তাদের নিয়ে পার্টিশন তৈরি করা যায়না। একত্র-সম্বদ্ধ ব্লকদের মিলিয়ে তৈরি একটা স্বতন্ত্র পার্টিশনকে ওএস একটা স্বতন্ত্র হার্ডডিস্ক হিশেবে গণ্য করে। হার্ডডিস্কের কোন অংশকে নিয়ে, কোন কোন ব্লক নিয়ে, একটা পার্টিশন তৈরি হচ্ছে, তার খুঁটিনাটিগুলো লেখা থাকে হার্ডডিস্কের পার্টিশন টেবিলে।

একাধিক পার্টিশন যে করি, করতে হয় আমাদের, অনেকসময় তার একটা কারণ হল তথ্যের নিরাপত্তা। উইন্ডোজে থাকাকালীন, ভাইরাসের ভয়ে কুঁকড়ে থাকতে হত সবসময়। কোনো বন্ধু কোনো মেল অ্যাটাচমেন্ট পাঠালেও খোলার আগে দিন দশেক ফেলে রাখতাম। যদি কোনো নতুন ভাইরাস চলে এসে থাকে, তাহলে তার ভাইরাসহন্তা আপডেট বেরিয়ে যাওয়ার সময়টা যাতে দেওয়া যায়। ভাইরাস তো বেশি আসে ওই অ্যাটাচমেন্টগুলো থেকেই। এমএসওয়ার্ড ফাইল অ্যাটাচমেন্ট হলে ম্যাক্রোগুলোয় থাকে প্রায়ই। উইন্ডোজ আমলে তাই একটা বাড়তি পার্টিশন

রাখতেই হত। ভাইরাস আসা মাত্রই একটা পার্টিশনের অত্যাবশ্যক তথ্য অন্যটায় চালান করে দাও, ভাইরাস থাকতে পারে এমন ফাইলগুলো বাদ দিয়ে। তারপর পার্টিশনটা ফরম্যাট করে ফেলো, প্রায়শই 'format/u' মানে আনকন্ডিশনাল ফরম্যাট দিয়ে। ভাইরাসের কোনো ভাই-ও যাতে আর না-থাকে। শুধু ভাইরাস না, অন্য কোনো কেলোও যদি ঘটে একটা পার্টিশনে, হঠাৎ করে রাশিরাশি ব্যাড সেক্টর এসে সব চচ্চড়ি হয়ে গেল, তখন অন্য পার্টিশনটায় রাখা ফাইলগুলো বেঁচে যাবে। তবে ব্যাড সেক্টর আসা মাত্রই পারিবারিক সম্পর্কে মনোযোগ দেওয়াটা অত্যন্ত জরুরি। শিগ্গিরি নতুন ডিস্ক কিনতে হবে অন্য বাজেট ছেঁটে। হার্ডওয়ারের জটিলতায় যাচ্ছিনা, মোদ্দা ব্যাপারটা হল, হার্ডডিস্কের নিজের মধ্যেই কিছু কায়দা ভরা থাকে, বেশ কিছুটা অব্দি খারাপ এলাকা সরিয়ে রাখার। তারপরেও ব্যাড সেক্টর দেখা দিচ্ছে মানে, আসলে কেলোটা ফল্গুর মত গভীর গোপনে অনেকটা চারিয়ে গেছে। যাইহোক, ব্যাড সেক্টর না এলেও, রেগুলার মূল কাজের পার্টিশনটার জরুরি ফাইলগুলোর ব্যাকআপ রাখা উচিত অন্য আর একটা পার্টিশনে। আমার দুটো হার্ডডিস্কেরই জরুরি ফাইলগুলো এটারটা অন্যটায় ব্যাকআপ পার্টিশনে তুলে দেওয়ার জন্যে একটা শেল-স্ক্রিপ্ট বানিয়ে রেখেছি। এক দিন দু দিন অন্তর অন্তর সেটা চালিয়ে দিয়ে চলে যাই। ব্যাশস্ক্রিপ্টের শেষ লাইন 'poweroff', কাজ শেষ হলে ব্যাশ নিজেই মেশিন অফ করে দেয়। পার্টিশন বাড়ানোর আর একটা কারণ হার্ডডিস্কের জায়গা আরো ভালো করে ব্যবহার করতে পারা। এটা খুব দরকার পড়ে তাদের যারা অনেক ছোট ছোট সাইজের ফাইল ব্যবহার করে। বড় বড় নেটওয়ার্ক মেশিনগুলোয় কোটি কোটি কুকি থাকে, বিভিন্ন ওয়েবসাইটের পরিচয়পত্র, সেই ফাইলগুলো খুব গুড়িগুড়ি সাইজের হয়। এবার তাদের জন্যে যদি ব্লক সাইজ স্বাভাবিক চার কিলোবাইটের মাপেই রাখা হয় তাহলে প্রতিটা জীবানু সাইজের ফাইলের জন্যেই নষ্ট হতে থাকে চার কিলোবাইট করে জায়গা, কারণ, একটা ব্লকে একটার বেশি ফাইল রাখেনা সিস্টেম। এই জন্যে ওই ছোট ছোট ফাইল ব্যবহারের বিশেষ পার্টিশন বানানো যায় যাদের ব্লক সাইজ স্বাভাবিকের চেয়ে অনেক কম।

অনেকসময় পার্টিশন বানাতে হয় খুব বেশি কমপ্ল্যান বা হরলিকস খাওয়া ব্যবহারকারীদের শায়েস্তা রাখার জন্যেও। ধরুন পাঁচ গিগাবাইট ডিস্কভূমির একটা পার্টিশনে মাউন্ট করা আছে সিস্টেমের '/home' ডিরেক্টরি, গ্নু-লিনাক্সে তো বলেছি, আলাদা আলাদা ডিরেক্টরি আলাদা আলাদা পার্টিশনে মাউন্ট করা যায়। মানে ডিরেক্টরির জন্যে সর্বমোট বরাদ্দ এখন পাঁচ জিবি, তার বেশি ফাইল তাতে রাখা যাবেনা। আমার মেশিনে এই '/home' ডিরেক্টরির মধ্যে চারটে সাবডিরেক্টরি আছে, '/home/atithi', '/home/dd', '/home/manu', আর '/home/piu'। এবার ধরুন 'piu' নামের ওই ইউজার ভুলভাল সব ফাইল এনে, গানের সিনেমার বা অঙ্কের গাবদা গাবদা ডায়াগ্রাম, ভরেই চলেছে নিজের হোম মানে '/home/piu' ডিরেক্টরিতে। '/home/piu' বেড়েই চলেছে। বাড়তে বাড়তে এমন একটা সময় চলে এল যে অন্য ইউজারদের আর সাধারণ ফাইলগুলো লেখার জায়গাটুকুও রইলনা। এধরনের বেআক্কেলে ব্যবহার সামলানোর একটা সহজ উপায় হল '/home/piu' ডিরেক্টরির জন্যে একটা আলাদা পার্টিশন করে দেওয়া, ধরুন পাঁচ জিবির একের চার মানে সোয়া এক জিবি। নে, এবার কী করবি কর, যা রাখতে পারিস রাখ। এখনো করিনি এটা, আমি খুব উদার বলে।

একখানা আসমুদ্রহিমাচল পার্টিশনের জায়গায় বাংলা বিহার উড়িষ্যা টাইপের ছোট ছোট পার্টিশনে ফাইল অনেক কম টুকরো হয়, যাকে বলে ফ্র্যাগমেন্টেশন। ফ্রাগমেন্টেশন বাড়লে ফাইল পড়ার বা লেখার সময় বেড়ে যায়, আইও ব্যবস্থা শ্লথ হয়ে পড়ে। একটা ফাইল মানেই একগুচ্ছ আলাদা আলাদা ব্লক, আলাদা ভাবে হেড নিয়ে যেতে হচ্ছে তার প্রতিটায়। সমস্যাটা অবশ্য উইন্ডোজ ফ্যাটথার্টিটুতে যে ধরনের, গ্নু-লিনাক্সের ফাইলসিস্টেমগুলোয়, রাইজার বা এক্স বা ইএক্সটিথ্রি, আদৌ তা নয়, পরে দেখবেন। আর, একাধিক পার্টিশনের একটা জরুরত তো আগেই বলেছি, একাধিক ওএস। আমার তিনটে ওএস, সুজে স্ল্যাকওয়ার আর উইন্ডোজ, তাই অন্তত তিনটে পার্টিশন তো লাগবেই।

পার্টিশন নিজের খুশিমত বাড়ানো যায়। শুধু মাথায় রাখতে হয়, পার্টিশন বানানোর সময় একটার জন্যে ঘোষিত ভূমি যেন অন্যটার মধ্যে না ঢুকে যায়। পার্টিশন বানানো বদলানো বা নাড়াচাড়ার জন্যে যেসব সফটওয়ার আছে, 'fdisk', 'cfdisk', 'sfdisk', 'parted' ইত্যাদি, সেগুলোয় এটা হওয়ার কথাও নয়। তবে একাধিক সফটওয়ার দিয়ে আলাদা আলাদা সময়ে পার্টিশনগুলোর আকারআকৃতি বদলালে কখনো কখনো সমস্যাটা হতে দেখেছি। এগুলো সবই ফ্রি সফটওয়ার, 'parted' তো গ্নু-র। 'cfdisk' তো আবার পিছনে 'fdisk'-কেই ব্যবহার করে, এনকার্সেস ব্যবহার করে ছবিতে দেখায় কনসোলেই। আমার দু একবার সমস্যা হয়েছে বাণিজ্যিক সফটওয়ার পার্টিশন-ম্যাজিক আর 'fdisk'-

এর মধ্যে। তবে 'sfdisk' সত্যিই খুব জাঁদরেল, একটু বেশি ক্রিপ্টিক বা সাঙ্কেতিক যদিও, একটু বেশি ভালো করে ম্যানপেজ পড়ে নিতে হয়। আর অনেক ডিস্ট্রিবিউশনের ভিতরে নিজেদের অনেক প্যাকেজ দেওয়া থাকে যা দিয়ে পার্টিশনগুলোর সাইজ বাড়ানো কমানো যায়, কখনো কপিও করা যায়। তবে সেসব বেশি না-করাই ভালো। আপনি এই প্যাকেজগুলোর যেটাই ব্যবহার করুন আগে বার তিনেক গোটা ম্যানপেজ পড়ে নেবেন। নয়তো ঘেঁটে ফেলাটা অবশ্যম্ভাবী, আর পাকস্থলীতে র টাইটানিয়াম লাইনিং না-থাকলে ম্যানপেজ একবারে হজম হয়না। এবং তথ্য হিশেবে বলে রাখি, একমাত্র আইনস্টাইন ছাড়া আর কারুর ভিসেরায়, এখনো, টাইটেনিয়াম পাওয়া যায়নি।

'man' আর 'info'-র কথা আগেই বলেছি। এবার হাউটু পড়ার সময় এসেছে, যে ডিস্ট্রোই ব্যবহার করুন, তার হাউটু-গুলো কোথায় আছে দেখে নিন। খোঁজার একটা সহজ কমান্ড আছে 'whereis'। এছাড়া 'find' তো আছেই। 'find' ব্যবহারের অপশানগুলো আগে শিখে নিন ম্যানপেজ পড়ে। কমান্ড দিন, '`find / -name howto`'। এবার, গোটা '/' ডিরেক্টরির সমস্ত সাবডিরেক্টরি 'find' খুঁজে চলবে, কোথায় কোন ফাইল বা ডিরেক্টরির নামে 'howto' শব্দটা আছে। এখানে আপনি রেগুলার এক্সপ্রেশনও দিতে পারেন। ধরুন '*howto*' ইত্যাদি। এই 'find' কমান্ডটা কাজ করে বাস্তব ডিরেক্টরিতে নেমে নেমে বাস্তব ফাইলগুলোকে খুঁজে খুঁজে, তাই সময় একটু বেশি লাগে। 'find'-এর চেয়ে অনেক দ্রুত কাজ করে 'locate'। 'locate' বাস্তব ফাইল বা ডিরেক্টরিগুলো খোঁজে না, খোঁজে একটা ডেটাবেস। সেই ডেটাবেসটা আগে বানিয়ে নিতে হয় 'updatedb' কমান্ড দিয়ে। এই কমান্ডটা দিতে গেলে 'root' হয়ে নিতে হয়। অন্য ইউজার হয়ে 'login' করা অবস্থায় 'root' হওয়ার জন্যে '`su`' কমান্ড দিন। সিস্টেম রুট পাসওয়ার্ড চাইবে, পাসওয়ার্ড দিয়ে দেওয়ার পর, 'root' হয়ে আপনি '`updatedb`' কমান্ডটা চালাতে পারবেন। ডেটাবেসটা তৈরি হতে বেশ সময় লাগে। যদি অন্য কাজের তাড়া থাকে তাহলে কাজটা করুন ব্যাকগ্রাউন্ডে। '`updatedb &`' কমান্ডে কাজটা পিছনে গোপনে গোপনে হতেই থাকবে। আপনাকে ডিস্টার্ব করবে না। 'Ctrl-F1' থেকে 'Ctrl-F6' টিপে অন্য কোনো ভৌতিক টার্মিনাল বা 'tty'-তে গিয়েও কাজটা করতে পারেন, ভৌতিক বা ভার্চুয়াল টার্মিনালের কথা মনে আছে? কিন্তু, ডেটাবেস আপডেট করার কাজটা বেশ রসদকাঙ্খী, সিপিইউ-র উপর চাপ ফেলে, ব্যাকগ্রাউন্ডে চলার সময়ে অন্য কাজকে শ্লথ করে দিতে পারে। যদি চান, কাজটা ব্যাকগ্রাউন্ডে চলুক এবং আপনার অন্য কাজের সঙ্গে নাইসলি বিহেভ করুক, তাদের কাজকে শ্লথ না-করে, না আটকে, কমান্ড দিন '`nice -19 updatedb &`'। এবার, ডেটাবেস তো তৈরি হয়ে গেল, হাউটু খুঁজবেন কী করে? কমান্ড দিন '`locate howto`'। সাবধান, আবার সেই রাশি রাশি লাইন হুড়হুড় করে নেমে যাচ্ছে পলক ঝপকতে। তার জন্যে উপায় তো বলেছি, 'less' দিয়ে একটু একটু করে পড়া, বা রিডাইরেক্ট করে একটা ফাইল বানানো, সেটাকে পড়া, খুঁজে দেখা। খোঁজার কমান্ড 'grep' মনে আছে তো? এবার এই গোটা প্যারাগ্রাফে নতুন নতুন যে চারটে ব্যাপার এল, এদের 'man' বা 'info' দিয়ে বুঝে নিন। '&' চিহ্নের ব্যাপারটা পাবেন ব্যাশে, বা ব্যাশ-স্ক্রিপ্টিং নিয়ে হাউটুতে, যে হাউটু খোঁজার জন্যে আমরা ডেটাবেস আপডেট করতে বললাম।

কোনো কোনো ডিস্ট্রোতে সমস্ত গ্নু-লিনাক্স হাউটু একসাথে দেওয়া থাকে, যার নাম টোটাল-হাউটু। যেমন সুজে বা ম্যানড্রেক। সুজেতে এর পাথটা হল '/usr/share/doc/howto/en/html/'। ম্যানড্রেকে যদ্দুর মনে হচ্ছে হাউটুর বানানটা ক্যাপিটাল কেসে। কোনো কোনো ডিস্ট্রোয় প্যাকেজ ভিত্তিক হাউটু দেওয়া থাকে, মোটটা থাকেনা, তখন নিজেই নামিয়ে নিতে পারেন, 'www.tldp.org' বা 'www.linuxdocs.org' থেকে। 'tar.bz2' করা ফরম্যাটে সাইজ সবচেয়ে ছোট হয়। আমরা যারা ডায়ালআপ কানেকশনে কাজ করি তাদের পক্ষে সবচেয়ে ভালো। ফাইলনামের এক্সটেনশন 'tar.bz2' মানে কোঁকড়ানো সিন্দুক, এর থেকে ভেঙে মূল ফাইলগুলো বার করার কমান্ড হল 'tar xvjf', মনে পড়ছে? নাকি একটু . . . *খেজুরগাছে* . . .? হাউটুর ভাণ্ডার রোজ বাড়ছে আর বদলাচ্ছে, আমার কাছে শেষ যেটা নামানো আছে তাতে চারশো উনসত্তরটা হাউটু আছে। পিডিএফ ফাইলে বদলে নিলে যার বেশিরভাগেরই আকার 'A4' পঞ্চাশ পাতারও বেশি। বুঝতে পারছেন, সিস্টেম কত দূর অব্দি তার সকল নিয়ে বসে আছে সর্বত্যাগের আশায় — আপনাকে দিয়ে উজাড় হবে বলে? ভেবে নিন, তাকে ভিখারি বানানোর মত আপনি যথেষ্ট গৌরী তো? হাউটুতেই পেয়ে যাবেন, এইমাত্র যে 'tar' আর 'bzip2'-র কথা বললাম, বা যে প্রসঙ্গে আমাদের আসা এই হাউটু-র কথায় — মানে হার্ডডিস্কের পার্টিশন। হার্ডডিস্ক আর পার্টিশন নিয়ে আছে চোদ্দটা। এর পরেও আপনার হার্ডডিস্ক এবং কারনেলের সম্পর্কের ডিটেইলস যদি জানতে চান তাহলে একটা ভারি ভাল উৎস হল সিস্টেমের ভিতরেই

থাকা '/usr/src/linux/Documentation/'। এর মধ্যে দেখুন, আইডিই এবং স্কাসি সম্পর্কে দুটো আলাদা ডকুমেন্ট দেওয়া আছে। আর সামগ্রিক ভাবে ডিভাইস নিয়েও আছে। এগুলোয় আসব, এখন ফেরত যাওয়া যাক পার্টিশনের কথায়।

৩.৩।। পার্টিশনের রকমফের

আগেই বলেছি, একটা পার্টিশনে কেবল একরকমের ফাইলব্যবস্থা থাকতে পারে। ফাইল লেখা/পড়া/রাখার সেই ব্যবস্থাটা, মানে, ফাইলসিস্টেমটা ইএক্সটিথ্রি রাইজারএফএস জেএফএস এক্সএফএস ভিফ্যাট ফ্যাটথার্টিটু যাই হোক, বা সোয়াপ পার্টিশনের নিজস্ব স্বতন্ত্র সোয়াপ ফাইলসিস্টেম হোক। বা গ্নু-লিনাক্স-এর বাইরের যেসব ফাইল ব্যবস্থার পার্টিশনকে মাউন্ট করে নেওয়া যায় কোনো ডিরেক্টরিতে, সেই মাইক্রোসফটের ফ্যাটথার্টিটু বা এনটিএফএস বা সান মাইক্রোসিস্টেমের ইউএফএস হোক, বা আইবিএম-এর জেএফএস, বা এরকম আর কিছু। গ্নু-লিনাক্সে, 'fdisk' জাতীয় পার্টিশন নাড়াচাড়ার প্রোগ্রামে, একটা কোড-তালিকা আছে। প্রত্যেক ধরনের পার্টিশনের নিজের একটা কোড। যেমন ইএক্সটিটু পার্টিশনের কোড '0x83', সোয়াপ পার্টিশনের কোড '0x82', ইত্যাদি। কোন পার্টিশনের কী কোড এটা জানার সহজ উপায় হল 'fdisk'-এ ঢুকে 'l' মারা মানে পার্টিশন কোডের লিস্ট দেখাতে বলা। উইন্ডোজেরও একটা 'fdisk' আছে। তার এসব বালাই-ই নেই, নিজের ফ্যাট ব্যবস্থা, আর সাম্প্রতিক এনটিএফএস আর কিছু তার পড়ার বালাই নেই, তাই কোড-তালিকারও দরকার পড়েনা।

ইন্টেল বা ইন্টেল ধাঁচে নির্মিত মেশিনের বেলায় একটা হার্ডডিস্ককে কটা পার্টিশনে ভাঙা যাবে, এর একটা সীমা গোড়া থেকেই ছিল। মূল পার্টিশন টেবিলটা লেখা হত হার্ডডিস্কের বুট সেক্টরে এবং তাতে জায়গা ছিল মাত্র চারটে পার্টিশনের নাম লেখার। এই ধরনের প্রাথমিক পার্টিশনকে বলে প্রাইমারি পার্টিশন। এদের সংখ্যা হতে পারে সর্বোচ্চ চার। কিন্তু স্বাভাবিকভাবেই এই কটা পার্টিশনে কুলোনো সম্ভব না-হলে তখন পার্টিশন বাড়াব কী করে? তার জন্যে এল লজিকাল বা যৌক্তিক পার্টিশন। যেখানে একটা ওই চারটে প্রাথমিক ভৌত প্রাইমারি পার্টিশনের কোনো একটাকে আবার ভেঙে ফেলা হচ্ছে কতকগুলো ছোটতর পার্টিশনে। তাদের আমরা ডাকছি লজিকাল পার্টিশন বলে। এতে করে ওই চার পার্টিশনের সীমাটাকে ভাঙা গেল। এবার, যে প্রাইমারি পার্টিশনের মধ্যে ছানা পার্টিশনগুলো বানাচ্ছি, তাকে এখন থেকে ডাকব এক্সটেন্ডেড বা পরিবর্দ্ধিত পার্টিশন বলে। এই এক্সটেন্ডেড পার্টিশনের আলাদা করে কোনো ফাইল ব্যবস্থা থাকেনা, কোনো ফাইলও না। এক্সটেন্ডেড পার্টিশনের কাজ নিজের ভিতরের লজিকাল পার্টিশনগুলোকে ধরে রাখা। এই ধরনের পার্টিশনের কোড হল '0x05'। লজিকাল পার্টিশন আর প্রাইমারি পার্টিশনের আর একটা তফাত মাথায় রাখবেন। মূল হার্ডডিস্কে কোনো একটা নির্দিষ্ট পরিমাণ ভূমিকে চাইলে ছেড়েও রাখা যায়, কোনো পার্টিশন না-বানিয়ে, পরে প্রয়োজনমত বানাব বলে। তাই প্রাইমারি পার্টিশনগুলোকে যে কন্টিগুয়াস বা পরপর সংলগ্ন হতেই হবে, মানে, একটা শেষ হওয়া মাত্রই অন্যটাকে শুরু হতে হবে, এমন কোনো বাধ্যতা নেই। এই একটা পার্টিশন, হুই আর একটা, মধ্যে কিছুটা ডিস্কজমি ভগবানের নামে দেবোত্তর হয়ে রইল। কিন্তু একটা এক্সটেন্ডেড পার্টিশনের মধ্যে লজিকাল পার্টিশনগুলোকে ঠিক পরপর বা কন্টিগুয়াস হতেই হবে। মধ্যে কোনো ফাঁক থাকবেনা। মোট কটা লজিকাল পার্টিশন থাকবে তার যৌক্তিক অর্থে কোনো নিষেধ নেই। কিন্তু কেজো অর্থে আছে। প্রত্যেকটা লজিকাল পার্টিশনের মধ্যেই পরবর্তী লজিকাল পার্টিশনের একটা পয়েন্টার বা চিহ্নক থাকে। একটা লজিকাল পার্টিশন তার ঠিক পরবর্তী লজিকাল পার্টিশনের ঠিকানা বলে দেয় সিস্টেমকে। এই করে পরপর অনন্ত সংখ্যক লজিকাল পার্টিশন থাকতে পারে। কিন্তু গ্নু-লিনাক্সের নিজস্ব কিছু নিষেধ আছে এখানে, একটা স্কাসি হার্ডডিস্কে গ্নু-লিনাক্সে মোট পার্টিশন থাকতে পারে পনেরোটা অব্দি, আর আইডিই হার্ডডিস্কে থাকতে পারে তেষট্টিটা অব্দি। তবে, স্বামী পার্টিশনানন্দ না-হতে চাইলে, এতগুলো পার্টিশনের কারুর দরকার পড়ে বলে শুনিনি।

প্রাথমিক ফাইলসিস্টেম পার্টিশনগুলো ছাড়া অন্য আর এক রকম পার্টিশন থাকে একটা গ্নু-লিনাক্স সিস্টেমে। সেটা হল সোয়াপ পার্টিশন। সোয়াপ ফাইল লেখার জন্যেই তুলে রাখা জমিজিরেত, এক কথায় সোয়াপস্পেস বা পরিবর্তভূমি। এক আর দুই নম্বর দিনের আলোচনাগুলো মনে করুন। কম্পিউটারে যখনি আমি কোনো প্রোগ্রাম চালাচ্ছি, একটা প্রসেস চালু হচ্ছে, আর এই প্রসেসের জন্যে কম্পিউটারের সিপিইউ ধার্য করে দিচ্ছে র‍্যাম ব্লকের একটা নম্বর। ঠিক যেভাবে আমরা দেখিয়েছি দুই নম্বর দিনে। যে ব্লক গুলোকে ডাকা হয় পেজ বা স্মৃতির পাতা বলে,

আগেই বলেছি। এই পাতায় যে কোডটা তুলে ফেলা হয়, সেটা খুব আশু ভবিষ্যতেই কাজে লাগবে সিপিইউ-র। সিপিইউ-র ভিতরকার রেজিস্টারগুলো দিয়ে সিপিইউ সেখানে বারবার পড়বে আর লিখবে। এই কোডগুলোকে ডাকা হয় কাজের সেট, 'ওয়ার্কিং সেট' নামে। সত্যিই, ভারি কাজের জিনিষ। এখানে একটা জ্যান্ত ব্যাপার আছে, প্রতি মুহূর্তে যেটা বদলে যেতে থাকে, ইউজারের এবং সিস্টেমের কাজের প্রতিটা মুহূর্তের সঙ্গে মিলিয়ে মিলিয়ে। গ্নু-লিনাক্স ধরে নেয় যে, সদ্য কাজে লাগানো জিনিষগুলো আবার শিগ্গিরি কাজে লাগবে, তাই তাদের র‍্যাম মেমরির কোড-পেজে তুলে ফেলে। কিন্তু যদি একইসঙ্গে তো অনেকগুলো প্রসেস চলছে, এক নম্বর দিনের মাল্টিপ্লেক্সিং-এর কথা মনে করুন। অনেকগুলো প্রসেস একত্রে চলাকালীন কারনেল চেষ্টা করে র‍্যামকে যথাসাধ্য মুক্ত করে তুলতে, যাতে পরপর, একের পর এক কাজকে সে র‍্যামে নিয়ে আসতে পারে। র‍্যামের এই ভার লাঘব করার জন্যেই থাকে সোয়াপ ফাইল। কাজের দিক থেকে বলতে গেলে, র‍্যামকে আসলে বাড়িয়ে তোলে সোয়াপ স্পেস বা পরিবর্ত ভূমি। র‍্যামের পরিবর্তে এই সোয়াপ ভূমিতেই অস্থায়ী ভাবে লেখা হয় র‍্যামের ওই কোডপেজগুলো। যদিও এখানে সবচেয়ে বড় গ্যাঁড়াকলটা হল আইও, ইনপুট/আউটপুট। আইও শ্লথতাটা প্রতিমুহূর্তে র‍্যামপেজগুলো সোয়াপফাইলে বারংবার প্রয়োজনমত লিখে ফেলার এবং পরে কাজের প্রয়োজন অনুযায়ী পড়ে ফেলার ব্যাপারে একটা বাধা হয়ে দাঁড়ায়। র‍্যামে কোনো কিছু লেখা যে অবিশ্বাস্য দ্রুততায় করতে পারে সিপিইউ, সেটা কখনোই হার্ডডিস্কের সোয়াপ ফাইলের বেলায় সম্ভব নয়। তাই সোয়াপ ফাইল র‍্যামকে বাড়িয়ে তোলে বটে, কিন্তু র‍্যামের তুলনায় বড্ড বেশি ধীরে কাজ করে। অনেক প্রসেসের ভিতর অনেক ভাগে ভাগ হতে হতে মেমরি যখন লম্ফর টিমটিমে হলুদ আগুনের মত, তৃতীয় বিশ্বের সাবসিস্টেন্স ইকনমির মত হয়ে পড়ে, তখন কারনেলকে বাধ্য হয়ে অন্য ব্যবস্থা নিতে হয়। প্রসেস থেকে প্রসেসে পার্থক্য আছে, কতটা তাড়াতাড়ি তার কাজের প্রয়োজন তৈরি হচ্ছে, কতটা নিকট অতীতে কার কাজের পয়োজন পড়েছে, তার উপর নির্ভর করে। সবচেয়ে কম সময় আগে যে প্রসেসকে রসদ খাওয়াতে হয়েছে সে হল সবচেয়ে সক্রিয়। তুলনামূলক ভাবে নিষ্ক্রিয় একটা প্রসেসের কাজের জিনিষ মানে ওয়ার্কিং সেটের কোড-পাতাগুলো কারনেল এখন সরিয়ে ফেলে। তাকে সরিয়ে ফেলতেই হয়, আর একটা ওয়েট করতে থাকা প্রসেসের ওয়ার্কিং সেট মেমরি পাতায় তোলার আগে। এই সরিয়ে ফেলা ওয়ার্কিং সেটটাকে কারনেল এখন সরিয়ে নিয়ে রাখে ওই সোয়াপ ফাইলে। একে বলে থ্র্যাশিং, মানে এক অর্থে কেলিয়ে বার করে দেওয়া। কিন্তু আগেই তো বলেছি বারবার, সিপিইউ আর র‍্যামের হটলাইন বেয়ে তথ্য যে স্পিডে লেখা-পড়া হয়, তার ধারেকাছেও আসেনা সোয়াপফাইলের হার্ডডিস্কের আইও। তাই, সোয়াপফাইল র‍্যামকে যতই বাড়িয়ে তুলুক, পর্যাপ্ত র‍্যাম ইজ পর্যাপ্ত র‍্যাম, তার জায়গা কদাচ সোয়াপ ফাইল নিতে পারেনা। গ্নু-লিনাক্সে সচরাচর র‍্যাম আর সোয়াপ ফাইল যোগ করে ভারচুয়াল বা ভৌতিক মেমরির আকার ধরা হয়। তার মানে ধরুন যদি দুশোছাপ্পান্ন মেগাবাইট র‍্যাম হয়, আর পাঁচশো চল্লিশ মেগাবাইট সোয়াপস্পেস রাখা হয়, তাহলে ভৌতিক স্মৃতির আকার দাঁড়াচ্ছে সাতশো ছিয়ানববই এমবি। সচরাচর ধরে নেওয়া হয় যে র‍্যাম যা হবে তার দ্বিগুণ হওয়া উচিত সোয়াপ স্পেস। র‍্যাম খুব বেশি বা খুব কম হলে এই হিশেবটা নাটকীয়ভাবে বদলে যায়। তবে এই হিশেবটারও আভ্যন্তরীন যুক্তি কী আমি জানিনা, তবে চলে কথাটা, এবং আমিও মোটামুটি মেনে চলি। যদিও বহুবার, '`free`' কমান্ড চালিয়ে দেখেছি, অতটা সোয়াপ ফাইল কখনোই প্রায় ব্যবহার হয়না, খুব বড় কোনো কম্পাইলেশনের সময়েও না। 'free' কী? আপনার ম্যান পড়ার free-দম?

এবার প্রশ্ন, ঠিক কী রকমের কটা পার্টিশন থাকবে? আগেই বলেছি, মাস্টার-বুট-রেকর্ড বা এমবিআর লেখা থাকে প্রথম হার্ডডিস্কের প্রথমতম ভূমিটায়। মাস্টার বুট রেকর্ডকে লজিকাল পার্টিশনে লেখা যায়না। তাই যদি আমরা ওই হার্ডডিস্ক থেকে একটাও ওএস বুট করতে চাই, তাহলে অন্তত একটা প্রাইমারি পার্টিশন লাগবেই। ওই হার্ডডিস্ক থেকে বুট করার পর ওএস-কে কাজ করতে গেলে একটা প্রাইমারি পার্টিশন ছাড়াও প্রয়োজন হবে এক বা একাধিক একাধিক সোয়াপ পার্টিশন। তবে সেটা এই হার্ডডিস্কে না-থেকে অন্য হার্ডডিস্কেও থাকতে পারে, যদি আপনার মেশিনে একাধিক হার্ডডিস্ক থাকে। বুট করার প্রাইমারি পার্টিশন, কাজ করার সোয়াপ পার্টিশন, তারপর, এর সঙ্গে, আপনার ইচ্ছে হলে প্রয়োজন হলে লজিকাল পার্টিশন, যা খুশি যেকটা খুশি, এর কোনো শর্ত নেই। আর, বুট করার ব্যাপার না-থাকলে ডিস্কটা গোটাটা একটা না একাধিক পার্টিশন, সেটা প্রাইমারি না লজিকাল তার কোনো শর্ত নেই। তবে সিস্টেমের কাজের স্বার্থে আপনার হার্ডডিস্কগুলো মিলিয়ে এক বা একাধিক সোয়াপ পার্টিশন থাকতে হবে। সোয়াপ পার্টিশন একাধিক, এবং একাধিক হার্ডডিস্কে রাখা, এটা কাজের জন্যে ভালো, যতটুকুই হোক।

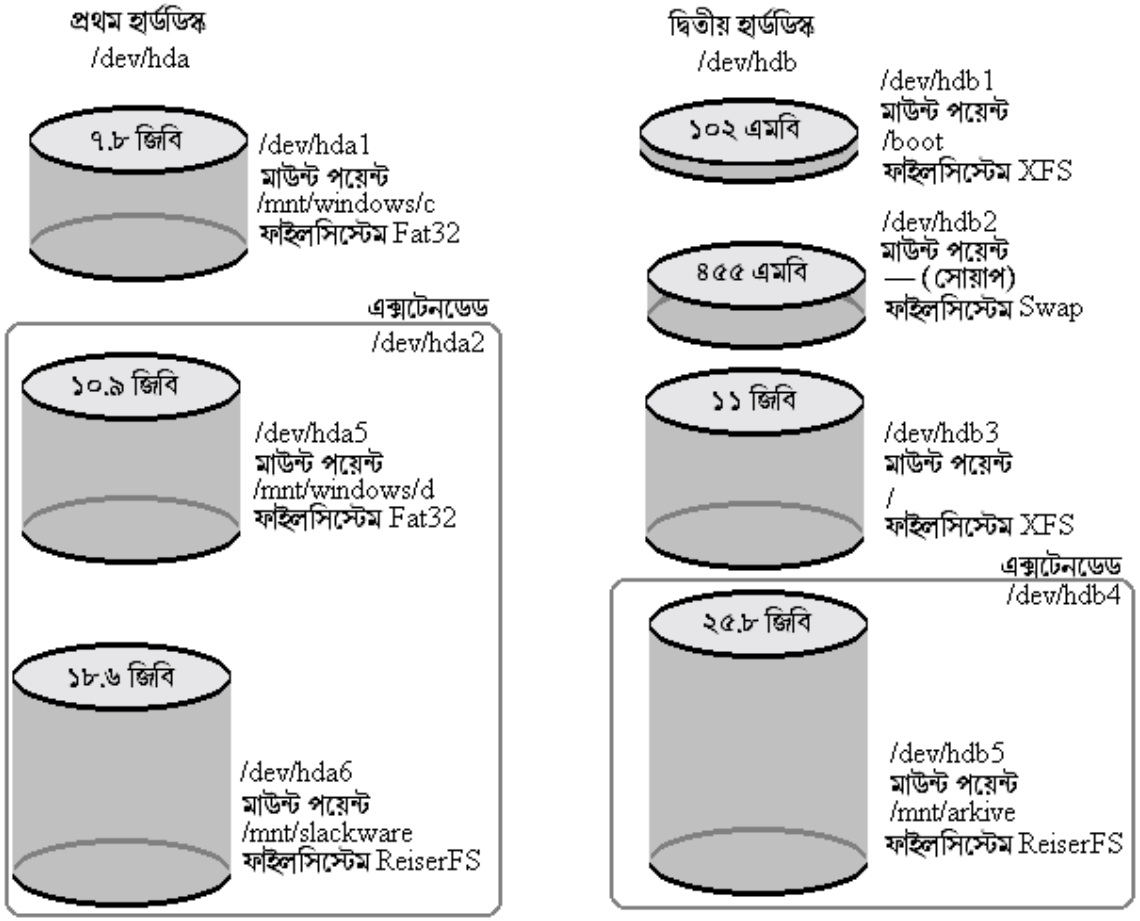
এর পরেও কিছু কথা আছে যেগুলো সেই অর্থে বাধ্যতামূলক না হলেও মেনে চলাটাই প্রথা। '/boot' ডিরেক্টরিটা যে পার্টিশনে মাউন্ট হয় সেটাকে সচরাচর ডাকা হয় বুট পার্টিশন বলে। এর জন্যে যে আলাদা পার্টিশন রাখতেই হবে তা নয়, কিন্তু রাখা ভালো। এই বুট পার্টিশন থেকেই বুট করার সময় তার ওএস তার কারনেল এবং কারনেল সংক্রান্ত ফাইলপত্তর পড়ে। এই বুট পার্টিশনটাও প্রাইমারি পার্টিশন করাটাই প্রথা। তাতে কোনো মেজর ঘাপলা ঘটলে পরিস্থিতিটা সামলানো সহজ হয়। আরো একটা জিনিষ চলে, এই বুট পার্টিশনের ফাইলসিস্টেমটা ইএক্সটিটু বা নিদেন ইএক্সটিথ্রি-তে রাখা। এই ইএক্সটিটু বা পরে তাকে সংস্কার করে ইএক্সটিথ্রি ব্যবস্থাটা গ্নু-লিনাক্সের পুরোনোতর ফাইল লেখা/রাখা/পড়ার ব্যবস্থাগুলোর একটা। পুরোনোতর বলেই পরীক্ষিততর। অনেক বেশি সিস্টেমে অনেক বেশি বার পরীক্ষিত হয়েছে এরা। তাই বেশি নির্ভরযোগ্য বলে অনেকেই মনে করে। এগুলো নিয়ে বিশদ আলোচনা একটু বাদেই আসছে, তখন এগুলো ভালো করে বুঝতে পারবেন। তবে এই নির্ভরযোগ্যতার আলোচনাটা হচ্ছে কিন্তু গ্নু-লিনাক্সের স্তরে। উইন্ডোজের ওই মুহূর্মুহু রিবুট রিস্টার্ট হ্যাং-এর দুনিয়া থেকে সম্পূর্ণ আলাদা অর্থে। ধরুন একটা জলহস্তী আর উটপাখির মধ্যে জল খাওয়া নিয়ে বিতর্কের মত। গ্নু-লিনাক্সে সবচেয়ে চ্যাংড়া ব্যাপারটাও উইন্ডোজের তুলনায় অনেক বেশি নির্ভরযোগ্য, মানে, আসলে তুলনাটাই চলেনা। যাই হোক, বুট পার্টিশনের এইসব নৈয়ায়িকতা আমি নিজেও প্রায়ই মানিনা। নিজেরটায় বহুবার ছাড়াও অন্তত শখানেক জায়গায় তো ইনস্টলেশন করেছি বটেই, অনেক সময় অনেক বিচিত্র কনফিগারেশন করতে হয়েছে, বাধ্য হয়েও করতে হয়েছে। সেসব কনফিগারেশন বললে বিশুদ্ধতাবাদী লিনাক্সীরা বাড়ি এসে আমায় কেলিয়ে যাবে। যারা বহু ইউজারের মেইনফ্রেম সিস্টেম চালায়, নেটওয়ার্কিত বহু মেশিনের সিস্টেম, বহু জটিলতা, তাদের মাইনর কেলো বলে কিছু নেই, কেলো মানেই মেজর, এসব ছ্যাবলামি করার উপায় থাকেনা। আমার আর কী, অন্য হার্ডডিস্কে ব্যাকআপ আছে, ঘেঁটে গেল তো ভি আচ্ছা, গোটা '/home' ডিরেক্টরিটা ব্যাকআপ করা আছে 'home.tar.bz2' নামের সিন্দুক ফাইলে, 'tar' করে 'bzip2' দিয়ে কোঁকড়ানো, জাস্ট আনটার করো, এবং রিইনস্টল করে নাও। বুট পার্টিশন নিয়ে আর একটা চালু একটা প্রথা আছে, সিলিন্ডার সংক্রান্ত, সেটা লিলোর জন্যে। 'lilo' না করে অন্য বুটলোডার ব্যবহার করলে, গ্রাব হোক, বা অন্য কোনো থার্ড-পার্টি প্যাকেজ, তখন এই নিয়মটা মানতে হয়না বোধহয়। 'lilo' দিয়ে বুট করলে বুট পার্টিশনটা প্রথম ১০২৪ সিলিন্ডারের মধ্যে রাখাই প্রথা। প্রায়শই হয়না। তখন একটু বকে দেয় সিস্টেম, কোনো গন্ডগোল ঘটতে দেখিনি। তবে, ভালো না, এসব ভালো না, বলেছেন শাক্যমুনি, চলো ভালো হই, জমকালো হই, বড়দার বাতেলা শুনি।

আমার মেশিনের হার্ডডিস্ক পার্টিশনের খুঁটিনাটি, '`fdisk -l`' করে পাওয়া, এটা ছাঁটকাট করে ছয় নম্বর দিনে ছিল, তার পুরোটা দেখুন এবার। ওখানে এক্সটেনডেড পার্টিশনের লাইনদুটো ছিলনা। ছিলনা ডিস্ক, সিলিন্ডার, ব্লক, ট্র্যাকের ব্যাপারগুলোও। ছিলনা শুরু আর শেষের সিলিন্ডার নম্বর, আর ব্লকের সংখ্যা। সহজবোধ্যতার জন্যে জন্যে জুড়ে দিয়েছিলাম মাউন্টপয়েন্টগুলো। এখানে গোটাটা আছে। নিজের মেশিনে '`fdisk -l`' মেরে মিলিয়ে নিন।

```
Disk /dev/hda: 40.0 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1        1020     8193118+   b  Win95 FAT32
/dev/hda2            1021        4870    30925125    f  Win95 Ext'd (LBA)
/dev/hda5            1021        2448    11470378+   b  Win95 FAT32
/dev/hda6            2449        4870    19454652   83  Linux

Disk /dev/hdb: 40.0 GB, 40060403712 bytes
255 heads, 63 sectors/track, 4870 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/hdb1   *           1          13      104391   83  Linux
/dev/hdb2              14          71      465885   82  Linux swap
/dev/hdb3              72        1508    11542702+  83  Linux
/dev/hdb4            1509        4870    27005265    5  Extended
/dev/hdb5            1509        4870    27005170   83  Linux
```

এই গোটা তালিকাটাকে মিলিয়ে মিলিয়ে পড়ুন, ছয় নম্বর দিনের ছয় নম্বর সেকশনের পার্টিশন তালিকার সঙ্গে। ওই তালিকাটায় '/dev/hda1'-এর পরেই '/dev/had5', আর '/dev/hdb3'-এর পরেই '/dev/hdb5', এটা কেন, সেটা কি বুঝতে পেরেছেন? না-পেরে থাকলে শেষ সেকশনটা আবার পড়ুন। এই প্রশ্নটা অনেক সদ্যলিনাক্সীই করে। আমার হার্ডডিস্কের পার্টিশনগুলোর একটা ছক ছবিতে দিই, দেখুন তাতে বুঝতে সুবিধে হয় কিনা। এখানের তালিকাটা আর ছয় নম্বর দিনের তালিকাটা মিলিয়ে ছবিটা দেখুন। ছবিতে কিন্তু পরিমাপযোগ্যতা বা স্কেল নেই, কী করে থাকবে, আঁকুন তো পাশাপাশি একটা ১০২ এমবি আর একটা ২৫.৮ জিবি পার্টিশন, মানে একটা অন্যটার ২৫৯ গুণ?

দুটো হার্ডডিস্কের পার্টিশন এবং তাদের সাইজ
সুজে সিস্টেমে তাদের মাউন্ট পয়েন্ট — ছ নম্বর দিনের ছ নম্বর সেকশনের তালিকার সঙ্গে মেলান

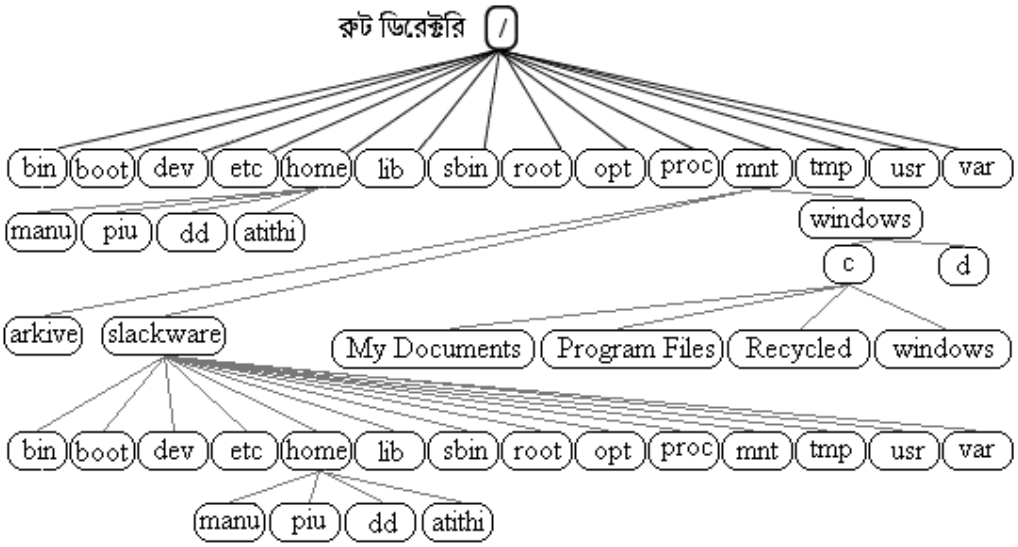


৪।। ফাইলসিস্টেম

পার্টিশন বানানো হল। এবার? সেই পার্টিশনভূমি এখনো শূন্য। এবার সেখানে রাজ্য জেলা মহকুমা পাড়া কায়েম করতে হবে, চোর এবং নেতা বানাতে হবে, মস্তান এবং পুলিশ, বানাতে হবে আদেশ যাওয়ার চোরালাইন হটলাইন ওপনলাইন, কার আদেশ কোথা থেকে কোথায় যাবে, কোন রুটে, সেই গতিপথগুলো। মানে, ফাইলসিস্টেম। পার্টিশন বানিয়ে আর ফাইলসিস্টেম বানিয়ে বারবার যাদের শূন্য করে ভরে দেওয়া হয়, সেই নির্বাক জনআধিক্যের ভার, শ্যাডো অফ দি সাইলেন্ট মেজরিটিজ, সেই আডিস্কমেমরি বাইটকূল, তারা এখনো ফাঁকা, ফাঁপা, ডেড মেন ওয়াকিং, নির্বাক চুপচাপ, আসমুদ্রহিমাচল জনগনমনের মত। বাইটের পর বাইট। মেমরির, হার্ডডিস্কের, সোয়াপ প্রাইমারি লজিকাল পার্টিশনের। সেই বাইটদের নিয়ে প্রত্যেক পার্টিশনেই এখন একটা করে ফাইলসিস্টেম বানানো হবে। ফাইল আর ডিরেক্টরি লেখার/পড়ার/রাখার ছকটা কাঠামোটা। যাতে সব গুলো আলাদা আলাদা পার্টিশনের আলাদা আলাদা ফাইলসিস্টেম পরে মিলে যেতে পারে কেজো চলমান জ্যান্ত একটা ওএস-এর এক

ঐক্যবদ্ধ ফাইলসিস্টেমে। আগেই বলেছি, গ্নু-লিনাক্স ফাইলসিস্টেমের মধ্যে একাধিক ফাইলসিস্টেম ঢুকে আসতে পারে, এসে থাকে, টেকনিকালি বললে, বিভিন্ন পার্টিশনের বিভিন্ন ফাইলসিস্টেমকে মাউন্ট করে নেওয়া হয় এক ঐক্যবদ্ধ ফাইলসিস্টেমে ('ফাইলসিস্টেম' শব্দটার দুটো আলাদা অর্থে দুবার ব্যবহার মাথায় রাখুন)। এই প্রতিটা পার্টিশনে বসবাসকারী প্রতিটা ফাইলসিস্টেমের প্রত্যেকের থাকে নিজস্ব কাঠামো, নিজস্ব শিকড়, নিজস্ব ডালপালা। সবসময়ে এটা আলাদা করে খেয়াল থাকেনা। কারণ, তাদের সবাইকে মিলিয়ে ঐক্যবদ্ধ ফাইলসিস্টেম তৈরি রয়েছে, যার মধ্যে এরা সবাই বিরাজ করে। আমার সিস্টেমে সুজে ওএস-এর '/mnt' ডিরেক্টরিটায় '/mnt/arkive', '/mnt/slackware', '/mnt/windows' এই তিনটে সাবডিরেক্টরি। '/mnt/windows' ডিরেক্টরির ভিতরে আবার দুটো সাবডিরেক্টরি, '/mnt/windows/c' এবং '/mnt/windows/d'। '/mnt/arkive' এবং '/mnt/slackware' ডিরেক্টরিদুটোয় মাউন্ট হয় যথাক্রমে '/dev/hdb5' আর '/dev/hda6' নামের হার্ডডিস্ক পার্টিশনদুটো। আর '/mnt/windows/c' এবং '/mnt/windows/d' ডিরেক্টরি দুটোতে মাউন্ট হয় '/dev/hda1' এবং '/dev/hda5' পার্টিশনদুটো। অর্থাৎ '/mnt' ডিরেক্টরির মধ্যে মাউন্ট হয় মোট চারটে হার্ডডিস্ক পার্টিশন। কিন্তু সুজে ওএস-এর ভিতরে যখন এক ডিরেক্টরি থেকে আর এক ডিরেক্টরিতে যাই, কাজ করি, তখন এটা আলাদা করে খেয়ালই হয়না যে, আসলে আমরা নানা পার্টিশন মিলিয়ে খেলছি। উইন্ডোজ অভিজ্ঞতায় যেমন, স্পষ্ট ভাবে, ডস প্রম্পটে 'c:' বা 'd:' লিখে এন্টার মেরে, বা, উইন্ডোজ হলে, 'c:' বা 'd:' লেখা হার্ডডিস্কের আইকনে ক্লিক করে আলাদা পার্টিশনে যেতে হয়। গোটাটা স্পষ্ট করার জন্যে, ছয় নম্বর দিনের আমাদের পুরোনো পরিচিত ছবিটাকেই একটু বদলানো চেহারায় আর একবার দেখা যাক, এবার শুধু ইউজার 'dd'-র হোম ডিরেক্টরি '/home/dd'-র ভিতরকার ডিটেইলসটা ছেঁটে দিয়ে, তার জায়গায় '/mnt/slackware' এবং '/mnt/windows' ডিরেক্টরির ডিটেইলসটা একটু এনে, জাস্ট এক স্টেপ নিচে অব্দি।

মূল ঐক্যবদ্ধ ফাইলসিস্টেমটা, রুট ডিরেক্টরি '/' থেকে শুরু করে, এখানে সুজে ওএস। দেখুন, '/mnt/slackware' ডিরেক্টরির ভিতরে সাবডিরেক্টরি কাঠামোটা প্রায় মূল '/' ডিরেক্টরির ভিতরকার কাঠামোর সঙ্গে এক। দুটো গ্নু-লিনাক্স ডিস্ট্রোর ফাইলব্যবস্থার আভ্যন্তরীন মিল। কিছু তফাত আছে, সুজে আর স্ল্যাকওয়ারের, কিন্তু জটিলতা এড়াতে সেসব বাদ দিয়েছি। স্ল্যাকওয়ারের '/home' ডিরেক্টরির ভিতর আবার সেই একই চারজন ব্যবহারকারীর হোম ডিরেক্টরি। সুজে সিস্টেমে '/mnt/slackware' ডিরেক্টরিতে মাউন্ট হয় '/dev/hda6' নামের হার্ডডিস্ক পার্টিশন, এই মাত্র বললাম। এই '/dev/hda6' পার্টিশনটাই আবার '/' ডিরেক্টরি হয়ে ওঠে যখন স্ল্যাকওয়ার ওএস-এ বুট করা হয়। গোটা ব্যাপারটা ভালো বুঝতে পারবেন মাউন্ট নিয়ে আলোচনাটা হয়ে গেলেই। এখন একটু দেখে রাখুন। যারা উইন্ডোজ সিস্টেমের সঙ্গে পরিচিত তাদের চেনার জন্যে '/mnt/windows/c' ডিরেক্টরির কয়েকটা সাবডিরেক্টরি দেখালাম ছবিতে।



অর্থাৎ, সুজের বা যে কোনো গ্নু-লিনাক্স ঐক্যবদ্ধ ফাইলসিস্টেমের সংগঠনটার একটা নিজস্ব চরিত্র আছে। সেই চরিত্রটা এবার আমরা বোঝার চেষ্টা করব। শুরু করব ব্লক থেকে। বারবার বলেছি, কম্পিউটারের তথ্যকে বোঝার এবং নাড়াচাড়ার একক হল ব্লক। একটা ব্লক মানে ১০২৪ বাইট। একবার হিশেব করে মিলিয়ে নিন একটু আগের '`fdisk -l`' করে পাওয়া তালিকার সঙ্গে। হিশেব করার সময় খেয়াল রাখবেন, এক্সটেন্ডেড পার্টিশনটা এক বা একাধিক লজিকাল পার্টিশনে ভাঙা হয়ে থাকলে, এক্সটেন্ডেড পার্টিশনের মাপটা তার ভিতরের লজিকাল পার্টিশনগুলোর মধ্যেই রয়েছে। তাই, মোট হিশেবের সময় সবগুলো যোগ করে দিলে, এক্সটেন্ডেড পার্টিশনের মাপটা কিন্তু দুবার চলে আসে। আর ইউনিট বলে যেটা উল্লেখ করা হয়েছে সেটা হল একটা সিলিন্ডারের মোট বাইট ধারণ করার সামর্থ্য। এই সামর্থ্যটা দিয়ে মোট হার্ডডিস্কের বাইটসাইজকে ভাগ করে দেখুন, পাবেন চার হাজার আটশো সত্তর। সেটা এই হার্ডডিস্কের সিলিন্ডারের সংখ্যা। হার্ডডিস্কের প্রতিটি পার্টিশনকেই দেখুন, একদিকে দেওয়া হয়েছে সিলিন্ডারের নিরিখে, কত নম্বর সিলিন্ডার থেকে শুরু, কত নম্বর সিলিন্ডারে শেষ। আর প্রতিটি পার্টিশনের তথ্য ধারণ করার সামর্থ্যটা দেওয়া আছে ব্লকের হিশেবে। এই ব্লকের ভিত্তিতেই তৈরি হয় গ্নু-লিনাক্স ফাইল সিস্টেমের সংগঠন। এই ব্লককে আবার ভাগ করা হয় চারটে আলাদা আলাদা রকমে। আমি মিলিয়ে মিলিয়ে দেখাতে পারতাম, ইচ্ছে করেই আপনার জন্যে ছেড়ে দিলাম। একবার অভ্যেশ হয়ে গেলে, পার্টিশন করার সময়ে সুবিধে হবে। এবার ব্লক গুলোকে আলাদা করে চেনা যাক।

ফাইলসিস্টেমের সংগঠনে বিভিন্ন ধরনের ব্লক

এক, বুট-ব্লকে থাকে একটা ছোট বুট প্রোগ্রাম। এখানেই পার্টিশন টেবিলটা লেখা থাকে। দুই, সুপার-ব্লকে থাকে ফাইলসিস্টেমের বিষয়ে সামগ্রিক তথ্য। এখানে একটা তালিকা রাখা হয় ফাঁকা আইনোডের, যারা এখনো ব্যবহার হয়নি, কারনেল চাইলেই যাদের ব্যবহার করতে পারে। আর রাখা হয় সেইসব ডেটা-ব্লকের তালিকা যারা ফাঁকা আছে, কারনেল চাইলেই যাদের ফাইল লেখার কাজে ব্যবহার করতে পারে। তিন, আইনোড-ব্লকে থাকে ফাইলসিস্টেমের প্রতিটি ফাইলের জন্যে একটা তালিকা, যাতে ফাইলের এবং ডিরেক্টরির প্রতিটি তথ্য লেখা থাকে, শুধু ফাইল বা ডিরেক্টরির নামটা বাদ দিয়ে। চার, ডেটা-ব্লকে থাকে যাবতীয় ফাইল। সে ফাইল যেই বানাক, ওএস বা কোনো প্রোগ্রাম বা কোনো ইউজার। সে ফাইল রেগুলার ডেটা ফাইলই হোক, বা প্রোগ্রাম ফাইল। সমস্ত ধরনের ফাইলেরই আবাস ডেটা-ব্লক। এখানে ছবিতে আমরা আইনোড-ব্লক আর ডেটা-ব্লককে ভেঙে দেখিয়েছি এটা বোঝাতে যে এরা অনির্দিষ্ট দৈর্ঘের হতে পারে, এবং স্বাভাবিকভাবেই আমরা আশা করতে পারি যে বুট-ব্লক বা সুপার-ব্লকের চেয়ে পরিমাণে এরা অনেক বেশি হবে। এবার একটু ভালো করে এদের বোঝা যাক।

৪.১।। বুট ব্লক এবং সুপার-ব্লক

পার্টিশনের ফাইলসিস্টেমের একদম গোড়াতেই থাকে বুট-ব্লক। এর পরেই আসে সুপার-ব্লক। বারবার এই পাঠমালায় যে এমবিআর বা মাস্টার-বুট-রেকর্ডের কথা এনেছি, এই বুট-ব্লকই তার বাসস্থান। বুট-ব্লকে লেখা থাকে একটা পার্টিশন টেবিল, এবং বুটিং প্রোগ্রামের হালহদিশ। এগুলো এখন মোটামুটি আমাদের পরিচিত, বুট ইনিট কারনেলের সূত্রে। পাঁচ নম্বর দিনটা আর একবার ঘুরে আসতে পারেন। সিস্টেম যখন বুট করে, সিস্টেমের বায়োস রম চিপে লেখা নির্দেশ থেকে বায়োস প্রথমেই মেপে নেয় প্রথম হার্ডডিস্কের অবস্থাটা, এবং বুট-ব্লকে লেখা গোটাটা সিস্টেমের মেমরিতে তুলে নেয়। তারপর নিয়ন্ত্রণ ছেড়ে দেয় 'lilo' জাতীয় কোনো বুট প্রোগ্রামের হাতে, আমরা 'lilo' দিয়ে করেছি, অন্য কিছু হতেই পারত। সেই বুট প্রোগ্রাম এবার কারনেল লোড করা শুরু করে, মনে আছে — '/boot' ডিরেক্টরি থেকে '/boot/vmlinuz'? সেই কারনেল, 'cat' করে যার কণ্ঠস্বর শুনেছিলেন। গলা হিশেবে অত্যন্ত থার্ড ক্লাস, কাজে যাই হোক। এই বুটলোডারটা থাকে সেই পার্টিশনের ফাইলসিস্টেমের গোড়ায়, যা দিয়ে বুট করি। বুট করতে হয়না এমন পার্টিশনের ফাইলসিস্টেমে বুট-ব্লকটা থাকে শূন্য। যেমন, আমার সিস্টেমে এমবিআর লেখা হয়েছে প্রথম হার্ডডিস্কের প্রথম পার্টিশন '/dev/hda1'-এর গোড়ায়, অন্য পার্টিশনগুলোর বুট-ব্লক ফাঁকা।

বুট-ব্লকের পরে এবং আইনোড ব্লকের আগে, পার্টিশনের ফাইলসিস্টেমে এবার আসে সুপার-ব্লক। গ্নু-লিনাক্স ওএস-এর পার্টিশন পিছু হিশেব-নিকেশের খাতা হল সুপার-ব্লক। বলতে পারেন। পার্টিশনটার তথ্যধারণের সামর্থ্য কতটা, সামর্থের কতটা ব্যবহৃত হয়েছে, কতটা জায়গা এখনো ব্যবহারযোগ্য আছে, কতগুলো আইনোড-ব্লক এবং ডেটা-ব্লক আছে যাতে এখনো ফাইল বানানো যায় — এই সমস্ত লিখে রাখার খাতা সুপার-ব্লক। এখানে যদি লেখায় ভুল হয় গোটা অপারেটিং সিস্টেমটাই ভুলভাল চলবে, মানে চলবে না। যে যে বিষয়গুলো লিপিবদ্ধ থাকে সুপার-ব্লকে তার মধ্যে মূল এই ছয়টা। এক, এই পার্টিশনে ফাইলসিস্টেমের সাইজ। দুই, ফাইলসিস্টেমে মোট লজিকাল ব্লক কতগুলো আছে (লজিকাল ব্লক বিষয়টায় আসছি এখুনি, আইনোড এবং ডেটা-ব্লকের আলোচনায়)। তিন, শেষ কবে কোনো ফাইল বদলানো হয়েছিল। চার, এই পার্টিশনে মোট কতগুলো ডেটা-ব্লক এখনো ফাঁকা আছে, এবং এখুনি তথ্য লেখা যায় এমন ডেটা-ব্লকের একটা আংশিক তালিকা। পাঁচ, এই পার্টিশনে মোট কতগুলো আইনোড এখনো ফাঁকা আছে, এবং এখুনি ব্যবহার করা যায় এমন আইনোড-ব্লকের একটা আংশিক তালিকা। ছয়, এই পার্টিশনের ফাইলসিস্টেমটা কী অবস্থায় আছে সেই সংক্রান্ত তথ্য, মানে সব ঠিকঠাক আছে, না কিছু জায়গা ঘেঁটে আছে।

পার্টিশনের ফাইলসিস্টেমটাকে ঐক্যবদ্ধ ফাইলসিস্টেমের মধ্যে ঢোকানোর সময় কারনেল সুপার-ব্লকের তথ্যটাকে সিস্টেমের মেমরিতে তুলে নেয়। যখনি কোনো নতুন আইনোড-ব্লক ব্যবহার হয়, বা কোনো ব্যবহৃত আইনোড ফাঁকা হয়, আয়ব্যয়ের হিশেবটা তুলে দেওয়া হয় মেমরিতে তুলে নেওয়া সুপার-ব্লকের কপিতে। ডেটা-ব্লকের বেলাতেও তাই। নতুন কোনো ডেটা-ব্লক যেই লেখা হয়, বা পুরোনো কোনো ডেটা-ব্লক ফাঁকা হয়, সঙ্গে সঙ্গে সেটাকেও কারনেল লিখে ফেলে মেমরিতে তোলা সুপার-ব্লকের কপিতে। কিছু সময় অন্তর অন্তর, কারনেল মেমরিতে তোলা সুপারব্লকের কপিটার সঙ্গে পার্টিশনের সুপার-ব্লককে মিলিয়ে নেয়। মেমরিতে রাখা সুপার-ব্লকের জ্যান্ত কপিতে নতুন যা হিশেব যোগ হয়েছে সেটা লিখে ফেলা হয় পার্টিশনের ফাইলসিস্টেমের সুপার-ব্লকে। এর নাম সিংক (sync) করা *...খেজুরগাছে* ... । ঠিক সিংক করার আগের মুহূর্তটায়, মূল পার্টিশনের সুপার-ব্লকটা কিন্তু মেমরিতে রাখা সুপার-ব্লকের কপির চেয়ে পুরোনো। কিন্তু মেমরির সুপার-ব্লকটা সবসময়ই নতুন। সিস্টেমে শেষবার এই সিংক করাটা ঘটে শাটডাউনের সময়, মেমরির জ্যান্ত সুপারব্লকটাকে পার্টিশনের সুপারব্লকে লিখে কারনেল ছুটি নেয়। বাড়ির লোকও শান্তি পায়, আজকের মত মেশিন অফ হয়েছে। সুপারব্লকে যদি কোনো গোলযোগ থাকে, সিস্টেম তাহলে বুট করতে পারেনা, তাই গ্নু-লিনাক্সে ডিস্কের নানা জায়গায় একাধিক সুপারব্লক লেখা থাকে। যদি একটা সুপার-ব্লক ঘেঁটে যায়, অন্য আর একটা সুপার-ব্লক ব্যবহার করা হয়। গ্নু-লিনাক্সের পূর্বপুরুষ ইউনিক্সে এই বন্দোবস্ত ছিলনা।

৪.২।। আইনোড-ব্লক

সুপার-ব্লকের পরেই আইনোড-ব্লক। ফাইলসিস্টেমের প্রতিটি ফাইলের জন্যে একটা করে আইনোড রাখা থাকে এই আইনোড-ব্লক এলাকায়। আইনোড-ব্লক এলাকাটায় কোনো ইউজারের ঢোকা নিষেধ, শুধু কারনেল ঢুকতে পারে। আইনোড-ব্লক এলাকায় ফাইলসিস্টেমের যাবতীয় ফাইলের আইনোডগুলো থাকে পরপর, পরস্পর-সন্নিহিত বা কন্টিগুয়াস রকমে। প্রতিটি ফাইলের আইনোডে থাকে ওই ফাইল সম্পর্কে এই চরাচরে যা যা জানার থাকতে পারে তার মোটামুটি সবই। শুধু ফাইলের নামটা ছাড়া। প্রতিটি ফাইলের এই আইনোড হল একশো আঠাশ বাইট লম্বা এক একটা তালিকা। যাতে লেখা থাকে — এক, ফাইলের রকম, মানে, সেটা রেগুলার ফাইল না ডিরেক্টরি ফাইল না ডিভাইস ফাইল ইত্যাদি। দুই, কটা লিংক আছে ফাইলটার (‘`ls -l`’-এর দীর্ঘ তালিকার উপাদানগুলো পরপর মিলিয়ে যান)। তিন, ইউজারের ব্যক্তি-পরিচিতি ইউআইডি। চার, ইউজারের গ্রুপ-পরিচিতি জিআইডি। পাঁচ, নানা রকম অনুমতির ওই তিন তিরিখ্খে নটা ড্যাশ। ছয়, ফাইলের মোট বাইট-সংখ্যা। সাত, ফাইল শেষ কত তারিখ কোন সময়ে বদলানো হয়েছিল। আট, শেষ কত তারিখ কোন সময়ে ফাইলটা খোলা হয়েছিল। নয়, শেষ কত তারিখ কোন সময়ে আইনোডটা বদলানো হয়েছিল। দশ, ফাইলের পনেরোটা পয়েন্টারের একটা তালিকা। এই পয়েন্টারগুলো ফাইলের কতকগুলো চিহ্নক, ফাইলটায় যারা পৌঁছে দেয়, যার আলোচনা আমাদের পাঠমালার এক্তিয়ারের বাইরে।

সিস্টেম আইনোডকে বোঝে আইনোড নম্বর দিয়ে। আইনোড নম্বরটা একটা ফাইলসিস্টেমে প্রতিটি ফাইলের নিজস্ব এবং স্বতন্ত্র। এই নম্বরটা আসলে আইনোড-ব্লকের সমগ্রে ওই বিশেষ আইনোডটার অবস্থানের সূচক বা ইন্ডেক্স। আইনোড কথাটাই এসেছে ইন্ডেক্স নোড থেকে। এই নম্বরটা থেকে নিছক যোগ বিয়োগ করে একটা আইনোডকে

পেয়ে যায় সিস্টেম। আইনোড নম্বর কী ভাবে পেতে হয়, মনে আছে, '`ls`'-এর সঙ্গে '`-i`' অপশান হিশেবে দিয়ে। আপনি সিস্টেমে সবকিছু নেড়ে চেড়ে দেখতে থাকুন, আর বোঝার জন্যে ডকুমেন্টেশন পড়ুন। অনেকটা বুঝতে পারবেন না, এই একটা ফাঁক, ওই একটা অস্পষ্টতা, কিছুটা আন্দাজ, এইরকম। একসময় দেখবেন, হঠাৎ কবে গ্নু-লিনাক্স সিস্টেম তার অভ্যন্তর নিয়ে আপনার সামনে হাজির হতে শুরু করেছে। গোটা পাঠমালায় যে চেষ্টাটা করছি। গ্নু-লিনাক্স সিস্টেম বহু মানুষের মিলিত ভাবে বানানো। বানানোর পিছনে তাদের একটা জ্যান্ত চিন্তাপ্রক্রিয়া ছিল। সিস্টেম বোঝা মানে এই চিন্তাপ্রক্রিয়াটাকে বোঝা। আমি যেভাবে প্রক্রিয়াটাকে বোঝার চেষ্টা করেছি তার কিছু প্রম্পট আপনাদের ধরে দিতে চাইছি। যাতে পিছনের জ্যান্ত প্রক্রিয়াটাকে চিনতে শুরু করতে পারেন। এখানে একটা জিনিষ খেয়াল করুন, আইনোডে কিন্তু ফাইলের নামও থাকেনা আবার আইনোডের নম্বরও থাকেনা। এই দুটোই রাখা থাকে ডিরেক্টরি ফাইলে। মনে আছে? যে ফাইল কেউই দেখতে পায়না, এমনকি সর্বত্রগামী সর্বশক্তিমান রুটও না, শুধু কারনেল ছাড়া। রুটও অব্দি শুধু ডিরেক্টরির মধ্যে কী কী সাবডিরেক্টরি আর ফাইল আছে সেটুকুই কেবল দেখতে পায়। অন্যদের সঙ্গে রুটের অধিকারের পার্থক্য এটুকুই যে অন্য কেউ যে কোনো ডিরেক্টরিতে যে কোনো ফাইলে 'ls' মেরে তথ্যটা দেখতে পায়না। আর রুট 'ls' মারতে পারে যে কোনো ডিরেক্টরিতে যে কোনো ফাইলে। ফাইল পিছু ওই পনেরোটা পয়েন্টার ছাড়া আর একটা ফাইল সম্পর্কে আর সবকিছুই জানতে পারা যায় 'ls' কমান্ড দিয়ে। নানা ধরনের তথ্য দেখার জন্যে 'ls' কমান্ডের সঙ্গে নানা আলাদা আলাদা অপশান দিতে হয়। ম্যান পড়ুন, খুঁজে দেখুন।

'ls' কমান্ডটা যখন চালাচ্ছি, কারনেল আসলে ওই ডিরেক্টরি ফাইল পড়েই আমাদের সামনে খুঁটিনাটিগুলো ফুটিয়ে তুলছে। কারনেলের বা অপারেটিং সিস্টেমের তো ডিরেক্টরি ফাইল পড়ার অধিকার আছে, আগেই বলেছি। যখনই আমরা একটা ফাইল খুলি, সুপার-ব্লকে যেমন ঘটছিল, মেমরিতে তোলা জ্যান্ত কপিতে সুপার-ব্লকের বদল লিখে নিচ্ছিল সিস্টেম, এখানেও ঠিক তাই হয়, সিস্টেম হার্ডডিস্ক থেকে ফাইলটার আইনোডটাকে তুলে নেয় মেমরিতে তুলে নেওয়া আইনোড টেবিলের কপিতে। বারবার এই যে নানা কিছুকে মেমরিতে তুলে নেওয়ার কথা বলছি, এটাই হল গ্নু-লিনাক্স কারনেলের কাজ করার পদ্ধতি। সবকিছুই সে প্রথমে মেমরিতে তুলে নেয়, তারপর সেখান থেকে কাজ করে, তাই কাজ করতে পারে অনেক দ্রুত। আমরা যারা উইন্ডোজে কাজ করে অভ্যস্ত হয়েছি গোড়ার দিকটায়, তাদের যেমন বলেছি না, প্রথম প্রথম একটু অবাক হয়ে আর একবার মিলিয়ে নিতে হয়, কাজটা সত্যি হয়েছে তো। যেসব কাজ দুটো সিস্টেমেই খুব প্রাত্যহিক ভাবেই করতে হয়, বড় বড় ফাইল কপি করা, এনকোড করা, কম্পাইল করা ইত্যাদি, যেখানে সময়টার তুলনা করা যায়, তফাতটা সত্যিই এতটাই বেশি। এই আইনোডেও তেমনি। কারনেল গোটা সময়টাই কাজ করে মেমরিতে রাখা ওই আইনোডের কপি বা প্রতিলিপি নিয়ে। এই প্রতিলিপিটাকে ডাকা হয় 'আইনোড বাফার' বলে। বাফার নিয়ে আগেও কথা হয়েছে, মনে করুন। এই আইনোড কপিটাও এই মুহূর্তে একটা বাফার। যাকে নির্দিষ্ট সময় অন্তর মিলিয়ে নেওয়া হয় ডিস্কে থাকা মূল আইনোডের সঙ্গে এবং নতুন গজিয়ে ওঠা পার্থক্যটুকু লিখে ফেলা হয় মূল আইনোডে। আবার সেই 'সিংক'। এবার ভাবুন, হঠাৎ করে যদি কারেন্ট অফ করে দেওয়া হয়, সিস্টেমের কাছে গোটাটাই তো গোলমেলে হয়ে যাবে। মেমরিতে তোলা আইনোডের জ্যান্ত কপি আর ডিস্কে রাখা মূল আইনোডের স্থির সংস্করণ, এদের মধ্যে একটা ফারাক এসে যাবে, কারণ, শাটডাউনের সময় সেই চূড়ান্ত লিখে ফেলে বেরিয়ে আসাটা আর ঘটেনি। এই রকম সব কেরোসিন পরিস্থিতিতে ফাইলসিস্টেম চেক করার একটা ব্যবস্থা থাকে, সিস্টেম নিজেই সেটা করে নেয়, আলাদা করেও করা যায়। কমান্ডটা হল ফাইলসিস্টেম-চেক 'fsck'। প্রত্যেক আলাদা আলাদা ধরনের ফাইলসিস্টেমের জন্যে এর আলাদা আলাদা প্যাকেজ কাজ করে, 'fsck.ext2', 'fsck.reiserfs', 'fsck.vfat' ইত্যাদি। ...*খেজুরগাছে*..। শুধু একটা জিনিষ ভয়ানকভাবে মনে করিয়ে দিই — একটা পার্টিশনে মাউন্ট করা অবস্থায় কদাচ তাতে 'fsck' চালাতে নেই, কদাচ নয়, গোটা হার্ডডিস্কটাই ভোগে চলে যেতে পারে। এখনো আপনি মাউন্ট আর মাউন্টের কনফিগারেশন ফাইল '/etc/fstab' জানেন না, তাই ম্যান বা ইনফো বা হাউটু পড়ে জানুন, কিন্তু সিস্টেম ঠিকমত বুঝে ওঠার আগে 'fsck' প্রয়োগ করতে যাবেন না, কিছুতেই না।

৪.৩।। ডেটা-ব্লক

পার্টিশন সিলিন্ডার ট্র্যাক সেক্টর এলবিএ এইসবের আলোচনার সূত্রে 'লজিকাল-ব্লক' কথাটা উল্লেখ করেছি, তখন তার সংজ্ঞাটা আমরা জানতাম না, জানা সম্ভবও ছিলনা। এবার সেটা বুঝব। লজিকাল-ব্লক বা যৌক্তিক ব্লক বলে একটা

ব্লককে আলাদা করা হয় ব্লকের ভৌত ধারণা বা ফিজিকাল-ব্লক থেকে। সচরাচর কোনো সিস্টেমে হার্ডডিস্ক কন্ট্রোলার ফাইল পড়া বা লেখার সময় তথ্য নাড়াচাড়া করে পাঁচশোবারো বাইটের এক একটা এককে। আইও ডিভাইস কন্ট্রোলার ব্যাপারটা, যদি ভুলে গিয়ে থাকেন, মনে পড়িয়ে নিন দুই নম্বর দিন থেকে। একটা সংস্থা যার হাতে কারনেল দায়িত্ব দেয় আইও ডিভাইসগুলোয় তথ্য লেখা/পড়ার। কারনেল এই আইও ডিভাইস কন্ট্রোলারকে নিয়ন্ত্রণ করে ডিভাইসের ড্রাইভার দিয়ে। এবার, যা বলছিলাম, ডিভাইস কন্ট্রোলারে তথ্য নাড়ানোর একক এই পাঁচশো বারো বাইটের প্যাকেটকে সচরাচর ডাকা হয় ফিজিকাল ব্লক বলে। কিন্তু কারনেল তার তথ্য লেখা বা পড়ার কাজে এই একক ব্যবহার করেনা। যে একক ব্যবহার করে সেটা এই ফিজিকাল ব্লকের কোনো একটা পূর্ণ সংখ্যার গুণিতক। দুইগুণ বা চারগুণ ইত্যাদি। কারণ, আসলে তো কাজটা হচ্ছে একদম নিম্নতম স্তরে এই ফিজিকাল ব্লক দিয়েই। ব্যাপারটা এরকম দাঁড়াচ্ছে যে কারনেলের কাজে কাজের একক মানে কন্ট্রোলারের দুবার বা চারবার বা আটবার কাজ করা। কারনেলের তথ্য বোঝার এই এককেকেই ডাকা হয় লজিকাল বা যৌক্তিক ব্লক বলে। সিস্টেমের কাজের গতি অনেকটাই এই লজিকাল-ব্লকের আয়তনের উপর নির্ভরশীল, কতটা দ্রুত বা শ্লথ ভাবে কাজটা সামলাবে সিস্টেম।

ব্লক-সাইজ বাড়ানো বা কমানোর অন্য একটা সমস্যার কথা উল্লেখ করে এসেছিলাম, সেটায় আসা যাক এবার। আগেই বলেছি, আমাদের ক্যারেকটার ডিভাইসগুলো, যেমন কনসোল বা প্রিন্টার, এরা তথ্য নাড়াচাড়া করে চিহ্ন বাই চিহ্ন, মানে ক্যারেকটারের এককে। কিন্তু সিস্টেম হার্ডডিস্কের মত ব্লক ডিভাইসগুলোর সঙ্গে তথ্য নাড়াচাড়া করে খাবলায়। এক খাবলা, আর এক খাবলা, এই ভাবে। এই খাবলাগুলোই ব্লক। এই লজিকাল ব্লক-সাইজ সিস্টেম থেকে সিস্টেমে বদলায়। ১০২৪ বাইট থেকে ৮১৯২ বাইট বা হয়তো তারও বেশি। আমরা আগেই বলেছি, গ্নু-লিনাক্সে লজিকাল-ব্লক হল ১০২৪ বাইট। একটু আগে দেওয়া '`fdisk -l`' কমান্ডের দেওয়া তালিকার সঙ্গে এর ঠিক পরেই দেওয়া ছবিটার পার্টিশন সাইজগুলো মেলান। প্রত্যেকটার ব্লক সংখ্যাকে ১০২৪ দিয়ে গুণ করুন, এতে আপনি মোট বাইট সাইজ পেলেন পার্টিশনটার, একে ১০২৪ দিয়ে ভাগ করলে কেবি বা কিলোবাইট, তাকে ১০২৪ দিয়ে ভাগ করলে এমবি বা মেগাবাইট, তাকে ১০২৪ দিয়ে ভাগ করলে জিবি বা গিগাবাইট। প্রত্যেকটা পার্টিশনেই মেলান, মিলছে? এবার ভাবুন, একবার একটা লেখা বা পড়ার কাজে একটা ব্লক ব্যবহার করছেন মানে, আপনি হয়ত পাঁচ বাইট মাত্র তথ্য লিখলেন, তার জন্যে খরচ হয়ে গেল একটা গোটা ব্লক মানে ১০২৪ বাইট। এই দুর্মূল্যের বাজারে এক হাজার উনিশ পিস বাইট সিম্পলি নেই হয়ে গেল। তবে আপাতত শুনতে যতটা নাটকীয় লাগছে, তা কিন্তু নয়। ধরুন, এই পাঠমালার ছয় নম্বর দিনের পিডিএফ ফাইলটা, সেটার সাইজ ৩৬০৮৮১ বাইট। ১০২৪ বাইট মানে ১ কেবি, সাইজটাকে ১০২৪ দিয়ে ভাগ করে দেখুন, উত্তর আসছে ৩৫২.৪২ কেবি। তার মানে ৩৫২ খানা লজিকাল ব্লক পুরো ব্যবহৃত হল, এর পরে পড়ে থাকা বাইট গুলোয় পরবর্তী ব্লকটার ৪২ শতাংশ ব্যবহৃত হয়েছে, তার মানে নষ্ট হয়েছে ৫৮ শতাংশ, মানে ৫৯২ বাইট। ৩৫২.৪২ কেবির জায়গায় লাগছে ৩৫৩ কেবি। অর্থাৎ, নষ্ট হচ্ছে ফাইলের মোট সাইজের ২ শতাংশেরও কম। এবং ফাইল যত বড় হবে, এই অনুপাতটা আরো কমবে।

এই ডেটা-ব্লকরা শুরু হয় আইনোড-ব্লকের পরে। ছবিটা দেখুন। প্রথমে বুট, তারপর সুপার, তারপর আইনোড, তারপর ডেটা। একদম এক নম্বর দিনের সেই আলোচনা মনে করুন, হার্ডডিস্কের প্রতিটা ফাইলের প্রতিটা তথ্যের একটা নিজের ঠিকানা আছে — সেই ঠিকানাটা হল মোট ডেটা-ব্লকের সমগ্রে একটা বিশেষ ডেটা-ব্লকের নিজস্ব অবস্থান সূচক নম্বর থেকে। মানে অমুক পার্টিশনের অমুক নম্বর ডেটা-ব্লক এই ঠিকানা দিয়ে আমরা ব্লকটাকে চিনি। যে ব্লকগুলোয় তথ্য আছে, তথ্য লেখা হয়ে গেছে, তাদের ডাকা হয় ডিরেক্ট ব্লক বলে। মনে হচ্ছেনা, হঠাৎ এর নাম 'ডিরেক্ট' কেন? এর মানে কী? নামটা যারা দিয়েছিল, তারা তাদের চালু ভাষার একটা শব্দ, 'ডিরেক্ট', যখন ব্যবহার করছে, তখন, 'ডিরেক্ট' শব্দটার অভ্যস্ত অর্থের কোনো না কোনো একটা ছায়াপাত নিশ্চয়ই থাকবে। এগুলো ডিরেক্ট ব্লক, যে বৈশিষ্ট্য এদের ডিরেক্ট করেছে, তার অনুপস্থিতি নিশ্চয়ই অন্য কিছু ব্লককে না-ডিরেক্ট করেছে। কোন ব্লকগুলোকে? একটু আগে, ঠিক এটার কথাই বলছিলাম। সিস্টেম যারা বানিয়েছে তাদের চিন্তার গতিটা এখন বোঝার চেষ্টা করছেন। তাদের ক্রিয়ার পিছনে সক্রিয় চিন্তাপ্রক্রিয়াটাকে। আর এক অর্থে, আপনি আসলে সেই লোকগুলোকে চেনার চেষ্টা করছেন। কাউকে চেনা মানে তো তার চিন্তাপ্রক্রিয়া চেনা। আর কোনো চেনা হয়না।

মোট ডেটা-ব্লকের সমগ্রে একটা বিশেষ ডেটা-ব্লকের নিজস্ব অবস্থান সূচক নম্বর দিয়ে ব্লকদের সূচিত করার ঠিকানা তো হল, কিন্তু, হারগিস এমন একটা ফাইল পাওয়া যায়না, যাতে সমস্ত ব্লক পরপর-সন্নিহিত। আরো সেই ফাইল

যদি ব্যবহার্য ফাইল হয়। যখনই ব্যবহার করছেন, একটা ফাইল বাড়ছে, ফাইলটার ঠিক শেষ ব্লকটার পরের ব্লকটা ফাঁকা নেই, তখন কারনেল কী করবে? তার কোনো দোষ নেই, সে গোটা ডিস্ক জুড়ে ছড়ানো ব্লকদের থেকে যেই কোনো ফাঁকা ব্লক পায় সেখানেই লিখতে থাকে। কিন্তু এই হুড়ুম-দাড়ুম ছিটিয়ে থাকা ব্লকে বিশ্বময় ছড়িয়ে যাওয়া আপনার ফাইলকে পড়া বা লেখাটা ক্রমশই কারনেলের কাছে আরো আরো সময়সাধ্য হয়ে ওঠে, রিড-রাইট মানে লেখা-পড়ার গতি কমে যায়। এরই নাম ডিস্ক ফ্র্যাগমেন্টেশন। আবার দেখুন, এই ফ্র্যাগমেন্টেশন আছে বলেই ইচ্ছেমত ফাইল বাড়ানো বা কমানো যায়। যদি বাধ্যতামূলক ভাবে একটা ফাইলকে শুধু তার সন্নিহিত ব্লকেই বাড়াতে হত, তাহলে কী ঘটত সেটা ভাবুন। আপনি হয়ত ৫০০ কেবি জায়গা ছেড়ে রেখেছিলেন আপনার ফাইলের জন্যে। এবার কাজের সঙ্গে বাড়িয়ে চলেছেন ফাইলটা, সিস্টেম সেভও করছে, করতে করতে ৫০০-তম লজিকাল ব্লক ব্যবহার হয়ে গেল। এবার, আপনি যতই চান, আর বাড়াতে পারবেন না। সিস্টেম জানাবে — পরস্পরসন্নিহিত ব্লকাভাবের অনিবার্য কারণবশত, আজ ধর্মঘট। পরে এই প্রসঙ্গটায় আসব আমরা, সিডিরমের আইএসও ফাইলসিস্টেম আলোচনার সূত্রে।

তার মানে, ফাইলের তথ্য আর বাড়াতে গেলে, ব্লকগুলো ছড়িয়ে ছিটিয়ে যাবেই, এবং কারনেলকে তার হিশেবও রাখতে হবে। হিশেবটা রাখা হয় দুটো ধাপে। প্রথম ধাপে, ডিরেক্ট ব্লকগুলোর ঠিকানা তুলে রাখে আইনোড। কিন্তু আইনোডে স্থানাভাবের সমস্যা থাকায় তাদের সবার ঠিকানা তুলে রাখতে পারেনা, পারে মাত্র বারোটা অব্দি। এবার, দ্বিতীয় ধাপে, থাকে কিছু ইনডিরেক্ট ব্লক। এদের নিজেদের কোনো তথ্য থাকেনা। শুধু থাকে সেইসব ডিরেক্ট ব্লকের ঠিকানা যাদের আর আইনোডে রাখা যায়নি। আইনোডের প্রথম বারোটায় ডিরেক্ট ব্লকদের ঠিকানা, আর বারোটার পর তেরো নম্বর ব্লক থেকে থাকে এই ইনডিরেক্ট ব্লকদের ঠিকানা। যেখানে গেলেই, ইনডিরেক্ট ব্লকের গায়ে লেখা ঠিকানা থেকে, কারনেল অবশিষ্ট ব্লকগুলোর হদিশ পেয়ে যায়। এবার আপনি বুঝতে পেরেছেন 'ডিরেক্ট' শব্দটার মানে কী?

আজো হলনা, পরের দিনই শুরু করা যাবে গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি। এটা সাদা-চামড়া মার্কিনীদের গুলিতে মৃত্যুমান শী-এনদের স্বপ্নের কানাডার মত না হয়ে যায়, শেষ ডিসেম্বরের বরফ জমা পাউডার নদী পেরিয়ে যেখানে তারা পৌঁছতে চেয়েছিল, কোনোদিনই পারেনি, হোক রেড ইন্ডিয়ান, এক একটা মানুষ আর কত গুলি নিতে পারে, এক একটা উপজাতি, এক একটা সভ্যতা? দুদিন বাদেই নববর্ষ, এখন তো কলকাতাতেও মার্কিন নববর্ষ হয়, পৃথিবী জুড়ে একটাই নববর্ষ, আমেরিকার নববর্ষ। কোনো এলিয়েন আক্রমন আসতে হয়নি, ইন্ডিপেন্ডেন্স ডে সিনেমার মত, এমনিতেই আমেরিকার ইন্ডিপেন্ডেন্স ডে গোটা পৃথিবীর ইন্ডিপেন্ডেন্স ডে করে দেওয়ার বিল পুলম্যানের সেই স্বপ্ন সত্যি হয়ে গেছে। কলকাতাতেও হয় মার্কিন নববর্ষ। পৃথিবীর সবাই মার্কিন হয়ে গেছে। মার্কিন নববর্ষ মানেই গোটা গোটা জাতির সভ্যতার ইতিহাসের অবশেষহীন অবলোপের সেলিব্রেশন। জাগো, কে কোথায় আছো, জাগো, তিন নম্বর বিশ্বের শুয়োরের কুকুরের এবং ছাগলের বাচ্চারা, মার্কিন নববর্ষের লেফটওভার বিতরণ হবে, জাগো, জেগে থাকো। পরের সেকশনের লেখাটাও ধরা যাবে নতুন বছরে আপনারা জেগে এবং চেগে যাওয়ার পরেই।

glt-mad@ilug-cal.org