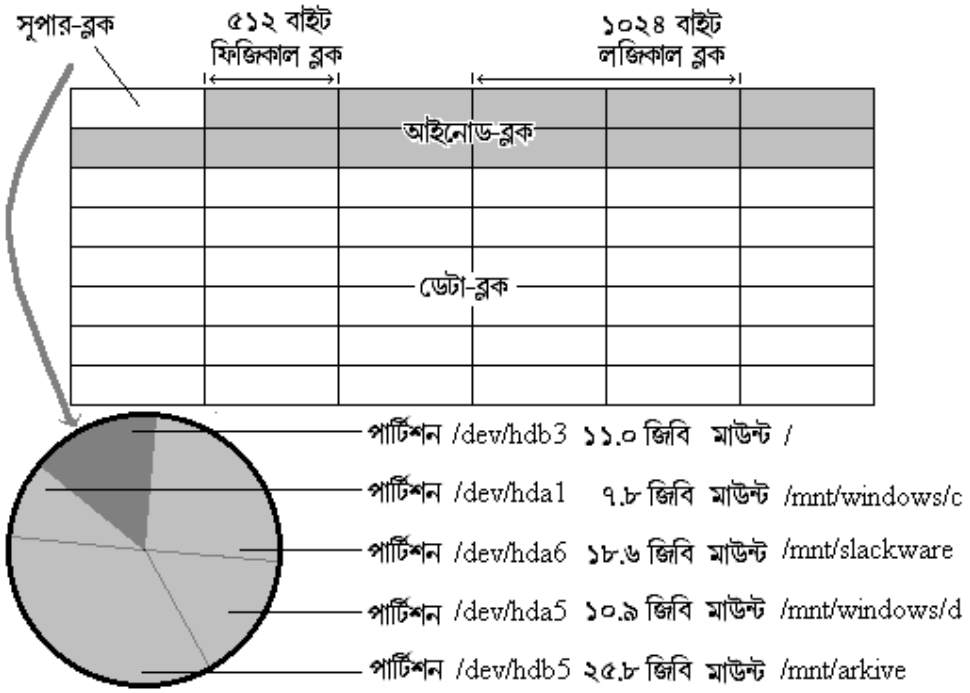
জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

আজ পাঁচই জানুয়ারি, এখন বেলা বারোটাতেও হাড় কেঁপে যাচ্ছে, আরো আমার লেখার টেবিলের পাশেই দুদুটো জানলা দিয়ে উত্তর দিকের পুকুরের হাওয়া। লেখাটায় শুধুই দেরি হয়ে যাচ্ছে, তবে শীতই তার একমাত্র কারণ নয়। এই পাঠমালার তিন নম্বর দিন মানে চার নম্বর অংশটা 'অপর' ২০০৪ বইমেলা সংখ্যায় 'অ্যাবাকাস থেকে শ্যানন — যন্ত্র চিন্তা কম্পিউটার' নামে আলাদা একটা প্রবন্ধ হিশেবে দিতে হল। পাঠমালাটার থেকে কর্তিত স্ট্যান্ড-অ্যালোন প্রবন্ধ বানানোর জন্যে আলগা ছেঁড়া একক সূত্রগুলো মেরামত করতে হল। আর ওই করতে গিয়ে শ্যাননের 'এ ম্যাথামেটিকাল থিওরি অফ কমিউনিকেশনস' নেটে পেয়ে গেলাম, সেটা পড়বার লোভটা সামলাতে পারছিলাম না। যাকগে, আবার ফেরত এসেছি গ্নু-লিনাক্স গোষ্ঠীর পিণ্ডদানে, যাকে শিক্ষিত ভদ্রলোকের বাংলায় বলে 'পেয়িং ব্যাক টু দি কমিউনিটি'।

।। দিন আট।।

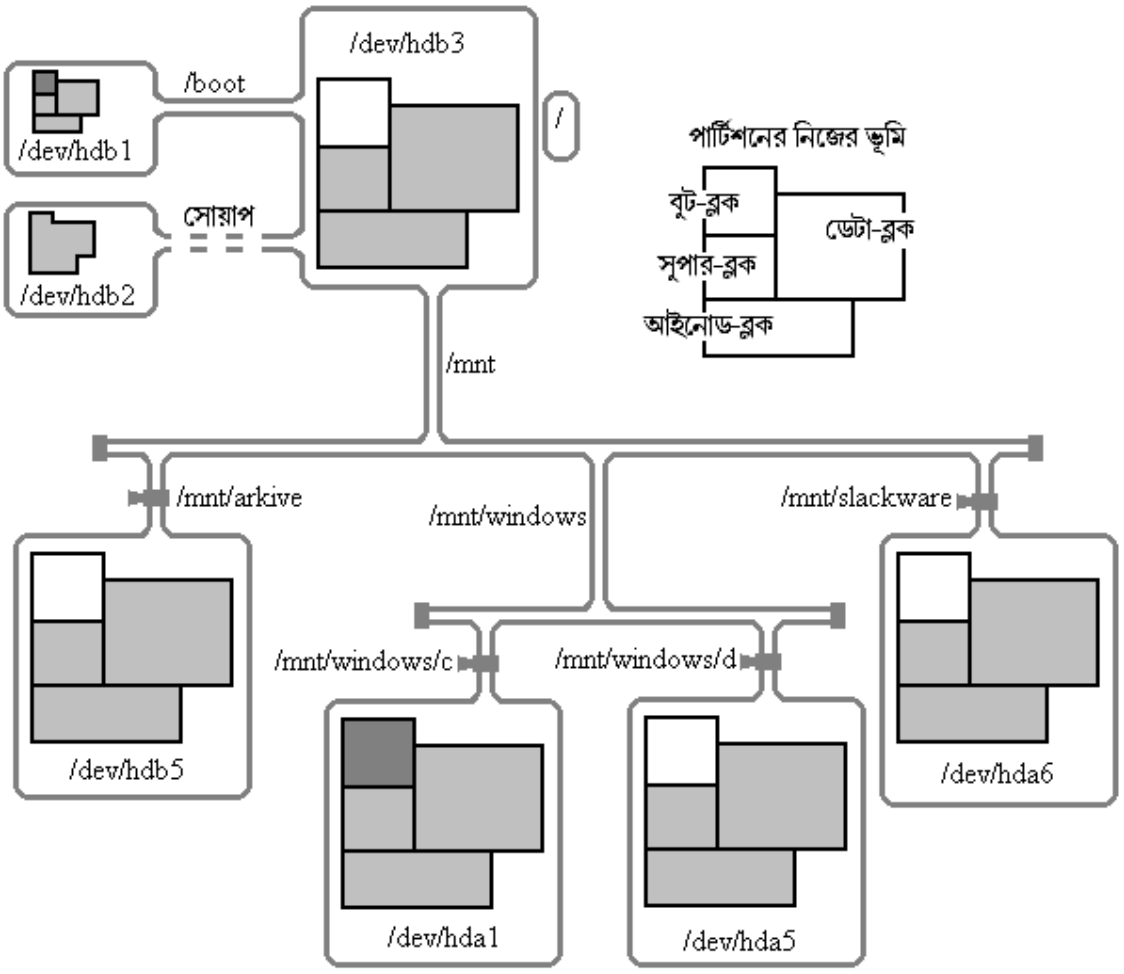
১।। ডিরেক্টরি ফাইল আর কারনেল

মাউন্টের প্রসঙ্গটা আগেও এসেছে, সাত নম্বর দিনে বারবার, প্রতিবারই মাঝপথে থামিয়ে দিতে হয়েছে। আজ সেটা বিশদে ধরব আমরা। গ্নু-লিনাক্সে সামগ্রিক ঐক্যবদ্ধ ফাইলসিস্টেমে এক একটা ডিরেক্টরি কিভাবে আলাদা আলাদা পার্টিশনের আলাদা আলাদা ফাইলসিস্টেমে মাউন্ট করা হয় (আবার, সেই 'ফাইলসিস্টেম' শব্দটা দুটো অর্থে দুবার ব্যবহার হল একই বাক্যে)। মাউন্টের প্রসঙ্গটা ধরার আগে সাত নম্বর দিনের শেষে আসা ফাইলসিস্টেমের নানা ধরনের ব্লক নিয়ে আলোচনাটা একটা ছবি দিয়ে একটু বুঝে নিই।



এই ছবিতে দেখুন, শুধু তিন রকমের ব্লক দেখানো হয়েছে। সুপার-ব্লক, আইনোড-ব্লক আর ডেটা-ব্লক। একটা ব্লক দেখানো হয়নি। কোনটা বলুন তো? বুট-ব্লক। এর পরের প্রশ্নটা এর চেয়ে একটু কঠিন। কেন বুট-ব্লকটা দেখানো হয়নি এই ছবিতে? আসছি সেটায়, দাঁড়ান। ছবির নিচের দিকে টুকরো করা কাঁচিপোড়া পিঠেটা (ঘটিরা বলে আশকে) দেখুন। এখানে আমার মেশিনের দুটো হার্ডডিস্কের মোট পার্টিশনগুলোকে দেখানো হয়েছে, তাদের আয়তনের অনুপাতে। সাত নম্বর দিনের ৩.৩ নম্বর সেকশনের তালিকার সঙ্গে মেলান, এখানে আমরা সবগুলো পার্টিশন

দেখাইনি। দুটো পার্টিশনকে বাদ দিয়েছি, সোয়াপ পার্টিশন মানে '/dev/hdb2'। আর সুজে সিস্টেমের বুট ফাইল যেখানে থাকে, সেই বুট ডিরেক্টরিতে যে পার্টিশন মাউন্ট করা হয়, মানে বুট পার্টিশন বা '/dev/hdb1'। এরা অন্যগুলোর তুলনায় বড্ড ছোট। ছবিতে দেখানোটা খুব অসুবিধের হত। কাঁচিপোড়াটাকে আমরা পাঁচটা টুকরোয় কেটেছি। '/dev/hdb3', '/dev/hda1', '/dev/hda6', '/dev/hda5', এবং '/dev/hdb5'। সাত নম্বর দিনের তালিকার সঙ্গে মেলান, এর প্রথমটা, '/dev/hdb3', হল সুজে ওএস-এর রুট পার্টিশন, মানে সুজের '/' ডিরেক্টরিটা যেখানে মাউন্ট করা থাকে। এই পার্টিশনের ব্লকগুলোকেই ছবির উপরদিকে চৌখুপি কেটে কেটে দেখানো হয়েছে। আর তাতে কোনো বুট-ব্লক দেখানো হয়নি, কারণ, এই পার্টিশনেই কোনো বুট-ব্লক নেই। সাত নম্বর দিনের 'fdisk -l' করে পাওয়া তালিকাটায় ফেরত যান। দেখুন মাত্র দুটো পার্টিশনের নামের পাশে তারা আঁকা আছে, মানে অ্যাসটেরিক্স '*' দেওয়া। পার্টিশন দুটো '/dev/hda1' আর '/dev/hdb1'। এই তারাটা খুব ভালো করে বুঝতে পারবেন 'fdisk'-এ ঢুকলে। '`fdisk`' কমান্ড দিয়ে ঢুকে '`m`' মারলে 'fdisk'-এর নিজের কমান্ডের মেনুটা পাওয়া যায়, মানে কী টিপলে ও কী করে। তাতে দেখুন একটা কমান্ড আছে 'make the partition bootable'। অর্থাৎ, সব পার্টিশন বুটনীয় বা বুটেবল নয়। শুধু সেই পার্টিশনেই এটা করতে হয় যা দিয়ে বুট করতে হবে। আমার মেশিনের সিস্টেমে '/dev/hda1' আর '/dev/hdb1' হল বুটনীয়। আর বুটযোগ্য পার্টিশনেই কেবল বুট-ব্লক থাকে। কাঁচিপোড়ার সেই টুকরোতেই কেবল বুট-ব্লক থাকবে, যা শুধু নারকেল-কোরা আর ঝোলাগুড় দিয়ে নয়, বুট পরে খেতে হয়। আর ছবিটার উপরদিকে দেখুন, ফিজিকাল ব্লক আর লজিকাল ব্লক দুটোকেই দেখানো হয়েছে। বোঝার সুবিধের জন্যে। তবে, ছবিটাকে বাস্তব করতে চাইলে আমার একটা নয় দুটো কাঁচিপোড়া আঁকা উচিত ছিল, কারণ, হার্ডডিস্ক দুটো। যাকগে। পরের মাউন্ট নিয়ে ছবিটা একবার দেখুন। এই ছবিটা এখন পুরোটা বোঝার চেষ্টা করবেন না, ধীরে ধীরে হবে।



একটা আলগা ধারণা করার চেষ্টা করুন। পাঁচটা পার্টিশনের পাঁচটা মাউন্টপয়েন্ট দেখিয়েছি, পাঁচটা ডিরেক্টরি। ব্লকের ছবিতে বাদ দেওয়া হাফটিকিট পার্টিশনদুটো, সোয়াপ আর সুজের বুট, তাদেরও দেখুন, এই ছবিতে দেখানো আছে বাঁদিকে উপরে, একটু ছোট ছোট দুটো ছকে। আর এই ছবিতেও ফাইলসিস্টেমের চার ধরনের ব্লকের ছকটা রেখেছি। প্রত্যেকটা পার্টিশনই যেন একই রকমের চার ধরনের ঘর নিয়ে তৈরি একই প্ল্যানে তৈরি বাড়ি, শুধু তাদের সাইজগুলো আলাদা। আর '/dev/hda1' এবং '/dev/hdb1' এই দুটো পার্টিশনে দেখুন, আলাদা করে বুট-ব্লক দেখানো আছে। এবং সোয়াপ পার্টিশন '/dev/hdb2' একটু অন্যভাবে দেখানো। গোটা সিস্টেমের সঙ্গে যুক্ত সছিদ্র দাগ দিয়ে, আর তার মধ্যে নানা ধরনের ব্লকের ভাগটাও নেই। এর যুক্তিটা আমরা একটু পরে বুঝতে পারব, সোয়াপ পার্টিশনের ফাইলসিস্টেম গঠনের পার্থক্যের আলোচনায়। মাউন্ট বিষয়টার কেজো জায়গাগুলো আপনারা একটু চিনেছেন এর মধ্যেই, এর গোটা আলোচনায় একটু বাদেই আসব। তার জন্যে আমাদের যেতে হবে একটু ডিরেক্টরি আর কারনেলের আলোচনায়, সেটাই এই সেকশনের হেডিং। ফাইলসিস্টেম আলোচনার জটিলতাগুলো বোঝার জন্যে সম্পর্কটা বুঝে নেওয়া জরুরি।

ডিরেক্টরি ফাইলের অপৌরুষেয় বিদঘুটেপনার কথা আমরা আগেই বলেছি। কারনেল ছাড়া কারুর পড়ার অধিকার নেই ডিরেক্টরি ফাইল। মায় রুটেরও না। কারনেল ছাড়া আর সবাই শুধু দেখতে পারে সেই ডিরেক্টরিতে কী আছে, তাও যদি সেই ইউজারের সেই ডিরেক্টরি দেখার অনুমতি থাকে। অবশ্য রুটের তো সব ডিরেক্টরি দেখারই অনুমতি আছে। একটা ডিরেক্টরি ফাইলে ডিরেক্টরির ভিতরকার প্রতিটি ফাইলের দুটো ব্যাপার লিপিবদ্ধ থাকে। এক, ফাইলটার নাম। আর দুই, ফাইলটার আইনোড নম্বর। সাত নম্বর দিনে আমরা বলেছিলাম, আইনোড ব্লকগুলোর তালিকায় ফাইলের নামটা থাকেনা। সেটা থাকে এখানে। ডিরেক্টরি ফাইলের কাছে যেহেতু ডিরেক্টরির ভিতরের প্রতিটি ফাইলের নাম আর আইনোড নম্বর দুই-ই লেখা থাকে, সহজেই ওই আইনোড নম্বর দিয়ে ফাইলটাকে সিস্টেম পেয়ে যেতে পারে। ডিরেক্টরি ফাইল আর আইনোড-ব্লক এই দুইয়ে মিলে একটা পরস্পর-সম্পর্কিত যৌথ-তালিকা তৈরি করে। ডিরেক্টরি ফাইল একটা তালিকা। আইনোড-ব্লক আর একটা তালিকা। আইনোড নম্বর হল এই দুটো তালিকার মধ্যে সম্পর্কটা। একটা ফাইলের সমস্ত কিছু যদি সিস্টেমের জানার দরকার হয় — খুব সোজা, ফাইলটার আইনোড নম্বর নাও, এবং তার সঙ্গে মিলিয়ে দুটো তালিকারই ওই বিশেষ নম্বর লাইনটা পড়ে ফেলো।

সাত নম্বর দিনে আমরা দেখলাম, যখনই একটা ফাইলের একটা লিংক বানানো হচ্ছে, তার আইনোডে উল্লিখিত লিংকের সংখ্যাটা এক এক করে বেড়ে যাচ্ছে। আর এই লিংক-ফাইলটার নতুন ফাইলনামের জন্যে ডিরেক্টরিতেও একটা নতুন এন্ট্রি তৈরি হচ্ছে। যার আইনোড নম্বরটা নিট কপি করে দেওয়া হচ্ছে সেই মূল ফাইলটার থেকে, যার লিংক এই লিংক-ফাইলটা। আপনার বানানো লিংকগুলোর যে কোনো একটাকে এবার '`rm`' করে দেখুন, লিংকটা উড়ে যাবে, আর তার সঙ্গে সঙ্গে '`ls -il`' করে বানানো আইনোড-সহ দীর্ঘ তালিকায় মূল ফাইলটার লিংক-সংখ্যা এক কমে যাবে। এখন আমরা জানি এর মানে কী — আইনোড-ব্লকে লেখা লিংক-সংখ্যাকে এক কমিয়ে দেওয়া হয়েছে। আর ডিরেক্টরি ফাইলে এই লিংকটার জন্যে বানানো এন্ট্রিটাও উড়িয়ে দেওয়া হয়েছে। একটা ফাইলের শেষ লিংকটা উড়িয়ে দেওয়া হলে, তখন ফাইলটাই মুছে গেল। তার ডেটা-ব্লকগুলো ধুয়ে মুছে পোস্কার করে রাখা হল, নতুন ফাইল লেখার জন্যে। আপনার সঙ্গে এই পৃথিবীর শেষতম লিংক তো আপনি নিজেই, এটা কখনো খেয়াল করেছেন? আমাদের মতো যাদের ছদ্মনামে লেখার অভ্যেস তৈরি হয়, তাদের কাছে এটা খুব স্পষ্ট। ত্রিদিব সেনগুপ্তর লেখা সামনে এনে ধরে দিতে পারে কেবল একমাত্র দীপঙ্কর দাশই, ত্রিদিব সেনগুপ্ত হল দীপঙ্কর দাশের লিংক ফাইল। ত্রিদিব সেনগুপ্ত মুছে গেলেও দীপঙ্কর দাশ থাকবে। কিন্তু দীপঙ্কর দাশ নিজে দুনিয়ার ডিস্ক থেকে মুছে গেলে, আর সে নিজেই নিজের লেখাকে হাজির করতে পারবে না। তার জায়গায় অন্য ফাইল এলো, এখন থেকে অন্য কেউ অপরের লেখা দিতে দেরি করায় সোমনাথের কাছে গালাগাল খাবে।

এবার, কারনেলের কাছে, সিস্টেমের কাছে, একটা ফাইলের সমস্ত হালহকিকত, সমস্ত ঠিকানা কী ভাবে থাকে, সেটা আমাদের কাছে স্পষ্ট। এবার ধরুন আপনি আপনার তৈরি 'onefile' ফাইলটায় কী আছে দেখতে চাইলেন। কমান্ড দিলেন, '`cat onefile | less`'। সামনে স্ক্রিন জুড়ে গোটা গোটা অক্ষরে 'onefile' ফাইলের প্রথম পাতা ভেসে উঠল। 'cat' রিডাইরেক্ট করে 'less'-কে দেওয়া আছে, তাই আপনি এখন স্ক্রিন বাই স্ক্রিন দেখতে পাবেন ফাইলটাকে। কিন্তু স্ক্রিনকা পিছে কেয়া হ্যায়? আপনার অগোচরে কী কী ঘটল? আপনার আদেশটা ব্যাশের হাত ঘুরে

পৌঁছল কারনেলের কাছে। কারনেল এই যে 'onefile' ফাইলটাকে খুঁজে এনে আপনার স্ক্রিনে মানে স্ট্যান্ডার্ড আউটপুটে পেতে দিল, তার আগে তাকে 'onefile' ফাইলটার ঠিকানা খুঁজে পেতে হল। সেই ঠিকানা থেকে তাকে হদিশ করে নিতে হল ফাইলটার শুরু আর শেষ ব্লকের সাকিন। তারপর সে পড়া শুরু করল। আপনি এক স্ক্রিন এক স্ক্রিন করে দেখছেন 'less'-এর দৌলতে, কিন্তু কারনেলকে এক সঙ্গে পড়ে নিতে হল গোটাটাই। কারণ, আপনাকে ওই ভাবে দেখানোর আগে তাকে তো গোটা টেক্সটাকে স্ক্রিনে পাতায় লাইনে ভেঙে নিতে হবে। এবার জাস্ট এই পড়ে ফেলা অব্দি অগোচরে ঘটা গোটা কাজটাকে স্টেপ বাই স্টেপ ধরা যাক।

এক। সাত নম্বর দিনের একদম শেষ দিকের আলোচনায় বলেছি, কারনেল ইউজারের কারেন্ট ডিরেক্টরির, মানে এখন যেখানে আছে, কাজ করছে, যেখানে দাঁড়িয়ে কমান্ড দিচ্ছে, সেই ডিরেক্টরির ডিরেক্টরি ফাইলের আইনোড নম্বরকে মেমরিতে তুলে নেয়। এই আইনোড নম্বরটাকে কারনেল এবার খুঁজে বার করে আইনোড-ব্লকগুলো থেকে। সেখান থেকে এই ডিরেক্টরি ফাইলের আইনোডটা বার করে।

দুই। এই ডিরেক্টরির আইনোড থেকে এবার কারনেল বার করে এই ডিরেক্টরি ফাইলের ডেটা-ব্লকের ঠিকানা।

তিন। ডিরেক্টরি ফাইলের ডেটা-ব্লক পড়ে কারনেল এবার খুঁজে পায় 'onefile' ফাইলটাকে এবং পেয়ে যায় 'onefile' ফাইলটার আইনোড নম্বর।

চার। 'onefile' ফাইলের আইনোড নম্বর পেয়ে যাওয়া মাত্রই কারনেল আবার ফেরত যায় আইনোড-ব্লকে। সেখানে গিয়ে 'onefile' ফাইলের আইনোড খুঁজে বার করে।

পাঁচ। এই আইনোড থেকে কারনেল এবার পড়ে নেয় 'onefile' ফাইলটার আয়তন, এবং ডিস্কের শরীরে তার ডেটা-ব্লকের ঠিকানাগুলো। এখানে ইনডিরেক্ট ব্লক যদি থাকে তাহলে সেইটাও পড়ে ফেলে কারনেল, সেখান থেকে বাড়তি ডেটা-ব্লকের ঠিকানাগুলো পেয়ে যায়।

ছয়। এবার কারনেল যায় তার ডিস্ক ড্রাইভারের কাছে। তাকে গিয়ে বলে তোমার ডিস্কের হেডগুলোকে নড়িয়ে তুমি তোমার প্ল্যাটারের গায়ে এত নম্বর সিলিন্ডারের এত এত সেক্টর থেকে এই এই ব্লকগুলোকে পড়ে ফেলো। এই হল ফাইলটার আয়তন, এই বাইটসাইজ হওয়া অব্দি তুমি পড়ে চলো। এইটা মিলে গেলে কারনেল বোঝে যে তার গোটা ফাইলটা পড়া হয়েছে, এবং তখন তার পড়ে ফেলা তথ্যটা ফুটিয়ে তোলার পালা, তার জন্যে সে যাবে ডিসপ্লে ড্রাইভারের কাছে। সে অন্য গল্প।

২।। সোয়াপ পার্টিশনের স্বতন্ত্র ফাইলব্যবস্থা

এতবার আলাদা আলাদা ভাবে দুটো অর্থে ব্যবহার করা হয়েছে যে আমরা এখন 'ফাইলসিস্টেম' শব্দটার এই দুটো ব্যবহারের সঙ্গে অভ্যস্ত। আমরা যখন '/' থেকে শুরু করে গোটা ডিরেক্টরিব্যবস্থাটার কথা ভাবছি, সেটা একটা ঐক্যবদ্ধ ফাইলব্যবস্থা বা ইউনিফায়েড ফাইলসিস্টেম। মাউন্ট করা বিভিন্ন পার্টিশনের ছবিটা, মানে আজকের দ্বিতীয় ছবিটা দেখুন, এর মধ্যে বিভিন্ন আলাদা আলাদা পার্টিশন মাউন্ট করা আছে বিভিন্ন আলাদা আলাদা ডিরেক্টরিতে। যেমন '/dev/hdb5' পার্টিশন আছে '/mnt/arkive' ডিরেক্টরিতে, '/dev/hda6' পার্টিশন আছে '/mnt/slackware' ডিরেক্টরিতে, আর '/dev/hda1' এবং '/dev/hda5' পার্টিশন দুটো মাউন্ট করা আছে '/mnt/windows' ডিরেক্টরির ভিতর '/mnt/windows/c' এবং '/mnt/windows/d' এই দুই সাবডিরেক্টরিতে। এই পার্টিশনগুলো আপনা থেকেই এই এই ডিরেক্টরিতে মাউন্ট করা থাকতে পারে, যাকে বলে অটোমাউন্ট, আবার নাও-পারে। এমনকি চাইলে অন্য কোনো ডিরেক্টরিতেও মাউন্ট করা যেতে পারে, সেটা আমরা একটু বাদেই দেখব। কিন্তু পার্টিশন মাউন্ট করা থাকুক আর নাই-থাকুক, ডিরেক্টরিটা কিন্তু বানানো আছেই। সেটাই আমরা দেখিয়েছি যেন নলপথ আর তা আটকে দেওয়ার স্টপকক দিয়ে। যেন এক একটা পার্টিশন, হার্ডডিস্কের এক একটা অংশ — তারা এক এক টুকরো আলাদা আলাদা রসদ।

আলাদা আলাদা পার্টিশনগুলোকে দেখানো হয়েছে আলাদা আলাদা চৌবাচ্চায়। তাদের সঙ্গে মূল চৌবাচ্চার যোগাযোগ ঘটে হায়েরার্কিকাল ডিরেক্টরি-ব্যবস্থার নলপথ বেয়ে। এই নলপথটা খোলাও থাকতে পারে, আবার বন্ধও থাকতে পারে। খোলা থাকে, যখন পার্টিশনটা মাউন্টেড। আর যখন মাউন্টেড নেই, তখন বন্ধ। মূল '/dev/hdb3'

চৌবাচ্চাটা দেখুন, এটাতেই মাউন্ট করা থাকে '/' ডিরেক্টরি। '/boot' ডিরেক্টরিতে মাউন্ট করা হয় '/dev/hdb1' পার্টিশনটা। এই মূল পার্টিশন '/dev/hdb3' আর বুট পার্টিশন '/dev/hdb1' নলপথ দিয়ে সংযুক্ত। কিন্তু সেই নলে কোনো স্টপকক নেই। মানে, এই পার্টিশনটা সবসময়েই মাউন্টেড আছে। সুজে তার মূল রুট পার্টিশনে কাজ করছে মানেই সে বুট পার্টিশনটা মাউন্ট করে নিয়েছে। মাউন্ট করার পর সেখান থেকে ফাইল পেয়েই সে বুট করতে পেরেছে নিজের ওএস-এ। বুট পার্টিশন যদি আলাদা করে দিয়ে দেওয়া না হয়, তাহলে স্বাভাবিকভাবেই, '/boot' ডিরেক্টরিটা থাকে স্ল্যাশ বা '/' ডিরেক্টরির ভিতরে, মানে রুট পার্টিশনেই।

ছবিতে এর ঠিক নিচেই রয়েছে '/dev/hdb2' পার্টিশন। কিন্তু তার কোনো নলপথ মানে ডিরেক্টরিপথ নেই। নলের দাগদুটো তাই, দেখুন, ভাঙা-ভাঙা আঁকা হয়েছে। এবং এর পাশে লেখা 'সোয়াপ'। এর মানে, এই সোয়াপ পার্টিশনটাও সবসময়েই সংযুক্ত, কিন্তু সেটা একটা সম্পূর্ণ অন্যরকম ব্যবস্থা। স্বাভাবিক ফাইলব্যবস্থা বা ফাইলসিস্টেম থেকে একদম স্বতন্ত্র। এখানে আমরা ফাইলব্যবস্থা বলতে বোঝাচ্ছি ফাইল লেখা বা রাখা বা পড়ার রকম, যার নানা আলাদা আলাদা ব্যবস্থা হয়, আমরা আগেই বলেছি, ইএক্সটিটু রাইজারএফএস এক্সএফএস ইত্যাদি। কিন্তু সেই সবগুলো ফাইলব্যবস্থা এক দিকে, আর সোয়াপ ব্যবস্থা অন্য দিকে। সোয়াপ একদম আলাদা, তার কারণটা তো আপনারা জানেনই, সোয়াপ ফাইলের জন্মই হয় মৃত্যুর জন্যে, নিজে মরে গিয়ে অন্য ফাইলকে বা ফাইলব্যবস্থাকে বদলে যাবে বলে, অস্থায়ী ভাবে স্মৃতিকে লিখে রাখার একটা জায়গা সোয়াপ-ফাইল, আমরা বারবার বলেছি। সোয়াপ ফাইল হল র‍্যামের সহচর, র‍্যামকে বাড়িয়ে তোলে। তাই এই পার্টিশনের ফাইল লেখা/রাখা/পড়ার কায়দা অন্যরকম হবেই।

অর্থাৎ, একটা মূল পার্টিশন যা মাউন্ট হবে '/' ডিরেক্টরিতে, আর একটা সোয়াপ পার্টিশন — এই দুই রকমের দুটো পার্টিশনে দুই ধরনের ফাইল-ব্যবস্থা, এটা একটা গ্নু-লিনাক্স সিস্টেমের ন্যূনতম প্রয়োজনীয় শর্ত। এরপর কী কী পার্টিশন থাকবে কী কী ফাইলব্যবস্থায় সেটা আপনার ইচ্ছের এবং প্রয়োজনের উপর নির্ভর করছে। জরুরি এই দুটো পার্টিশনের প্রথমটাকে বলা হয় রুট পার্টিশন আর দ্বিতীয়টাকে সোয়াপ পার্টিশন। এমনকি খুব বিদঘুটে বিতিকিচ্ছিরি একটা অবস্থায়, যাকে ইংরিজিতে বলে পরিস্থিতি, এই সোয়াপ পার্টিশনও না-রাখা যায়। কিন্তু রুট পার্টিশন না-থাকা মানে সিস্টেমটাই না-থাকা। গ্নু-লিনাক্স একটা সিস্টেমে এই রুট পার্টিশনেই থাকে সিস্টেমের গোটা কাঠামোটা। রুট ডিরেক্টরি এবং তার থেকে বেরোনো সমস্ত সাবডিরেক্টরির কাঠামোটা মনে করুন, ছ নম্বর দিনে এসেছিল, সাতেও। এই রুট পার্টিশনের মধ্যে ওই হায়েরার্কিকাল ডিরেক্টরি ব্যবস্থাটা থাকে, থাকে সমস্ত অবশ্য প্রয়োজনীয় ফাইল এবং প্রোগ্রাম। রুট পার্টিশন ছাড়া অন্য কোনো পার্টিশন যদি না-থাকে, তখন অন্য পার্টিশনগুলোয় মাউন্ট করা ডিরেক্টরিতে যে যে ফাইল আর ডিরেক্টরি থাকত, তারা রুট পার্টিশনে মাউন্ট করা '/' ডিরেক্টরির ভিতরে তাদের নিজের নিজের বরাদ্দ ডিরেক্টরিতেই থাকে, শুধু তাদের জন্যে আলাদা করে কোনো পার্টিশন থাকেনা। তারা সবাই থাকে রুট পার্টিশনেই।

সোয়াপ পার্টিশন অবশ্যই থাকা উচিত একটা সিস্টেমে, কাজ করার সুবিধের জন্যেই। কারনেলের কাজের স্বার্থেই। এক আর দুই নম্বর দিনে বিভিন্ন ধরনের মাল্টিপ্লেক্সিং-এর আলোচনার সূত্রে আমরা যে একত্র চলমান অগণ্য প্রসেস বা প্রক্রিয়ার কথা বলেছি, সেই প্রক্রিয়াগুলোর প্রতিটির গতিশীলতাকে নিয়ন্ত্রণ করে কারনেল, প্রসেসগুলোর কাজকে নিয়ন্ত্রণ করে, এবং নিয়ন্ত্রণ করে কাজ করতে গিয়ে সিস্টেমের নানা রসদের ব্যবহারকে। নিয়ন্ত্রিত সীমিত রসদকে সুবিধেজনক ভাবে বিভিন্ন প্রক্রিয়ার মধ্যে ভাগবাঁটোয়ারার কাজে কারনেলকে সহায়তা করে সোয়াপ। অনেকগুলো সমান্তরাল প্রক্রিয়ার ভারে সিস্টেম যখন ধুঁকছে, সিস্টেমের মেমরির পিঠ বেঁকে গেছে বোঝা তুলতে তুলতে, তখন কিছুটা বোঝা অস্থায়ী ভাবে এই সোয়াপে নামিয়ে কারনেলের একটু দম ফেলার ব্যবস্থা করে। 'সোয়াপ' শব্দটার অর্থ একটা কিছুর সঙ্গে অন্য কিছুকে বদলানো। এখানেও তাই ঘটে, র‍্যামের জ্যান্ত চলমানতাকে বদলে ফেলা হয় সোয়াপের অস্থায়ী আরামে। তারপর, যেই ফুরসত আসে, একটু আরাম-পাওয়া প্রক্রিয়াটাকে ফের চালাতে শুরু করে কারনেল, সোয়াপ থেকে তুলে র‍্যামে ফেরত এনে। সোয়াপ পার্টিশনের বিশেষ ধরনের ফাইলব্যবস্থার বৈশিষ্ট্য এই যে, একে সরাসরি কোনো ইউজার পড়তে পারেনা, এমনকি রুটও না। সোয়াপকে ব্যবহার করে শুধু কারনেল। সিস্টেমে বসে আপনি সোয়াপ সম্পর্কে আর একটু কিছু জানতে চাইলে, তার উপায় কী? '`man -k swap`'

কমান্ডটা দিন। স্ক্রিনে অনেকটা লেখা ফুটে উঠলে তাকে ফাইল বানিয়েও পড়তে পারেন। তবে এখানে তার দরকার পড়বে না। সোয়াপটা মোটামুটি ভদ্রগোছেরই আছে। মাত্র আট দশটা লাইন। এখানে তার কয়েকটা লাইন তুলি।

```
mkswap(8)   - set up a Linux swap area
swapon(2)   - start/stop swapping to file/device
swapoff(2) - start/stop swapping to file/device
swapon(8)   - enable/disable devices and files for paging and swapping
swapoff(8) - enable/disable devices and files for paging and swapping
```

ওই গোটা দশেক লাইনের থেকেও কয়েকটা ছেঁটে দিলাম, এত বেশি টেকনিকাল সেগুলো। এবার দেখুন তো, নিজে পড়ে, কী আন্দাজ পাচ্ছেন কমান্ডগুলো সম্পর্কে? 'mkswap' স্বাভাবিক ভাবেই, সোয়াপফাইল বানায়। মানে, যে পার্টিশনটাকে আপনি সোয়াপ পার্টিশন বানাতে চান, সেই পার্টিশনে সোয়াপ ফাইল লেখার বানানোর ব্যবহার করার মত করে নিজস্ব স্বতন্ত্র ফাইলব্যবস্থা বানিয়ে তোলে 'mkswap'। সচরাচর এটা চালু প্রথা যে নতুন ব্যবহারকারীর গ্নু-লিনাক্স ইনস্টলেশন করে দেয় তার চেনাজানা কোনো লিনাক্সী। এমনকি আপনি নিজেও যদি ইনস্টল করে থাকেন, তাহলেও, ইনস্টল করার সময়, আপনি আসলে এই কমান্ডটাই ব্যবহার করেছেন। ইনস্টলের সময় দিতে হয়েছে, কোথায় সোয়াপ পার্টিশন হবে, কতটা তার সাইজ। চালু প্রথায় মোটামুটি র‍্যামের ডাবল সাইজ বরাদ্দ করে দিয়েছেন সোয়াপের, কোথায় বানাবে পার্টিশনটা সেটাও বলে দিয়েছেন। অনেক ডিস্ট্রোয় সিস্টেম নিজেও সেটা করতে পারে, মোটামুটি গড় একটা ব্যবস্থা ছকে নিয়ে। সিস্টেম নিজেই করুক, অন্য কেউ করে দিক, আপনি করে থাকুন, শেষ অব্দি ব্যবহার হয়েছে এই 'mkswap'। সামনে আপনি কোনো ছবিওলা ঝিনচাক দেখানোর দাঁত দেখেছেন — হলুদ লাল নীল ধূসর, কোনোটা ইএক্সটিটু, কোনোটা রাইজার, কোনোটা ফ্যাটথার্টিটু, কোনোটা পদিপিসির বর্মিবাক্স, তারা এই রোগা হচ্ছে, এই মোটা হচ্ছে, এই রং পালটাচ্ছে, আপনার ক্লিকের সঙ্গে তাল রেখে। সম্মুখের সেই সার্কাসের পিছনে খাওয়ার দাঁত মানে 'mkswap' দিয়েই হার্ডডিস্ক ভূমিটাকে চিবিয়ে চিবিয়ে সোয়াপ ফাইল লেখার মত করে সাইজ করেছে সিস্টেম। কোন পার্টিশনে আপনি সোয়াপ বানাবেন সেটা সিস্টেমকে দিয়েছেন 'swapon' ব্যবহার করে। আপনি হয়ত ইঁদুর টিপেছেন, ব্যাক-এন্ডে ছিল 'swapon'। তেমনি আপনি যদি কোনো পার্টিশনে সোয়াপ ফাইল আর না-রাখতে চান, সেটাকে অন্যকাজে ব্যবহার করতে চান, তখন, আপনার ইঁদুরের নাড়িভুড়ি বেয়ে দৌড় করেছে 'swapoff'। এবার আপনার এই 'হয় হয় কিন্তু Zানতে না-পারা' ক্যালিটা সরাসরি কমান্ড প্রম্পটে শো করে দিয়ে ইঁদুরের পেট না-চুলকে নিজের ইগোর পিঠ-চুলকে দিতে পারেন। কিন্তু দেখবেন, ভালো করে পার্টিশন ছক না-বুঝে সিস্টেমকে কাতুকুতু দেবেন না, গোটাটা ঘেঁটে যাবে। কমান্ডগুলো জানবেন কী করে? মরুঅঞ্চলের উদ্ভিদের বাংলার পলিমৃত্তিকায় কৃষির সাফল্য বিষয়ে গবেষণা নিয়ে কী একটা গান আমাদের বারবার কানে আসছে না, এই পাঠমালায়?

ও, একটা কথার উত্তর দিন তো, এই তালিকাটায় 'swapon' আর 'swapoff' কমান্ডদুটোর দুবার করে দুটো আলাদা নম্বরের ম্যানপেজের কথা বলার মানে কী? দেখুন, দুটো কমান্ডই দুবার করে আছে একবার '2' আর একবার '8'। তার মানে একটা দেখতে হবে '`man 2 swapon`' বা '`man 2 swapoff`' কমান্ড দিয়ে, আর একবার দেখতে হবে '`man 8 swapon`' বা '`man 8 swapoff`' কমান্ড দিয়ে। কিন্তু কেন? যদি না বুঝতে পারেন ছয় নম্বর দিনের গোড়ার দিকটা একবার উল্টে আসুন। গোটা পুরোনো আলোচনাটা আপনার মাথায় একদম হাতেগরম থাকা চাই। সচরাচর টেক্সটগুলোয় যা হয়, ঠিক গঠনের নিয়ম মেনে প্রসঙ্গগুলো পরপর আসে, তাতে একটু বোরিং হয় ঠিকই, কিন্তু একটা সুবিধাও থাকে, এগোনোর পিছনোর, আটকে গেলে পিছিয়ে গিয়ে বুঝে-আসার। একদম বিষয়-নির্ভর বা অবজেকটিভ পাঠ। এই লেখাটা তার বিপরীত, ধরুন বিষয়ী-নির্ভর বা সাবজেকটিভ পাঠ। একজন জ্যান্ত কেউ সিস্টেমটা বুঝতে বুঝতে চলেছে, যেখানে তার যেটুকু আটকে যাচ্ছে সেটুকু আলোচনা করে নিচ্ছে। জ্যান্ত ইউজার বেশি উপস্থিত ব্যবহারের বিদ্যার চেয়ে। আপনার দায়টাও তাই বেশি, গোটাটা নিজের মাথায় তুলতে তুলতে চলার।

৩।। নানা ধরনের ফাইলব্যবস্থা

আগের সেকশনে আমরা বললাম রুট আর সোয়াপ — ফাইলব্যবস্থার এই দুটো বড় বিভাগের কথা। এবার এই রুট পার্টিশনের ফাইলব্যবস্থাটা ঠিক কী রকম হবে? তার নানা প্রকারভেদ আছে। সোয়াপ ফাইলব্যবস্থায় যা নেই। আমার মেশিনের একই সোয়াপ পার্টিশন সুজে আর স্ল্যাকওয়ার দুজনেই সমানভাবে ব্যবহার করে, যখন যেটা দিয়ে বুট

করি। আগে যখন সুজে স্ল্যাকওয়ার আর রেডহ্যাট তিনটেই ছিল, তিনটেই তাই করত। সোয়াপ ফাইলব্যবস্থার প্রকারভেদ হয়না। কিন্তু রুট পার্টিশনের ফাইলব্যবস্থার হয়। আর শুধু তো রুট পার্টিশন না, আরো পার্টিশন থাকতে পারে। তারা নানারকম ফাইলসিস্টেম হতে পারে, আগেই বলেছি। সিস্টেমের নিরাপত্তা, ব্যবহারের সুবিধার জন্যেও নানা পার্টিশন রাখতে হয়, জানেন সেটা। আপনার গ্নু-লিনাক্স সিস্টেমে একটা বিশেষ মুহূর্তে মোট কত রকমের ফাইলব্যবস্থা ব্যবহৃত হচ্ছে, সেটা জানার একটা উপায় হল, 'cat /proc/filesystems'। এই '/proc' ডিরেক্টরিটা আমরা ঐক্যবদ্ধ ফাইলসিস্টেমের ডিরেক্টরি কাঠামোর ছবিতে দেখেছি, এখানে কী থাকে সেটায় আমরা একটু বাদে আসব গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির আলোচনায়। এই যে একটা বিশেষ মুহূর্তের কথা বললাম, এটা খুব গুরুত্বপূর্ণ। ধরুন, আপনার সিস্টেমে একটা পার্টিশন আছে যেখানে ফাইল লেখার রাখার পড়ার ব্যবস্থাটা হল রাইজারএফএস। সেই পার্টিশনটা আপনি সবসময় মাউন্ট করেন না, যখন দরকার পড়ে মাউন্ট করে নেন। আমার সিস্টেমে '/dev/hdb5' আর '/dev/hda6' পার্টিশনদুটো যেমন। যখন দরকার পড়ে মাউন্ট করে নিই '/mnt/arkive' আর '/mnt/slackware' ডিরেক্টরিতে। মাউন্ট কখন করবে, আপনা থেকেই একটা পার্টিশনে সিস্টেম মাউন্ট করে নেবে কিনা — এসব নিজে ঠিক করে নেওয়া যায়, '/etc/fstab' বলে একটা ফাইল মারফত, আসব সেকথায়। আমার সিস্টেমে এছাড়া আর কোনো পার্টিশনে রাইজারএফএস নেই। যখন সিস্টেমে এই পার্টিশনদুটো মাউন্ট করা নেই, তখন 'cat /proc/filesystems' দিলে, তালিকায় রাইজার দেখায় না, কিন্তু মাউন্ট করা থাকলে দেখায়।

```
2004-01-07 11:07                                   Page 1

rootfs          tmpfs           minix           usbdevfs
bdev            shm             iso9660         usbfs
proc            pipefs          nfs             reiserfs
sockfs          ext2            devpts          vfat
futexfs         ramfs           xfs
```

আমার সিস্টেমে সবগুলো পার্টিশন মাউন্ট করা অবস্থায় 'cat /proc/filesystems' দিয়ে যে তালিকাটা পেলাম সেটা এখানে তুলে দিলাম। অবশ্যই এই চার স্তম্ভে, তারিখ সময় সহ পাইনি, সেটাকে করে নিতে হয়েছে 'pr' বলে একটা কমান্ড-সহ, 'cat /proc/filesystems|pr -4>filesystems' কমান্ড দিয়ে। প্রথমে তালিকাটাকে পাইপ করা হয়েছে 'pr'-এর কাছে। 'pr' তালিকাটাকে চার স্তম্ভে স্তম্ভিত অবস্থায় রিডাইরেক্ট করেছে 'filesystems' নামের ফাইলে। 'pr' আসলে প্রিন্ট সংক্রান্ত কমান্ড। লাইন প্রিন্ট করার কাজে ব্যবহারের, কিন্তু এই সব কাজে বেড়ে ব্যবহার করা যায়, পাতার নম্বর, কলাম, রো, হেডার, ফুটার, সব ঠিক করে দেয় একটা টেক্সট ফাইলে। ব্যবহার করে দেখুন। সেই ফাইল থেকে তুলে দিলাম, একটু ছাঁটকাট করে।

যখন ম্যানপেজ পড়ছেন, তখন তাকেও রিডাইরেক্ট করে একটা ফাইলে এনে সেটাকে এইভাবে 'pr' দিয়ে ফরমাট করে নিতে পারেন। পড়ার সুবিধে হয়। এছাড়া ম্যানপেজ থেকে ওয়েবপেজ বা '*.html' বা অন্য কোনো ফরম্যাটে বদলে নেওয়ার অনেক উপায় আছে। আমি ব্যবহার করি 'rman'। ধরুন আপনি 'ls'-এর ম্যানপেজ পড়তে চাইছেন। তাকে রিডাইরেক্ট করে ফাইল বানিয়ে নিতে পারেন 'man ls>lsman'। একে 'pr' দিয়েও ফরমাট করে নিতে পারেন। বা 'rman' দিয়ে, 'man ls|rman -f html>lsman.html'। এতে 'ls' কমান্ডের ম্যানপেজটাকে 'rman' একটা ওয়েবপেজ করে দেবে যার নাম 'lsman.html'। একে আপনি আপনার ইন্টারনেট ব্রাউজার দিয়ে পড়তে পারবেন। কমান্ড প্রম্পটে ব্যবহার করার জবর দুটো টেক্সট মোড ব্রাউজার হল 'links' আর 'lynx'। আরো কী কী আছে, মনে করতে পারছি না। এই ব্রাউজারদুটো দিয়ে চাইলে ছবিসহ একদম স্বাভাবিক চেহারাতেও ব্রাউজ করা যায়, তখন আলাদা করে '-graphics' অপশান দিতে হয়। আমি অনেকসময়ই নেটে ব্রাউজ করি টেক্সট মোডে। এতে সুবিধে হল, প্রচুর গাম্বাট লোক তাদের ওয়েবপেজে গাড়লের মত দুম্বো দুম্বো সব গ্রাফিক্স আর জাভাস্ক্রিপ্ট দিয়ে রাখে। ওদের তো আর ডায়াল-আপ কানেকশনে কাজ করতে হচ্ছে না, অসুবিধে কী। খরচ আপনার, লোড হতে হতে আপনি আরো দুচারবার আবার আসিব ফিরে এই বাংলায় হয়ে যাবেন। 'links' বা 'lynx' টেক্সট মোডে নিট দরকারি জিনিষটুকু এনে দেয়, দরকার পড়লে জাস্ট একটা 'd' মারো, ডাউনলোড করো। আর আপনার ফরম্যাট করা ম্যানপেজগুলো বা যে ম্যানপেজের সংগ্রহ এমনিতেই দেওয়া আছে আপনার সিস্টেমে তাদেরও পড়তে পারেন এই দিয়ে। কমান্ড প্রম্পটে থাকলে তো ভালোই, যদি এক্স-উইনডোজেও থাকেন, একটা

কনসোল খুলুন, নেটে কানেক্ট করে কমান্ড দিন 'links www.geocities.com/ddipankardas/'। আর যদি ব্যাকগ্রাউন্ডে জিএলটির লোগোর অনবদ্য ছবিটা সহ (অনবদ্য না হয়ে উপায় আছে, আমার করা) পাতাটা দেখতে চান, তখন জাস্ট একটা '-g' যোগ করে দিন কমান্ডটার সঙ্গে। ছবিদুটো কুচো সাইজের বললেও কম বলা হয়, 'glt-logo.png' আর 'glt-bkg.png' দুটোই দুই কেবির কম। এটা ইয়াহু জিওসিটিতে মধ্যমগ্রাম জিএলটির সাইট, বানানো এখনো শেষ হয়নি, উপরের কয়েকটা মাত্র লিংক কাজ করছে। এই লেখাটা শেষ হলে, তথাগত অশেষ সঙ্কর্ষণ ওদের সব পোকাবাছা হয়ে গেলে সবগুলো চ্যাপ্টারই থাকবে এখানে, পিডিএফে। 'links' কমান্ড দেওয়ার পরে যদি দেখায় 'command not found', তার মানে, আপনার সিস্টেমে 'links' ইনস্টল করা নেই, জাস্ট ইনস্টলেশন সিডিটা দিয়ে ইনস্টল করে নিন। 'rman'-ও তাই। সব কমান্ডই তাই, খুঁজে পাওয়ার মানে কী সেটা তো এখন জানেন। যাকগে, এবার আসা যাক ফাইলসিস্টেমের তালিকায়।

এই ফাইলসিস্টেমের নামের তালিকা থেকে দেখুন, কয়েকটা নাম আপনার চেনা, অন্য কয়েকটা নয়। দু-চারটে কথা বলে নেওয়া যাক। শুরু করা যাক গ্নু-লিনাক্সে সবচেয়ে বেশি জনপ্রিয় ফাইলসিস্টেমগুলো থেকে। কবছর আগেও বিষয়টা ছিল খুব সরল। ব্যবহারযোগ্য ফাইলব্যবস্থা গ্নু-লিনাক্সে দুই কি ম্যাক্সিমাম তিনের বেশি ছিলনা। মূলত ব্যবহার হত ইএক্সটিটু বা থ্রি, নিদেন রাইজারএফএস। এখন তা নয়, অনেকগুলো ফাইলসিস্টেম রেগুলার ব্যবহার হয়ে চলেছে অজস্র গ্নু-লিনাক্স সিস্টেমে। ইএক্সটিটু ক্রমে বাতিল হয়ে যাচ্ছে। পরিবর্তনের গড়িমসিতে প্রখ্যাত রেডহ্যাট লিনাক্সও তার ভার্শন নাইন থেকে ডিফল্ট করে দিয়েছে ইএক্সটিথ্রি। ম্যানড্রেক সুজে এগুলো ছেড়েই দিন। এমনকী শুদ্ধতাবাদী স্ল্যাকওয়ারও তার ইনস্টল সিডির কারনেলগুলোর ভিতর একটা এক্সএফএস কারনেল দিচ্ছে। কারনেল ভারশন ২.৪ থেকে গ্নু-লিনাক্স অনেকগুলো ফাইলসিস্টেম ব্যবহার করার সুযোগ দেয়। তার অনেকগুলোর খুঁটিনাটিই দেখবেন দেওয়া আছে '/usr/src/linux/Documentation' ডিরেক্টরির টেক্সট ফাইলে।

এখানে আমরা একটা ফাইলসিস্টেমেরও গঠনের খুঁটিনাটি জটিলতাতেও যাবনা, সে এক বীভৎস বিরাট ব্যাপার। কিছুটা দেওয়ার প্ল্যানে অনেকটা পরিশ্রম করলাম কয়েকদিন ধরে। সঙ্কর্ষণও একটা ভালো লিংক দিয়েছে, আইবিএম সাইটে, ড্যানিয়েল রবিনস-এর দশটা নিবন্ধ, http://www-106.ibm.com/developerworks/library/l-fs.html, লিনাক্স ফাইলসিস্টেম নিয়ে। আইবিএম-এর এই ডেভেলপার-ওয়ার্কস সাইটটা যে যে কী ভয়ানক ভালো, সে আর বলার না। আর একটা খুব ভালো লিংক জুয়ান স্যান্টোস ফ্লোরিডোর লেখা, জারনালাউজড ফাইল সিস্টেম নিয়ে, http://www.linuxgazette.com/issue55/index.html। নেমসিস-এর (Namesys) নিজের সাইটে, মানে যাদের তৈরি রাইজারএফএস, রাইজারের উপর খুব ভালো একটা ডকুমেন্ট আছে। ঠিক এসজিআই-এ (SGI) যেমন আছে তাদের নিজেদের তৈরি এক্সএফএস নিয়ে। ইচ্ছে ছিল বেশ কিছুটা খুঁটিনাটি নিয়ে এগুলোকে ধরা, শেষে নিজেরই কাপড়চোপড় খুলে যাওয়ার যোগাড়, একটাকে বুঝতে আর একটা, তাকে বুঝতে আর একটা। তার উপর খুব বেশি সময় নেটে থাকতে গেলে নিজেরই অপরাধবোধ হয় একটা, সংসার তো চালাতে হবে। আর লেখাটাও শেষের দিকে চলে আসছে, খুব বেশি লম্বা করার ইচ্ছে নেই, এখানে আমরা তাই শর্টে সারব, জাস্ট এক একটা ফাইলসিস্টেমের সুবিধে অসুবিধে নিয়ে, কী কাজের উপযোগী, ইত্যাদি। একটা কথা মাথায় রাখবেন, নানা ধরনের ফাইলসিস্টেম লাগে নানা ধরনের কাজের জন্যে। এমন একটা ফাইলসিস্টেমও নেই যা সব কাজ সবচেয়ে ভালো করে করতে পারে। সবচেয়ে নিরাপদ এবং সুস্থির ফাইলসিস্টেমেও প্রচুর গন্ডগোল তৈরি হয়, তাদের সামলাতে হয়, তাই সবসময়ই ব্যাকআপ করা উচিত নিজের তথ্যের, নিয়মিত ভাবে।

৩.১।। ইএক্সটিটু

গ্নু-লিনাক্স জন্মানোর প্রায় একদম প্রথমতম ইতিহাস থেকেই চালু ইএক্সটিটু (Ext2)। ইএক্সটিটু এসেছে তার আগেকার এক্সটেন্ডেড ফাইলসিস্টেম ব্যবস্থা থেকে, নামের আত্মীয়তাটা খেয়াল করুন। এক্সটেন্ডেড ফাইল সিস্টেম (Extended-File-System) সিনে এসেছিল এপ্রিল ১৯৯২-এ। কারনেল ভার্শন ০.৯৬সি-র সময়। এরপর বারবার বদলাতে থেকেছে। এই বদলের সূত্র ধরেই এসেছিল ইএক্সটিটু, এবং গ্নু-লিনাক্সের সবচেয়ে জনপ্রিয় ফাইলসিস্টেম হয়ে উঠেছিল একটানা বেশ কয়েকবছর ধরে। জার্নালিং ফাইলসিস্টেম মানে জার্নাল ব্যবহার করে এমন ফাইলসিস্টেম আসার পরই ধীরে ধীরে সিন থেকে সরে যেতে শুরু করল ইএক্সটিটু। জার্নাল ব্যাপারটা নিয়ে বিশদ

আলোচনার লিংক তো আগেই দিয়েছি, এই লেখাটা তার জন্যে খুব একটা উপযুক্ত জায়গাও নয়, এবং আমিও আলোচক হিশেবে আদৌ খুব উপযুক্ত নই, এত বেশি টেকনিকাল সেটা। তবু ব্যাপারটা কী সেটা একটু বলে নিই। একটা ফাইলসিস্টেমের জার্নাল বলতে বোঝায় একটা ছক, একটা কাঠামো। এই ছকটাও রাখা থাকে ডিস্কের ওই পার্টিশনেই, যে পার্টিশনের ফাইলসিস্টেমকে জার্নাল করা হচ্ছে। ছক বা কাঠামোটা হল একধরনের একটা নথি বা লগ। সমস্ত বদলগুলো লিপিবদ্ধ হয় এই লগে। এখানে একটা মজা আছে। বদলের লিপিটা লেখা হয় বদলের পরে নয়, বদলের আগে। বদলগুলো ঘটানোর আগেই বদলগুলোকে তুলে দেওয়া হয় এই জার্নালের লগে। ফাইলসিস্টেম তার নিজের মেটাডেটায় কোনো বদল ঘটানোর আগে সেই বদলের হিশেবনিকেশ এই জার্নালে লিখে রাখে। মেটাডেটা বলতে বোঝায় একটা ফাইলসিস্টেমের আভ্যন্তরীন তথ্যের কাঠামো। মেটাডেটা একটা সেকেন্ড-অর্ডার কাঠামো। ফাইলসিস্টেমে রাখা/লেখা তথ্যের গোটা কাঠামোটার হিশেব নিকেশ তুলে রাখা হয় যে কাঠামোতে। এই কাঠামোটার কাজ এটা নিশ্চিত করা যে তথ্যের এবং তথ্যের সজ্জাটার থোবড়া বিলা হয়ে যায়নি, হুলিয়া বদলে যায়নি, ঘেঁটে যায়নি সেগুলো। এই 'মেটা' কথাটা উত্তরআধুনিক সংস্কৃতিচর্চারও একটা খুব প্রিয় টার্ম। মেটান্যারেটিভ, মেটাটেক্সট — ন্যারেটিভ বিষয়ে ন্যারেটিভ, টেক্সট বিষয়ে টেক্সট। মেটাডেটা মানে ডেটা বিষয়ে ডেটা, তথ্য বিষয়ে তথ্য। ধরুন একটা বইয়ের সূচীপত্র, সে তো নিজেও ডেটা, কিন্তু সে তো ডেটা হয়ে উঠছে কারণ বইয়ের ডেটার কাঠামোটাকে সে হাজির করছে, আবার তার নিজেরও একটা কাঠামো আছে। প্রত্যেক ফাইলসিস্টেমেরই নিজের একটা মেটাডেটা থাকে, এবং মেটাডেটা লেখার একটা নিজস্ব রকম। ফাইলসিস্টেম থেকে ফাইলসিস্টেমের তফাতের একটা বড় জায়গা মেটাডেটার কাঠামোর পার্থক্য। মেটাডেটাকে বদলে যেতে না-দেওয়া — এটা ফাইলসিস্টেমের জরুরি কাজগুলোর মধ্যেও সবচেয়ে জরুরি। মেটাডেটা বদলে গেলেই সিস্টেম আর পার্টিশনের ফাইলসিস্টেমে রাখা তথ্যের সজ্জাকে, কাঠামোকে, তাই তথ্যকে চিনতে পারবেনা। তথ্যগুলো তখন আছে এবং নেই। মানে নেই। ওরকম সন্ধ্যাভাষার অস্পষ্টতা নিয়ে সাইকোলজি কাজ করে, কাব্যসাহিত্যও, হার্ডডিস্কের প্ল্যাটার আর হেড নয়।

জার্নালাইজড ফাইলসিস্টেমের জার্নালের কাজ এটাই, মেটাডেটায় কোনো বদল ঘটানোর আগেই আশু বদলটাকে জার্নালে লিখে রাখা। সিস্টেম বুট করার সময় বা বিদ্যুৎবিভ্রাট ইত্যাদি কোনো গন্ডগোল ঘটলে ফাইলসিস্টেম যে চেক করা হয়, সেই চেকের বা নিরীক্ষার কাজে সময়ের প্রয়োজনটা ভয়ানকভাবে কমিয়ে দেয় জার্নাল। এমনিতে ভাবুন, প্রতিটি সিলিন্ডারের ট্র্যাকের কাঠামো ধরে ধরে তথ্য এবং তথ্যের সজ্জাকে মেলানোর কাজটা কমে গিয়ে দাঁড়াচ্ছে জাস্ট হাতে-গরম একটা জার্নাল মেলানো। গোটা ডিরেক্টরি-সিস্টেম ফাইলসিস্টেম আর মেলাতে হচ্ছেনা। শুধু জার্নালটা আর একবার চালিয়ে দেওয়া হচ্ছে। জার্নালিত ফাইলসিস্টেম আসার পরেই ইএক্সটিটু ফাইলসিস্টেমের জনপ্রিয়তা কমে যেতে শুরু করল। ফাইলসিস্টেম চেকের সময়টাকে এমন অবিশ্বাস্যভাবে কমিয়ে দেয় জার্নাল। কিন্তু এখনো, একটা বিরাট সংখ্যক লিনাক্সীর কাছেই ইএক্সটিটু খুব প্রিয় একটা ফাইলসিস্টেম। এর একটা কারণ দীর্ঘদিনের অভ্যস্ততা আর নির্ভরতা। এত বেশি মেশিনের এত বেশি সিস্টেমে এত বেশি ইউজার ইএক্সটিটুকে ব্যবহার করেছেন যে, এরকম বহুপরীক্ষিত ফাইলসিস্টেম আর হওয়া সম্ভব নয়।

গ্নু-লিনাক্সে, স্বাভাবিক অবস্থায়, সিস্টেম অফ হওয়ার সময়, কারনেল সমস্ত বাফারকে তার নিজের নিজের জায়গায় সঠিকভাবে লিখে দিয়ে পার্টিশন থেকে বেরিয়ে আসে। মানে, পার্টিশনটাকে সিস্টেমের বাইরে বার করে দেয়, বা, টেকনিকাল ভাষায়, আনমাউন্ট করে। কারনেল এক এক করে প্রতিটি পার্টিশন থেকে আনমাউন্ট করে, এবং সব শেষে নিজেই নিজের কাছ থেকে ছুটি নেয়, আমরা স্ক্রিনে ফুটে উঠতে দেখি, 'Sending all processes the TERM signal... Sending all processes the KILL signal...'। মেশিন অফ হয়। বিদ্যুৎবিভ্রাট বা অন্য কোনো কারণে, ভুল ভাবে মেশিন বন্ধ করার কারণে, বা অন্য কোনো গোলযোগে, যখন স্বাভাবিক আনমাউন্ট ঘটেনা, মানে, সঠিকভাবে বাফারগুলোকে লিখে সমস্ত জরুরি তথ্যকে তার নিজের জায়গায় রেখে কারনেল সঙ্গতভাবে বেরিয়ে আসতে পারেনা, সেই না-পারাটা একটা অপরিচ্ছন্নতা তৈরি করে ফাইলসিস্টেমে। টেকনিকাল ভাষায় বললে, ক্লিন আনমাউন্ট হয়নি। তখন পরের বার বুট হওয়ার সময় ফাইলসিস্টেম চেকিং শুরু হয়।

আমরা ফাইলসিস্টেম চেক করার কমান্ড 'fsck'-এর কথা বলেছিলাম। এর অনেকগুলো ভার্শন হয়। প্রতিটি আলাদা আলাদা ফাইলসিস্টেমের জন্যে একটা আলাদা আলাদা 'fsck' সংস্করণ। এমনকি আমরা যখন রুট হয়ে কমান্ড প্রম্পটে '`fsck`' দিয়ে একে চালাই কোনো একটা বিশেষ পার্টিশনের উপর, তখনো সিস্টেম নিজেই দেখে নেয়,

পার্টিশনটায় কী ফাইলসিস্টেম আছে, এবং সেই উপযোগী 'fsck' ভার্শন চালায়। আমার মেশিনের সিস্টেমে ধরুন, '`fsck /dev/hdb3`' কমান্ড দিলে, সিস্টেম নিজেই চালাবে 'fsck.xfs', কারণ এই পার্টিশনে ফাইলসিস্টেম এক্সএফএস, আবার '`fsck /dev/hdb5`' দিলে, চালাবে 'fsck.reiser', কারণ এই পার্টিশনে আছে রাইজার, সাত নম্বর দিনের ৩.৩ নম্বর সেকশনের ছবির সঙ্গে মিলিয়ে নিন।

'fsck' কমান্ডটার কথা ইএক্সটিটু ভার্শনটার নাম 'fsck.ext2'। ইএক্সটিটু ফাইলসিস্টেমের একটা পার্টিশন যখন ক্লিন আনমাউন্ট হয়না, তখন ফের সিস্টেম চালু হওয়ার সময়ে, বা ওই পার্টিশনটা নতুন করে মাউন্ট করতে গেলে তখন 'fsck.ext2' চালু হয়। তার কাজ গোটা মেটাডেটাকে পুনঃসংস্থাপিত করা তার সঠিকতায়। এইটা করতে গিয়ে তাকে গোটা ফাইলসিস্টেমটাই তন্নতন্ন করে খুঁটিয়ে খুঁটিয়ে পরখ করতে হয়, শুধু মেটাডেটার বিটগুলোর সাম্প্রতিকতম বদলটাকে বিশ্লেষণ করলেই কাজ হয়না। জার্নাল থাকলে যা হতে পারত। 'fsck.ext2' বকেয়া থেকে যাওয়া তথ্য বা ডেটা-ব্লকগুলোকে একটা 'lost+found' নামের ডিরেক্টরিতে তুলে রাখে। মোটের উপর লাগে প্রচুর প্রচুর সময়, জার্নালের লগটা চালিয়ে দেখতে যা সময় লাগে তার চেয়ে অনেক বেশি। বড় হার্ডডিস্কে এই কাজে দুতিন ঘন্টাও লেগে যেতে পারে। কিন্তু, খেয়াল করুন, জার্নাল রাখেনা বলেই আবার ইএক্সটিটু-তে মেটাডেটার পরিমাণটা অনেক কম হয়। তাই, মেমরি অনেক কম লাগে এই ফাইলসিস্টেমের ব্যবহারে। সিস্টেমের র‍্যাম যদি খুব কম হয়, ইএক্সটিটু অনেক তাড়াতাড়ি কাজ করতে পারে অন্য অনেক ফাইলসিস্টেমের চেয়ে। প্রকৌশলের নিরবচ্ছিন্ন বৈপ্লবিকায়ন বা কনটিনিউয়াস রিভলিউশনাইজেশন অফ প্রোডাকশনের কল্যাণে, বাজারে টিকে থাকতে গেলেই কোম্পানিগুলোকে যা করে চলতে হয়, র‍্যাম এখন লোকে মেগাবাইটে না-কিনে গিগায় বা হাফ-গিগায় কিনতে শুরু করেছে, এবং ১২০ গিগার হার্ডডিস্ক তো আমাদেরই অনেকে কিনে ফেলেছে এর মধ্যেই, টেরাবাইটের হার্ডডিস্ক এলো বলে, তাই ইএক্সটিটু-র এই জোরের জায়গাটাও নেই হতে চলল বলে।

৩.২।। ইএক্সটিথ্রি

আপগ্রেড করা বা উন্নততর করে তোলাটা কম্পিউটার জগতে, হার্ডওয়ার আর সফটওয়ার দুটোতেই, খুব বড় একটা ইশু। ইএক্সটিটুকে ইএক্সটিথ্রিতে আপগ্রেড করে তোলার কাজটা খুবই সহজ। ইএক্সটিথ্রি (Ext3) আবার জার্নালিত ফাইলসিস্টেম। ইএক্সটিটুর সহজতা এবং জার্নালের দ্রুততা একসঙ্গে পাওয়া যায়। ইএক্সটিথ্রির সঙ্গে অন্যান্য পরবর্তী যুগের জার্নালিত ফাইলসিস্টেমের তফাত এটাই যে ইএক্সটিথ্রি কিন্তু এক্সটেন্ডেড ফাইলসিস্টেমের কাঠামো থেকে বেরিয়ে যায়নি। নামের ধারাবাহিকতাটা দেখুন। ইএক্সটিথ্রি প্রায় সরাসরি ইএক্সটিটু-র উত্তরাধিকার বহন করে। দুটোর আত্মীয়তার জায়গাটা খুবই পোক্ত। আর এই কারণেই একটা ইএক্সটিটু ফাইলসিস্টেমের উপর খুবই সহজে একটা ইএক্সটিথ্রি ফাইলসিস্টেম বানিয়ে তোলা যায়। এদের একমাত্র গুরুত্বপূর্ণ তফাত ওই জার্নাল। তাই, ইএক্সটিথ্রি-র সুবিধেগুলোর একটা, খুব সহজে ইএক্সটিটু থেকে বদলে নিতে পারা। বদলে নেওয়ার সময় ফাইলসিস্টেমের তথ্য এবং তথ্যসজ্জার কোনো গোলমাল ঘটে যাওয়ার হওয়ার সম্ভাবনা থাকেনা।

ইএক্সটিটু-র কোডের উপরেই তৈরি ইএক্সটিথ্রি-র কোড, আর ডিস্কের ভিতর তথ্য রাখার কৌশল বা মেটাডেটার কাঠামো দুটো ফাইলসিস্টেমেই প্রায় হুবহু এক, তাই ইএক্সটিটু-তে কোনো পার্টিশন ফরম্যাট করা থাকলে, এবং পরে সেটা বদলে কোনো জার্নালিত ফাইলব্যবস্থায় যেতে চাইলে, রাইজারএফএস (ReiserFS) এক্সএফএস (XFS) বা জেএফএস (JFS) জাতীয় অন্য কোনো সিস্টেমে যাওয়ার যে ঝামেলা সেটা ইএক্সটিথ্রির বেলায় হয়না। ওগুলোয় যাওয়ার আগে পার্টিশনের গোটা তথ্যের ব্যাকআপ করে নিতে হয়, এবং সেই ব্যাকআপ থেকে ফের শুরু করে গোটাটা বানিয়ে নিতে হয়। সেটা ইএক্সটিথ্রি-তে হয়না। আর ফের গোটাটা বানিয়ে নিতে গিয়ে কিছু ঘাপলা না-ঘটাটাই আশ্চর্য। কোনো একটা লিংক কাজ করছে না, কোনো একটা লাইব্রেরি পাচ্ছেনা, ইত্যাদি। এটা ইএক্সটিথ্রি-র বেলায় হয়না, কারণ আসলে ওই কাঠামোটা তো বদলাচ্ছেই না। আবার ইএক্সটিথ্রি থেকে কখনো ইএক্সটিটু-তে ফেরত যেতে চাইলে সেটাও দিব্য সহজ। একটা পরিচ্ছন্ন রক্তপাতহীন আনমাউন্ট করো ইএক্সটিথ্রি পার্টিশন থেকে, আবার তাকে রিমাউন্ট করো ইএক্সটিটু হিশেবে।

জার্নালিত ফাইলব্যবস্থাগুলোয় তথ্যসততা কায়েম রাখার কায়দাটা হল শুধু মেটাডেটাকে বদলে যেতে না-দেওয়া, ফাইলসিস্টেমের বদলের সঙ্গে সঙ্গত কোনো বদলের বাইরে মেটাডেটার অন্য যে কোনো বদলকে নিয়ন্ত্রণ করা।

মেটাডেটা যেন সবসময়েই অনাঘ্রাত অনাহত থাকে। বরং, এই জার্নালিত ব্যবস্থাগুলোয়, মেটাডেটা যেরকম সুরক্ষিত, মূল ফাইলব্যবস্থার অপরিবর্তনীয়তাটা ততটা বর্মচর্মাবৃত নয়। ইএক্সটিথ্রি-র বেলায় তা নয়, সে তথ্য এবং তথ্য-বিষয়ক-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই নিরাপত্তার দেখভাল করে। কতটা যত্ন নিয়ে দেখভাল করবে সেটাও বলে দেওয়া যায় সিস্টেমকে, মানে, নিজের মত করে বদলে নেওয়া যায়, কাস্টমাইজ করা যায়। ডেটা এবং জার্নালকে তুল্যমূল্য করে দিলে, 'data=journal' মোডে সবচেয়ে বেশি নিরাপত্তা, কিন্তু এতে কাজের গতি খুব কমে যায়। এতে তথ্যের গায়ে কোনো আঁচড় পড়েনা, কারণ তথ্য এবং তথ্যের-তথ্য, ডেটা এবং মেটাডেটা, দুয়েরই জার্নাল রাখা হচ্ছে। বরং 'data=ordered' মোডে ডেটা এবং মেটাডেটা দুয়ের অটুটতাতেও নজর রাখা হয়, কিন্তু জার্নাল রাখা হয় শুধু মেটাডেটার। মেটাডেটার এক একবার বদলের সঙ্গে সম্পৃক্ত যাবতীয় ব্লকগুলোকে সংগ্রহ করে রাখে ফাইলসিস্টেম ড্রাইভার, এক একটা দলে, এক একটা 'ট্রানজাকশন'-এর ('transaction') আকারে। এবং মেটাডেটায় বদল ঘটানোর আগে এক একটা গোটা ট্রানজাকশনকে লিখে ফেলা হয় হার্ডডিস্কের শরীরে। এতে কাজের গতিও ঠিক থাকে, আবার তথ্যের নিরাপত্তাও বজায় থাকে। আর একটা মোড হল 'data=writeback'। এই মোডে মেটাডেটার খাতায় বদলের নথী লিখে ফেলার পর হার্ডডিস্কের মূল ফাইলসিস্টেমে তথ্য লেখা হয়। মোডের নামটা দেখুন, আগে-পরে-টা বুঝতে পারবেন। এই তিন-নম্বর মোডে কাজ সবচেয়ে তাড়াতাড়ি হয়। কিন্তু একটা কোনো বিদ্যুৎবিভ্রাট বা ওই জাতীয় কোনো কেলোর পরে একটা বিদঘুটে ব্যাপার ঘটার সম্ভাবনা থাকে। আপনার কবিতার ফাইলে যেসব লাইন আপনি ইতিমধ্যেই উড়িয়ে দিয়েছেন, সেগুলো ফের পেয়ে যেতে পারেন ফাইলে। কেন ভাবুন তো? হঠাৎ করে ওই অপরিচ্ছন্ন আনমাউন্টে পার্টিশনের আভ্যন্তরীন ফাইলসিস্টেমের তো কোনো হানি হয়নি, সেটা অটুট আছে। এবার, আপনি যে বদল ঘটিয়েছিলেন, সেটা মেটাডেটায় লিখে ফেলার, রাইটব্যাক করার আগেই কেলোটা ঘটায় আপনার বর্জিত লাইনেরা আবার কোরাসে গাইতে শুরু করেছে, লাইনেই ছিলাম বাবা। যদি নিজের থেকে কোনো পছন্দ না দিয়ে দেন, তাহলে ইএক্সটিথ্রি চলে তার ডিফল্ট সেটিং-এ, মানে 'data=ordered', মানে, শুরুতে খানিক মৃণাল-মাণিক অন্তে সুভাষ ঘাই দিয়ে কোনাকুনি বনসাইয়ে সাজানো টোনা-টুনির নিরাপদ সংসার।

৩.৩।। রাইজারএফএস

কারনেল ভার্শন ২.৪ থেকে আনুষ্ঠানিক ভাবে, অফিশিয়ালি, রাইজারকে গ্নু-লিনাক্সে পাওয়া গেল। কিন্তু তার কিছু আগে থেকেই রাইজারএফএস একটা কারনেল প্যাচের আকারে পাওয়া যেত। প্যাচ বলতে একদম তাপ্পি বলতে আমরা যা বুঝি। শুধু নেগেটিভ নয়, পজিটিভ তাপ্পি। মূল প্যান্টুলুনের ফুটো ঢাকতে ব্যবহার করা নেগেটিভ তাপ্পি নয়, একটা বাড়তি পকেট পাওয়ার আশায় পজিটিভ তাপ্পি। প্যাচকে মূল কাঠামোর গায়ে সেঁটে দেওয়া হয় বাড়তি কিছু পাওয়ার আশায়। বাড়তি কোনো কাজ, বাড়তি কোনো সুবিধা। কখনো কখনো অবশ্য কোনো খুচরো খ্যাঁচ সামাল দিতেও ব্যবহার হয় প্যাচ। ল্যারি ওয়াল আর পল এগার্টের এই প্যাচের সফটওয়ার ধারণাটা বেশ ইন্টারেস্টিং। কিন্তু এখানে সেটা আলোচনার জায়গা নয়। কারনেলের মধ্যে যা নেই, কিন্তু বাইরে থেকে যা যোগ করে নেওয়া যেতে পারে, বা, কারনেলের ভার্শন স্থির হয়ে যাওয়ার পরে আর যা কাজ হয়েছে সেগুলো দিয়ে দেওয়া যেতে পারে, এই প্যাচ দিয়ে। একটু '`man patch`' করে নিতে পারেন। প্যাচ বুঝতে গেলে আবার একটু ডিফ (diff) বুঝতে হবে। ডিফ পড়ে দেখুন, দুটো ফাইলের তফাত বা ডিফারেন্স নিয়ে কাজ করে, খুব মজার। দুটো ডিরেক্টরির পার্থক্য বোঝার জন্যেও কাজে লাগানো যায়। যাকগে।

রাইজার যাদের তৈরি সেই কোম্পানির নাম নেমসিস-কে (Namesys) খুব সহজেই নেমেসিস পড়া যায়, কিন্তু গ্রীক পুরাণ মাফিক তথ্যের কোনো সমুহ বিনাশ আদৌ এর উদ্দেশ্য নয়, বরং তার উল্টোটা, রাইজারএফএস এসেছিল পুরোনো ইএক্সটিটু-র একটা শক্তিশালী বিকল্প হিশেবে, তথ্যের শক্তিশালীতর যত্নআত্তির আশায়। সীমিত ডিস্কভূমির আরো কাবেল ব্যবহার, আরো কম জায়গায় মোট আরো তথ্য রাখা, রাইজারের একটা জোরের জায়গা। এবং ক্র্যাশ বা কেলো থেকে দ্রুততর আরোগ্য, যার কথা আমরা আগেই বলেছি। কিন্তু একটা সমস্যার জায়গা এই যে রাইজারের বেশি মনোযোগ ডেটার চেয়েও মেটাডেটার দিকে। পরে হয়ত রাইজার শুধু মেটাডেটা নয়, ডেটারও জার্নাল রাখতে শুরু করবে, ইএক্সটিথ্রির বেলায় যেমন বলছিলাম। ডিস্কভূমির খুব ভালো ব্যবহার হয় রাইজারে। এই কাজে রাইজারএফএস মূলত ব্যবহার করে ব্যালান্সড-ট্রি অ্যালগরিদম। শুধু রাইজারএফএস নয়, এই ব্যালান্সড ট্রি

অ্যালগরিদম ব্যবহার করে এইচএফএস (HFS), এনএসএস (NSS) এবং স্পাইরালগ (Spiralog) ফাইলসিস্টেমও। যদিও রাইজারএফএস ভার্শন ফোর, ব্যালান্সড ট্রি-র জায়গায় ড্যান্সিং ট্রি অ্যালগরিদমের কথা বলেছে। কিন্তু এটা এমন একটা জায়গায় ঢুকে পড়ছি, আমি ঢোকার চেষ্টা করে দেখেছি, অন্তত এখনো আমার সেখানে কথা বলার যোগ্যতা নেই। আমি আপনাদের এই রাইজার, এরপর জেএফএস আর এক্সএফএস-এর একটা হালকা উপরসা আন্দাজ দিয়ে বেরিয়ে যাব। যতটুকু কথা না-জানলে অন্য খুব সাধারণ কাজগুলোও বুঝতে অসুবিধে হয়। কাজ করতে গিয়ে আমি দেখেছি, রাইজারএফএস আর এক্সএফএস অনেক সুবিধাজনক, রাইজার ব্যবহারের পরামর্শ আমায় দিয়েছিল অরিজিৎ, আর এক্সএফএসেরটা দিয়েছিল তথাগত।

এই প্যারাটা একটু বেশি টেকনিকাল। যদি না-বোঝেন, এই পাঠমালার কোনো কোনো বাছাই অংশের মত, মাথা ঘামাবেন না, তাতে মূল কাজের কোনো হেলদোল হবে না। ইএক্সটিটু বা ইএক্সটিথ্রির সঙ্গে রাইজার এক্সএফএস জেএফএসের গতির পার্থক্য এতটাই যে একবার ম্যানড্রেক ৮.২-এ এক্সএফএসে অভ্যস্ত হয়ে যাওয়ার পরে রেডহ্যাটে ইএক্সটিটু বা থ্রিতে, বিশেষ করে কিছু কম্পাইল করার সময়, আমার কান্না পেয়ে যেত। রেডহ্যাট উড়িয়েই দিয়েছিলাম। শেষে ইনস্টল করার সময়ে দ্বিতীয় ভার্চুয়াল কনসোলে গিয়ে 'mkfs.reiserfs' ব্যবহার করে পার্টিশনটা রাইজার বানিয়ে এক নম্বর কনসোলে ফেরত এসে পার্টিশন আর ফরম্যাট না-করে ইনস্টল করে, রাইজারে রেডহ্যাট করেছিলাম, তবে সিস্টেম ভদ্রস্থ হয়েছিল। রেডহ্যাটে এক্সএফএস করার কোনো উপায় আমি পাইনি, তবে ইন্দ্র বলেছিল ওর কাছে কী একটা সিডি আছে যা দিয়ে এক্সএফএসে করা যায়, আমায় দেবে, তা ইন্দ্র যা ব্যস্ত থাকে, ফোন করবে এবং ফোন করল এই দুটোর মধ্যে থাকে একমাসের দূরত্ব, যে আমি সামনের জন্মের আগে ওটা পাওয়ার খুব বেশি আশা করিনা। আমার নিজের অন্তত এক্সএফএস সবচেয়ে পছন্দের, কিন্তু স্ল্যাকওয়্যার নাইনে আমায় এক্সএফএস থেকে রাইজারে ফেরত যেতে হয়েছে, কারণ এক্সএফএস কারনেল যেটা থাকে সেটার সঙ্গে কিছু কিছু কারনেল মডিউলের বেশ সমস্যা হয়েছে। হতে পারে আমার এনভিডিয়া জিফোর্স কার্ডের জন্যে। কারণ, জিফোর্সের ড্রাইভারে দেখেছি কিছু সমস্যা হয়। আমার স্ল্যাকওয়্যার সিস্টেমে এক্সএফএস কারনেলে, এমপ্লেয়ার আর জিএসএল মানে গ্নু সায়েন্টিফিক লাইব্রেরি, সি-তে গণিত আর বিজ্ঞানের কাজ করার একটা লাইব্রেরি, কিছুতেই কম্পাইলড হচ্ছিল না। রাইজারে ফেরত আসার পর আবার সব চমৎকার হয়ে গেছে। দেখি, স্ল্যাকওয়্যার নাইন পয়েন্ট ওয়ানের সিডি ইমেজ ডাউনলোড করছে তথাগত, কালই ফোন করে জানিয়েছে, অনেকটাই হয়ে গেছে, যদি সেটায় ভালো ভাবে এক্সএফএসে করা যায় তাহলে আমার সবচেয়ে পছন্দের ডিস্ট্রো স্ল্যাকওয়ারে সবচেয়ে পছন্দের ফাইলসিস্টেম এক্সএফএসে কাজ করতে পারব। একই সঙ্গে এটাও বলে রাখা ভাল এই পছন্দগুলো কোনো রকমেই সব জেনে বুঝে পছন্দ নয়। হতেই পারে অন্য কোনো কিছু আমার আরো পছন্দ হত একটু বিশেষ রকমে বদলে নিলেই, যাকে টেকনিকাল ভাষায় বলে 'টুইক করা'। সেটার কথা আদৌ জানিই-না, এতকম জানি কম্পিউটারের।

তবে, চারদিকের অবস্থা যা, কালকেই যাদবপুরে অংকুর বাংলা সিডি নিয়ে গ্নু-লিনাক্স সেমিনারের গল্প শুনছিলাম। সেখানে একজন সিনিয়র সিটিজেন কোনো একটা মন্তব্যের সূত্রে বলেছেন, আমি তো 'লিনাক্স এইট' ব্যবহার করি। এবং এই ঘটনাটার উপাখ্যানতার মূল খোঁচটা এইখানে যে লোকটা যাদবপুর কম্পিউটার সায়েন্স ডিপার্টমেন্টের ফ্যাকাল্টি মেম্বার, সেখানে পড়ায়, এবং সে, নাকি, "লিনাক্স নিয়ে কাজ করে"। এটা যদি আমি বলতাম, আমাদের জিএলটি-র নতুন ছেলেদের কেউ বলত, সেটা অবাক লাগত, এমা কিছুই তো জানিস না — 'লিনাক্স এইট' বলেই কিছু হয়না। কিন্তু এই পরিস্থিতিতে এটা একটা অপদার্থ ধৃষ্টতা, একটা স্পর্ধা, একটা জাতিসত্তার ধর্ষণ। ইংরেজরা আমাদের যেমন চাইত তেমনি হয়ে থাকা। চারপাশটাই এইরকমই — একটা শক্তির কেন্দ্র, তার চারদিকে ছোট ছোট বোকা বোকা পকেট-সাইজ জমিদাররা। সমস্ত বিশ্বাস মূল্যবোধ সবকিছুই, ঠিক ওই লোকটার লিনাক্সে কাজের মতই, একটা জনশ্রুতি কিম্বদন্তী হিয়ারসে — 'নাকি' কাজ করে। অ্যাকাডেমির ওই সিনিয়র সিটিজেনদের মহিমাময় হয়ে বিরাজ করার কল্যানেই আমার পাড়ার ছোট ছোট ছেলেদের মধ্যে কম্পিউটার নিয়ে মহিমার ত্রাসের এলিটতার আবহ বানানো হয়, বয়ে নিয়ে যাওয়া হয়। সবকিছুই নিজে করার, নিজে করে মিলিয়ে নেওয়ার উপায় থাকে বলেই, গ্নু-লিনাক্স আসলে এই আবহটার উপরেই আক্রমণ। এই গোটা লেখাটায় যা করে চলার চেষ্টা করছি, পরিশ্রম করো, শেখো, সবকিছু তোমার জন্যে দেওয়া আছে গ্নু-লিনাক্সে। এমনকি আমিও, কম্পিউটারের বাইরের লোক হয়ে, যদি নিজের চেষ্টায় দুবছরেরও কমে এই লেখাটা লেখার মত শিখে উঠতে পারি, তাহলে তুইও পারবি। এটা আমি রোজ

অশেষকে পিউকে কেয়াকে বলি। সম্পদটা সামনেই আছে, দ্যাখ, শেখ। ওই মৌরসি পাট্টার বাইরে, ওদের নাকি-পাণ্ডিত্যের জনশ্রুতির বাইরে, নিজে হাঁট। নিজে দেখে শেখার চেষ্টা কর, যা যা তোর কাছে আসছে তার সবই এক একটা প্রচার জনশ্রুতি রূপকথা, মূল্যবোধ মনন অবস্থানও এখন একটা জনশ্রুতি, একটা 'নাকি', নিজে হাঁট — যদি তোর ডাক শুনে কেউ না আসে, তবে 'নাকি' একলা চলতে হয় ... (কার্টসি নচিকেতা)

হ্যাঁ, রাইজার। রাইজারএফএস-এ গোটা তথ্যটাই সজ্জিত থাকে একটা কাঠামোয় যার নাম 'বি-ব্যালান্সড ট্রি'। এই ট্রি বা গাছের আকারে সজ্জার কল্যাণেই খুব চমৎকার ভাবে ডিস্কভূমিটাকে ব্যবহার করতে পারে রাইজারএফএস। খুব ছোট ছোট ফাইলগুলোকে সরাসরি ওই গাছের পাতায় পাতায় রাখতে রাখতে যায়, শিকড় গুঁড়ি ডালপালা এরকম কোথাও না-রেখে — শুধুমাত্র বাস্তব ডিস্কের বাস্তব শরীরে তার অবস্থানের পয়েন্টারটুকু লিখে রেখে। এই লিখে রাখাটাও এক কেবি বা চার কেবির এক একটা খণ্ডে ঘটেনা, ঘটে ঠিক যতটা আয়তন প্রয়োজন ততটা আয়তনের এক একটা অংশে। ফিজিকাল আর লজিকাল ব্লকের প্রসঙ্গে ডিস্কজমি নষ্ট হওয়ার সেই আলোচনাগুলো মনে করুন। রাইজারএফএসের আর একটা ভালো জায়গা, আইনোডগুলোকে গতিশীল বা ডায়নামিক রকমে ধার্য করার প্রক্রিয়া। ইএক্সটিটু জাতীয় পুরোনো ফাইলসিস্টেমগুলোর চেয়ে এটা ভালো, যেখানে ফাইলসিস্টেম বানানোর সময়েই স্থির করা থাকে আইনোডদের ঘনত্ব কতটা অব্দি হতে পারে। রাইজারে, খুব ছোট ছোট ফাইলের বেলায়, ফাইল আর ফাইলের 'stat_data' বা আইনোড তথ্য পাশাপাশি রাখা থাকে, যাতে মাত্র একবার ডিস্কে পড়ে বা লিখেই ফাইল সংক্রান্ত গোটা কাজটা সিস্টেম করে নিতে পারে। এতে ডিস্ক অনেক দ্রুত কাজ করে। আর জার্নালের কল্যাণে কোনো কেলো বা ক্র্যাশ থেকে ফেরত আসতে রাইজারের সময় লাগে মাত্র কয়েক সেকেন্ড। এমনকি গাবদা গাবদা সাইজের ফাইলসিস্টেমেও। মেশিনে আমি এত কেলোপরায়ণ রকমে কাজ করি যে, টেক ইট ফ্রম মি, সত্যিই কয়েক সেকেন্ড।

৩.৪।। জেএফএস

জেএফএস আমি নিজে খুব সামান্য সময়ের জন্যেই ব্যবহার করেছি। ম্যানড্রেক নাইন ইনস্টল করেছিলাম একটা জেএফএস পার্টিশনে। সেটা খুব দ্রুতই উড়িয়ে তার জায়গায় স্ল্যাকওয়ার করে নিতে হয়েছিল, তার কারণ ফের সেই এনভিডিয়া। এনভিডিয়া জিফোর্স ড্রাইভারটার সঙ্গে, আমার সিস্টেমে ম্যানড্রেক কিছুতেই সুস্থিত হচ্ছিল না। তাই নিজে খুব ভালো বলতে পারব না, কিন্তু লাগের একাধিক জন বলেছে, তারা জেএফএস ব্যবহার করে খুবই খুশি। ও, একটা কথা বলতে ভুলে গেছি, রাইজার, ইএক্সটিথ্রি, জেএফএস, এক্সএফএস এগুলো সবই জিপিএল বা গ্নু-পাবলিক-লাইসেন্সের আওতায় পড়ে। চার নম্বর দিনে, ইউনিক্স থেকে গ্নু-লিনাক্সে আসার প্রসঙ্গে আলোচনায় আমরা জিপিএল-এর কথা বলেছিলাম, সফটওয়ার নিয়ে মনোপলি ব্যবসার বিরুদ্ধে লড়াইয়ে রিচার্ড স্টলম্যানের সেই ব্রিলিয়ান্ট বজ্জাতি। জার্নালাইজড ফাইলসিস্টেম নিয়ে আর একটা ভালো রেফারেন্সও মনে পড়ল, বার্নার্ড কান-এর একটা চমৎকার প্রবন্ধ, জেএফএস-এর সঙ্গে অন্য জার্নালিত ফাইলসিস্টেমগুলোর তুলনামূলক আলোচনা পাবেন। নেটে গিয়ে 'Bernard Kuhn' দিয়ে সার্চ দিন, পেয়ে যাবেন।

'জার্নাল' শব্দটা জেএফএস-এর নামেই ঘোষিত — জার্নালিং ফাইল সিস্টেম। এটা এনেছিল আইবিএম। গ্নু-লিনাক্সে জেএফএস এসেছে খুব অল্পদিন, দুহাজার সালে বিটা ভার্শন বা প্রস্তুতিকালীন সংস্করণ, আর ২০০১-এ এসে প্রথম পূর্ণাঙ্গ ভার্শন। মূলত খুব বেশি কাজের চাপে থাকা সার্ভার মেশিনগুলোর কথা ভেবেই জেএফএস তৈরি, কাজ কত দ্রুত করা যায় এটাই যেখানে প্রায় একমাত্র বিবেচ্য। জেএফএস পুরোপুরি একটা ৬৪-বিট ফাইলসিস্টেম, তাই খুব বড় বড় সাইজের ফাইল এবং পার্টিশন বানানো জেএফএসে কোনো সমস্যাই নয়। আমরা এই সেকশনের শেষে বড় ফাইলের আয়তনের কথায় আসছি। সার্ভার সিস্টেমের বেলায় জেএফএস খুবই কার্যকরী।

রাইজারের মত জেএফএস ব্যবস্থাতেও জার্নাল রাখা হয় শুধু মেটাডেটার। খুব বিশদ সর্বব্যাপী চেকিং-এর জায়গায় চেক করা হয় ফাইলসিস্টেমের সাম্প্রতিকতম বদলের কারণে মেটাডেটার শরীরে ঘটে যাওয়া পরিবর্তনগুলোকেই। এতে ক্র্যাশ থেকে প্রত্যাবর্তনের প্রচুর সময় বেঁচে যায়। তথ্য লেখা এবং পড়ার কাজ, এবং তার সঙ্গে সঙ্গে লগ বা নথীর বদলের কাজ একই সঙ্গে একাধিক জায়গায় ঘটতে থাকে জেএফএসে, এক একটা গুচ্ছে বা দলে বা গ্রুপে। আলাদা আলাদা ভাবে লেখার কাজের জায়গায় এই সহঘটমান বা কনকারেন্ট কাজের প্রথা জেএফএসের দ্রুততার একটা বড় কারণ। জেএফএসে দু ধরনের ডিরেক্টরি কাঠামো একই সঙ্গে থাকে। ছোট ছোট ডিরেক্টরির জন্যে

ডিরেক্টরির মধ্যেকার মালপত্তর সরাসরি আইনোডে রাখা হয়। আর বড় বড় ডিরেক্টরির বেলায় বি-ট্রি ব্যবহার করা হয়। এতে ডিরেক্টরি সংগঠন খুব ভালো ভাবে হতে পারে। এবং আইনোড ধার্য করার প্রক্রিয়াটা জেএফএস ব্যবস্থাতেও গতিশীল বা ডায়নামিক। আমরা আগেই বলেছি ইএক্সটিটু ফাইলব্যবস্থায় এই আইনোড ঘনত্বটা আগে থেকেই স্থির-করা। তাই, ফাইলসিস্টেমে কতগুলো ডিরেক্টরি থাকতে পারবে, বা ডিরেক্টরিতে কতগুলো ফাইল থাকতে পারবে তার কিছু বিধিনিষেধ এসে পড়ে। জেএফএস ব্যবস্থায় এইসব ঝামেলা নেই, এখানে ডিরেক্টরি বা ফাইলদের ভূমি বণ্টন করা হয় গতিশীল রকমে। আর যেই কোনো একটা ফাইলের বা ডিরেক্টরির কোনো একটা জমির প্রয়োজন ফুরিয়ে যায় অমনি সেই জমিটাকে ফের বণ্টনের জন্যে তুলে রাখে।

৩.৫।। এক্সএফএস

ইউনিক্সের বিভিন্ন ফ্লেভারের কথা বলতে গিয়ে আমরা আইরিক্সের (Irix) কথা উল্লেখ করেছি, চার এবং পাঁচ নম্বর দিনের আলোচনায়। এসজিআই বা সিলিকন গ্রাফিক্স এই এক্সএফএস ফাইলসিস্টেম প্রথম তৈরি করে এই আইরিক্স অপারেটিং সিস্টেমের জন্যে, ১৯৯০-এ। খুব দ্রুত কাজ করতে পারে এমন একটা ৬৪-বিট ফাইলসিস্টেম বানানোর উদ্দেশ্যে। কম্পিউটারের তথা মেশিনের কাছে সিস্টেমের চাহিদা শুধুই বাড়ছিল। বড় বড় ফাইল নিয়ে কাজ করার ব্যাপারে এক্সএফএস খুবই দড়। পরে এক্সএফএসকে গ্নু-লিনাক্সে পোর্ট করা হয়। তবে ঠিক রাইজারএফএস বা জেএফএস-এর মতই এক্সএফএস-ও বেশি মাথা ঘামায় ডেটার চেয়েও মেটাডেটা নিয়ে। উন্নত হার্ডওয়ারে এক্সএফএস বোধহয় অন্য ফাইলসিস্টেমের চেয়ে দ্রুততর কাজ করতে পারে।

এক্সএফএস ফাইলসিস্টেম তৈরির সময়েই, যে ব্লক-ডিভাইসের উপর ফাইলসিস্টেমটা তৈরি হচ্ছে, সেটাকে আটটা বা তার বেশি সমান মাপের একরৈখিক বা লিনিয়ার অঞ্চলে ভেঙে ফেলা হয়। এদেরকে ডাকা হয় বণ্টন-দল বা অ্যালোকেশন গ্রুপ বলে। প্রত্যেকটা অ্যালোকেশন গ্রুপের থাকে নিজের নিজের আইনোড এবং মুক্ত ডিস্ক ভূমি। এর মালিক ওই অ্যালোকেশন গ্রুপ। অ্যালোকেশন গ্রুপগুলো হল ফাইলসিস্টেমের মধ্যে ছোট ছোট ছানা ফাইলসিস্টেম। প্রত্যেকটা অ্যালোকেশন গ্রুপ অন্য অ্যালোকেশন গ্রুপদের থেকে স্বতন্ত্র এবং স্বাধীন। তাই কারনেল একই সাথে একাধিক অ্যালোকেশন গ্রুপের সঙ্গে কথোপকথনে আসতে পারে। জেএফএসের কনকারেন্ট ব্যবস্থার কথা মনে করুন। এই অ্যালোকেশন গ্রুপ হল এক্সএফএস ফাইলব্যবস্থার একটা জোরের জায়গা। কিন্তু, স্বাভাবিক ভাবেই, একাধিক প্রসেসর সম্পন্ন কম্পিউটারে, সিমেট্রিক-মাল্টি-প্রসেসিং গোছের, সবচেয়ে ভালোভাবে কাজ করতে পারে এই অ্যালোকেশন গ্রুপের ধারণাটা। এসএমপি-র কাজের সঙ্গে খুব ভালো যায় অ্যালোকেশন গ্রুপের এই ব্যবস্থাটা।

ডিস্কভূমিটাকে নিয়ন্ত্রণের কাজটা এক্সএফএস খুব ভালো ভাবে করে। এক্সএফএসের কাজের দ্রুততার একটা কারণ ওই বি-ট্রি। অন্য ফাইলসিস্টেম থেকে সম্পূর্ণ স্বতন্ত্র এক্সএফএস-এর একটা বৈশিষ্ট্য হল 'ঢিলে বণ্টন' বা ডিলেড অ্যালোকেশন ('delayed allocation')। ডিস্কভূমিতে তথ্য বণ্টনের গোটা কাজটাকে এক্সএফএস দুটো ভাগে ভাগ করে নেয়। স্থগিত-রাখা বা দেরি-করিয়ে-দেওয়া বা ডিলেড কাজটাকে রেখে দেওয়া হয় র‍্যামে, আর তার জন্যে চিহ্নিত করে রাখা হয় প্রয়োজনীয় ভূমির পরিমাপ। কিন্তু সত্যিকারের বণ্টনটা করার জন্যে এক্সএফএস একদম শেষ মুহূর্ত অব্দি অপেক্ষা করে। এই ধরে রাখা তথ্যের মধ্যে অনেক সময় অনেক অস্থায়ী অংশ থাকে, অনেক তথ্যের চরিত্রই তাই হয়। খুব ছোট্ট একটা সময়ের জন্যে জন্মায় তারা, তার পরেই মরে যায়। সিস্টেমের তাদের লিখে ফেলার আর কোনো দরকার পড়েনা। এই ঢিলে-বণ্টন ব্যবস্থায় সেই তথ্য আর কোনোদিনই ডিস্কের গায়ে লেখা হয়না। কারণ, যতক্ষণে এক্সএফএস গোটা তথ্যটাকে লিখে ফেলার সিদ্ধান্ত নেয় তার মধ্যে সে মরে গেছে। খেয়াল করুন, এই প্রক্রিয়ার কারণে মোট লেখালেখির পরিমাণ কমে যাচ্ছে। এটা এক্সএফএস ফাইলব্যবস্থার দ্রুততার একটা কারণ। আর ফাইলব্যবস্থার মধ্যে ফাইলগুলোর যে টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশনের কথা বলেছিলাম আমরা, সেই ফ্র্যাগমেন্টেশনের পরিমাণও এতে কমে যাবে। এই ব্যবস্থায় সিস্টেমকে অনেক কম সংখ্যক বার বাস্তব ফাইল বাস্তব হার্ডডিস্কে লিখতে হচ্ছে। কিন্তু এই লেখালেখির প্রক্রিয়ার মাঝপথে সিস্টেমে যদি কোনো ঝাড় নামে, কোনো ক্র্যাশ, তাহলে অবভিয়াসলি তথ্য হারানোর পরিমাণও অন্য ফাইলসিস্টেমের চেয়ে বেশি হবে। এই টুকরো হয়ে যাওয়া বা ফ্র্যাগমেন্টেশন এড়ানোর আর একটা কায়দাও আছে এক্সএফএস-এর। যার নাম প্রাকবণ্টন বা প্রিঅ্যালোকেশন ('preallocation')। ফাইলটাকে ডিস্কের ফাইলসিস্টেমে লেখার আগে এক্সএফএস ফাইলটার সঙ্গে

মানানসই কিছুটা ডিস্কভূমিকে রিজার্ভে সরিয়ে রাখে। তাই পরে যখন লেখে, স্বাভাবিকভাবেই, ফাইলটার টুকরো হওয়ার সম্ভাবনা অনেক কমে যায়। আর একটা ফাইলের মোট তথ্য যেহেতু একটা গোটা ডিস্কের নানা জায়গায় ছড়িয়ে থাকেনা, ফাইলসিস্টেম কাজও করতে পারে অনেক তাড়াতাড়ি।

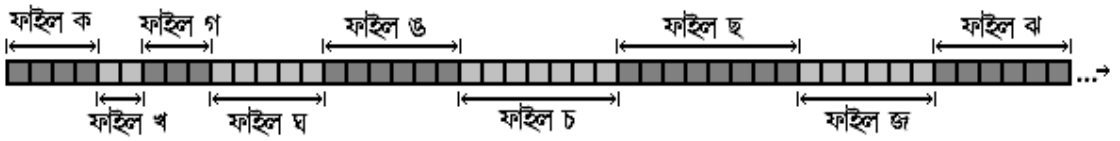
৩.৬।। আইএসও-৯৬৬০ (iso9660)

সিডি পোড়ানোর প্রচুর সফটওয়ার গ্নু-লিনাক্সে এমনিতেই দেওয়া থাকে। ছবি-ইঁদুর দিয়েও সিডি পোড়ানোর রাশি রাশি সফটওয়ার আছে। লাগ-এর ইন্দ্র জিএলটির মিটিং-এ একটা মজার কথা বলেছিল, গ্নু-লিনাক্সে প্রবলেমটা স্কেয়ারসিটি বা সফটওয়ারের অভাব নয়। প্রবলেম অফ প্লেনটি। প্রাচুর্যের সমস্যা। আসলে চারে এবং পাঁচে দেওয়া গ্নু-লিনাক্সের জন্ম এবং বিবর্তনের ইতিহাসটা খেয়াল করুন, তাহলেই বুঝতে পারবেন। ধরুন আপনি প্রোগ্রাম করতে জানেন, আপনি দেখছেন আপনার কোনো একটা কাজ করতে একটা সমস্যা হচ্ছে, আপনি চালু সফটওয়ারটাকে একটু বদলে নিতে চাইলেন। এই কাজের প্রোগ্রামের যতগুলো সফটওয়ার পাওয়া যায়, তাদের সবাইকে আপনি নেড়ে ঘেঁটে দেখলেন। গ্নু-লিনাক্স জগতের বাইরে এটা অসম্ভব। কারণ, আপনি তো সোর্স কোড পাবেন না। আপনি প্রোগ্রামটা ব্যবহার করতে পারেন, জানতে বা বুঝতে পারেন না। এবার আপনার প্রয়োজন মোতাবেক সেটাকে মানিয়ে নিতে গিয়ে তৈরি হয়ে গেল একটা আদ্যোপান্ত নতুন প্রোগ্রাম। এবং স্টলম্যানের জিপিএলের খচড়ামিটা মনে করুন, আপনি জিপিএল-এর আওতায় আসা সফটওয়ার ব্যবহার করেছেন মানেই আপনার সফটওয়্যারও জিপিএলে এসে গেল, এটাও একইরকম উন্মুক্ত। এবং আপনি লিনাক্সী মানে আপনিও তো তাই চান। শুধু মাত্র নিজে পয়সা পিটে গাড়ি বাড়ি করলেন, আপনার নিজের পরবর্তী প্রজন্মও জানতে বুঝতে পারলনা, বিদ্যা থেমে গেল, এটা চাইলে তো আপনি গ্নু-লিনাক্সে আসতেনই না। এই প্রোগ্রাম বানাতে অন্য নানা প্রোগ্রামের চমৎকারিত্বটা পাবেন এরিক রেমন্ডস-এর 'দি ক্যাথিড্রাল অ্যান্ড দি বাজার' লেখাটায় (http://www.tuxedo.org/~esr/)। এভাবেই গ্নু-লিনাক্সে যে কোনো কাজের জন্যে রাশি রাশি প্রোগ্রাম তৈরি হয়ে গেছে।

ইন্দ্র ঠিকই বলেছে, একজন নতুন ব্যবহারকারীর সত্যিই মাথা খারাপ হয়ে যায়। তাও তো, এই লেখাটায়, আগেই ঠিক করে নিয়েছিলাম, কোনো গুই প্রোগ্রামকে আনব না। কমান্ড-প্রম্পটে সিডি পোড়ানোর যে প্রোগ্রাম আছে, তার মধ্যে প্রমুখতম দুটো হল সিডিআরডিএও (cdrdao) এবং সিডিরেকর্ড (cdrecord)। দুটোই বেড়ে জিনিষ। এই দুটোই হল, একটা সিডি রয়েছে, বা তার ইমেজ রয়েছে, সেটাকে একটা নতুন সিডিতে পোড়ানোর। কিন্তু ধরুন আপনার কিছু ডিরেক্টরি আর ফাইল আপনি সিডিতে তুলতে চান, সেই অবস্থায় আপনার সেই ডিরেক্টরি এবং ফাইলগুলো থেকে একটা ইমেজ বা প্রতিচিত্র বানাতে হবে। ইমেজটা আসলে কী করবে, ফাইলব্যবস্থা নিয়ে এতটা আলোচনার পর, এখন বোধহয় আপনি সেটার একটা আন্দাজ করতে পারছেন। হার্ডডিস্কে থাকার সময়ে তার যে ফাইলব্যবস্থা ছিল সেটাকে বদলে সিডিতে থাকার ফাইলব্যবস্থায় নিয়ে আসা, এটারই টেকনিকাল নাম ইমেজ বানানো। ইমেজ বানানোর সফটওয়ারও দেওয়া থাকে গ্নু-লিনাক্স ডিস্ট্রিবিউশনগুলোতেই, কমান্ড মোডে করার প্রোগ্রাম, মেকআইএসওএফএস (mkisofs) এবং মেকহাইব্রিডএফএস (mkhybridfs)। গুই-তেও নিশ্চয়ই আছে, অনেকই আছে, গ্নু-লিনাক্সে সবই অনেক থাকে, কিন্তু আমি জানিনা, কোনোদিন ব্যবহার করিনি। 'mkisofs' এবং 'mkhybridfs' নামদুটোকে দেখুন তো ভেঙে বুঝতে পারছেন কিনা। এদুটোর নিজের ডকুমেন্টেশন, মূলত ম্যানপেজ, পড়ে ফেললেই সিডি বিষয়ে বাংলা বাজারে নিজেকে একজন ধূরন্ধর বলে আপনি চালাতে পারবেন। ঠিক যেটা হয় এমপ্লেয়ারের (mplayer) ডকুমেন্টেশন পড়ে। বিভিন্ন ধরনের মাল্টিমিডিয়া ফাইল, তার ফরম্যাট, তার কাজের তফাৎ, একদম মাল্টিমিডিয়ার হদ্দমুদ্দ — যা বেবি, দৌড়ে যাবি, শিখবি মাল্টিমিডিয়া, নমস্তস্যৈ, নমস্তস্যৈ, নমস্তস্যৈ, নমোনমঃ। ম্যানপেজগুলো ব্রাউজ করে একবার দেখে নিন, আইএসও নিয়ে ওরা কী বলেছে। আর একটা ভারি চমৎকার হাউ-টু আছে। সিডি-বার্নিং হাউদুটো। হাউটুর মূল সূচীতে 'cd' দিয়ে সার্চ দিন, পেয়ে যাবেন।

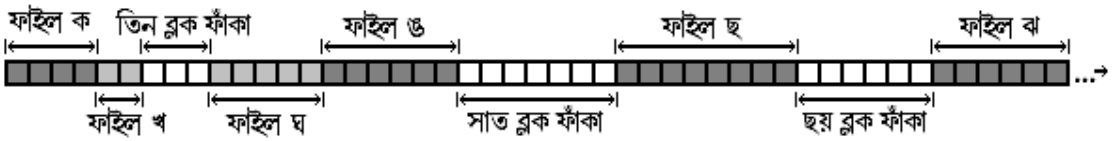
সিডিরমের ফাইলব্যবস্থা নিয়ে একটু বড় করে এখানে বলতে হচ্ছে, তার কারণ, খেয়াল করুন, যে মাধ্যমটায় এই ফাইলব্যবস্থাটা তৈরি হচ্ছে সেটা একদম আলাদা। আমরা হার্ডডিস্কের চৌম্বক মাধ্যম নিয়ে অনেক কথা বলেছি। সিডির সেই অপ্টিকাল ভৌত মাধ্যমের আলোচনায় আর গেলাম না। মাধ্যম বদলে যাওয়ায় ফাইলব্যবস্থাটাও তো আমূল বদলে যাবেই, অন্যরকম হয়ে যাবেই। এবং মজার কথা কী বলুন তো, পুরোনো দিনের হার্ডডিস্কেও একসময়

এই জাতীয় ফাইলব্যবস্থাই ব্যবহার হত, যা সিডিরমের বেলায় এখন হয়। এর আগেও এসেছে প্রসঙ্গটা, এবার আমরা সেটাকে একটু ভালো করে বুঝব, পরপর পরস্পর-সন্নিহিত বা কনটিগুয়াস ফাইলসিস্টেম। একটা জায়গায় এই কনটিগুয়াস ফাইলসিস্টেম কিন্তু ভারি উৎকৃষ্ট জিনিষ, ফাইল টুকরো হয়ে যাওয়ার বা ফ্র্যাগমেন্টেশনের কোনো সমস্যাই এতে ছিলনা। কিন্তু কাজের ঝামেলার জন্যে সেটা বদলে যায়। পরে যা ফের ফেরত এল সিডি এবং ডিভিডি লেখার ব্যবস্থায়। কারণ এখানে ফাইল বা ডিরেক্টরিগুলো লেখার পরে আর বদলাচ্ছেনা। বারবার পোড়ানোর, মানে রিরাইটেবল সিডির বেলায় আপনি যে সিডিতে থাকা একটা ফাইলসিস্টেমে নতুন কোনো ডিরেক্টরি বা ফাইল নতুন করে পুড়িয়ে ঢোকাচ্ছেন, তা কিন্তু নয়। সেখানে আপনি আগের বার পোড়ানো ইমেজ বা সিডির ফাইলসিস্টেমকে উড়িয়ে নতুন একটা ইমেজ বা ফাইলসিস্টেম পোড়াচ্ছেন, মানে রিরাইট করছেন। বা, মাল্টিসেশন বা বহুব্যবহার রকমে হলে, সিডিতে পুড়িয়ে রাখা একটা আংশিক ইমেজ বা ফাইলসিস্টেমের সঙ্গে আর একটা আংশিক ইমেজ বা ফাইলসিস্টেম পুড়িয়ে যোগ করে দিচ্ছেন। অর্থাৎ, মূল জায়গাটা খেয়াল করুন, ফাইলসিস্টেমের সাপেক্ষে একটা ডিরেক্টরি বা ফাইল আর বদলাচ্ছে না। সিডিতে লিখে রাখা অবস্থায় লিখে রাখা অবস্থায় আপনি ফাইল বা ডিরেক্টরি আর বদলাতে পারছেন না। হার্ডডিস্কে বদলে নেওয়া একটা ফাইল নতুন করে লিখে দিতে পারছেন। এমনকি উইন্ডোজের নেরোর সঙ্গে দেওয়া 'ইন-সিডি' গোছের প্যাকেজ, যারা আপনাকে সুযোগ দেয় একটা রিরাইটেবল সিডিকে ফ্লপির মত করে ব্যবহার করার, তারাও আসলে, খুব ছোট্ট ছোট্ট ইমেজে কুচো কুচো ফাইলসিস্টেমকে মুছে



তার জায়গায় নতুন একটা কুচো ইমেজকে লিখে দেয়। দেখে আপনার মনে হয় যেন ফাইলটাই বদলাল।

এই ছবিতে দেখুন, একটা কনটিগুয়াস ফাইলব্যবস্থায় ফাইলরা কী ভাবে থাকে। ফাইলব্যবস্থাটার শুরু থেকে শুরু করে ছবিটায় মোট ৪৭-টা ব্লক আমরা দেখিয়েছি। পরপর একরৈখিক বা লিনিয়ার সজ্জায়। ফাইল ক থেকে ফাইল ঝ। ফাইলব্যবস্থাটা কিন্তু এর পরেও চলছে, এখানেই শেষ নয়। আমরা তার থেকে এখানে কয়েকটা মাত্র ফাইলকে দেখালাম। ফাইল ক, খ, গ, ঘ, ঙ, চ, ছ, জ এবং ঝ, পরপর যথাক্রমে ৪, ২, ৩, ৫, ৬, ৭, ৮, ৬, এবং ৬ ব্লকের। আদতে এরা যে কোনো মাপেরই হতে পারত। ফাইলদের এই পরপর কনটিগুয়াস রকমে থাকাটা বেশ সহজ এবং আরামদায়ক একটা ব্যবস্থা। সিস্টেম যদি একটা ফাইলের মোট পরিমাপ জানতে বা মনে রাখতে চায়, তাকে ব্যবহার করতে হচ্ছে জাস্ট দুটো সংখ্যা — এক, প্রথম ব্লকের নম্বর, এবং, দুই, ফাইলটায় কটা ব্লক আছে। এই দুটো সংখ্যাকে পেলে সিস্টেম গোটা ফাইলটাকেই পেয়ে যাচ্ছে কনটিগুয়াস ফাইলসিস্টেমের শরীর থেকে। ব্লকগুলোর নম্বর যদি ০ থেকে শুরু করি, তাহলে ফাইল ক-এর জন্যে এই সংখ্যা দুটো হচ্ছে ০ আর ৪। মানে, পড়া শুরু করো ০ নম্বর ব্লক থেকে, পড়াটা চালিয়ে যাও ৪ ব্লক ধরে। তাহলেই ক ফাইলের গোটা তথ্যটা সিস্টেমের মেমরিতে উঠে এল। ফাইল খ-এর বেলায় এই সংখ্যা দুটো দাঁড়াচ্ছে ৫ আর ২। ফাইল গ মানে ৭ আর ৩। ফাইল ঘ মানে ১০ আর ৫। ইত্যাদি। অর্থাৎ, এই ব্যবস্থার আর একটা সুবিধে, ফাইল পড়তে সিস্টেমের সময় খরচ কম, একটা ফাইল এক বারে গোটাটা পড়ে নাও। একটা ফাইল সিস্টেমকে খুঁজতে হচ্ছে মাত্র একবার। শুরুর ব্লকটা বার করো, তারপর পড়ে যাও। অর্থাৎ, কনটিগুয়াস ব্যবস্থাটা সহজ এবং দ্রুত। তাহলে বাতিল হল কেন? গন্ডগোলটা পাকতে শুরু করল সময়ের সঙ্গে সঙ্গে। আপনি যদি আপনার মেশিনটাকে সচন্দন সবেলপাতা তুলে রাখেন, যেমন অনেকেরই মেশিনই দেখি, ঢাকনা লাগানো থাকে, যেন ওটা কাজের যন্ত্রপাতি নয়, একটা সযত্নে তুলে রাখার গয়না, তাহলে কোনো ঘাপলা নেই। আপনার ফাইলব্যবস্থাটা চিরকাল একইরকম রইল, কিন্তু মেশিন ব্যবহার করলেই গেলেন।



এই ছবিটা দেখুন। আপনি মেশিনটা ব্যবহার করছেন। ব্যবহার করা মানেই কিছু পুরোনো ফাইল ওড়ানো, কিছু নতুন ফাইল লেখা। এই ছবিতে আমরা ধরে নিয়েছি, ফাইল গ, ফাইল চ এবং ফাইল জ ইহজগতে আর নেই। তাদের আত্মার শান্তিকামনা করে আপনি আরো আরো ফাইল লিখে জীবনে কিছু ফাইল রেখে যাওয়ার দিকে এগোচ্ছেন। এই ফাইল গ চ এবং জ তাদের ডিস্কপাতায় চৌম্বকজলের মত অস্থায়ী জীবন দিয়ে কিছুই যে রেখে যায়নি তা কিন্তু নয়। তারা প্রত্যেকেই ডিস্কের পাতায় এক একটা ফুটো রেখে গেছে। ঠিক যত যত ব্লকের উপস্থিতি ছিল তাদের প্রত্যেকের, সেই সেই মাপের এক একটা ফুটো রেখে গেছে তারা ডিস্কভূমিতে। ডিস্কের প্রথম সাতচল্লিশটা ব্লক জুড়ে এখন আছে ছটা ফাইল এবং তিনটে ফুটো। ফুটো তিনটের সাইজ ৩, ৭ এবং ৬ ব্লক। এখনো আপনার সমূহ বেদনার দিন শুরু হয়নি, আরো আসিতেছে।

ফাইলব্যবস্থার আরো পরের আরো পরের দিকে, মানে আমাদের ছবির অদৃশ্য ডানদিকে আপনি ফাইল লিখে এবং লিখে চলেছেন। তাদের কিছু কিছু ওড়াচ্ছেনও নিজের খুশিমত। সেই বাবু আর কোথায় যারা পায়রা ওড়াত এবং বাজি পোড়াত, আমরা শুধু ফাইল ওড়াই এবং সিডি পোড়াই। একসময় আসবে যখন এইভাবে হরেক সাইজের ফুটোয় আর ফাইলে আপনার গোটা ডিস্কটা ভরে যাবে। এবার? এই অবস্থায় ভাবুন, সিস্টেম যদি হিশেব রাখতে চায়, আর ফাইল লেখার মত কতটা জায়গা আর ডিস্কে আছে, তাকে হিশেব রাখতে হচ্ছে ফুটো গুনে এবং ফুটোর সাইজ গুনে। কোন কোন ফুটো ভরিয়ে দেওয়া যাবে ঠিক কী কী সাইজের ফাইল দিয়ে। কোনো ফাইল লেখার আগেই এখন খুঁজে বার করতে হবে সেই ফাইল রাখার জন্যে সঠিক সাইজের একটা ফুটো। আপনি মেজাজে কবিতা লিখতে বসলেন, একটা ওয়ার্ড প্রসেসর খুললেন, প্রথমেই সিস্টেম আপনার কাছে জানতে চাইল, কত সাইজের ফাইল হবে? কারণ, সিস্টেমের মাথায় আছে, এই মহাবিশ্ব ফুটোময়। আপনি হয়তো, আপনার প্রগাঢ় আত্মকাব্যসচেতনতার বলে বলীয়ান হয়ে, এবং পরে যাতে কাব্যোচ্ছ্বাস বাড়লে ফাইলসাইজের কার্পণ্যে আপনাকে আটকে যেতে না-হয়, ফাইলের নিরিখে ফুটো যাতে অযথেষ্ট না হয়, জানিয়ে দিলেন, পাঁচ এমবি। সিস্টেম জানাল, বাতিল, ওই সাইজের ফুটো নেই। আবার ওয়ার্ড প্রসেসর খুলে চার এমবি তিন এমবি করে নিজের উচ্চাকাঙ্খা যে কমিয়ে আনতে পারেন না, তা নয়, কিন্তু, নিজেই ভেবে দেখুন, কবিতা লেখার তথা পিসি ব্যবহারের এটা একটা স্ট্যান্ডার্ড প্রক্রিয়া?

কিন্তু এবার ভাবুন, হার্ডডিস্কে যে ব্যবস্থা অসম্ভব, সেটা চমৎকার ভাবে সম্ভব হয়ে ওঠে যখন ফাইলসিস্টেমের কাঠামো এবং সাইজ এবং আভ্যন্তরীন ডিরেক্টরি আর ফাইলের সংখ্যা আর বদলাবে না। ঠিক সিডি-রম এবং ডিভিডি-রমের বেলায় যেটা ঘটে। অনেক আগে কম্পিউটার প্রকৌশলের গোড়ার দিকে চৌম্বক হার্ডডিস্কে এই কলটিগুয়াস রকমেই ফাইল রাখা হত, এর সরলতা এবং দ্রুততার জন্যে। ব্যবস্থাটা আবার ফেরত এসেছে সিডি-রম, ডিভিডি-রম এই জাতীয় অপ্টিকাল মাধ্যমে, যেখানে ফাইলব্যবস্থা মানে একবার লেখো এবং বারবার পড়ো। আইএসও-৯৬৬০ হল সিডি-রমের ফাইলব্যবস্থা। রম মানে মনে আছে তো, রিড-অনলি, শুধু পড়া যায়। এই রিড-অনলি ফাইলসিস্টেমের একটা আরাম ভাবুন, ফ্রি-ব্লক কী রইল, কী কী সাইজের কটা ফুটো, তা নিয়ে আর মাথা ঘামাতে হচ্ছেনা। আর তো লিখতে হবেনা। রিরাইটেবল সিডি মানে যেখানে পুরোনো তথ্য মুছে নতুন করে লেখা যায়, বা মাল্টিসেসন সিডি, মানে যেখানে আপনি সিডির গোটা তথ্যটা এক বারের জায়গায় একাধিক বারে পোড়াচ্ছেন, সেখানেও, যেমন বললাম, লেখার মুহূর্তে গোটা ব্যবস্থাটা কিন্তু প্রদত্ত, পড়ার সময়ে তারা আর বদলাচ্ছে না। 'ইন-সিডি' জাতীয় যে ব্যবস্থার কথা বললাম, সিডিকে ফ্লপির মত ব্যবহার করার, তাদের ফাইলব্যবস্থাগুলো, স্বাভাবিকভাবেই, আলাদা।

আইএসও-৯৬৬০ হল সবচেয়ে জনপ্রিয় এবং বহুল ব্যবহারের সিডি-রম ফাইলব্যবস্থা। আইএসও-৯৬৬০ হল একটা আন্তর্জাতিক স্ট্যান্ডার্ড, অষ্টাশিতে গৃহীত হয়েছিল। বাজারে প্রাপ্তব্য প্রতিটি সিডি-রমই প্রাথমিকভাবে এই স্ট্যান্ডার্ড মেনে চলে, তার সঙ্গে কোথাও কোথাও আরো বাড়তি কিছু থাকে। এই স্ট্যান্ডার্ড গৃহীত হয়েছিল যাতে এই গ্রহের প্রতিটি সিডি-ড্রাইভেই যে কোনো সিডি-রম পড়া যায়, যে অপারেটিং সিস্টেমেই সেটা লাগানো হোক, আর যে

অপারেটিং সিস্টেম থেকেই সেটা বানানো হয়ে থাক, যে রকম বাইট-ব্যবস্থাই ব্যবহার করা হোক সেই সিস্টেমে। পোসিক্স বা এলএসবি স্ট্যান্ডার্ডের ঠিক যে যুক্তি, আগেই বলেছি। হার্ডডিস্কের বেলায় আমরা যে এককেন্দ্রিক সিলিন্ডারের কথা বলেছি, সিডি-রমে ওরকম কিছু থাকে না। তার বদলে থাকে অপটিকাল ডিস্কের তলজুড়ে একটা ছেদহীন স্পাইরাল, আর এই স্পাইরাল বরাবর পরপর সাজানো বিটের সারি। সাজানোটা একরৈখিক বা লিনিয়ার। তবে খোঁজার সময় স্পাইরালটার শুরু থেকেই শুরু করতে হবে, তা নয়। যে কোনো জায়গাতেই খোঁজা যেতে পারে। স্পাইরাল শব্দটার কোনো বাংলা প্রতিশব্দ আছে বলে জানিনা। একটা ছবি দিই। একটা বৃত্তাকার পথ, যার পরিধি ক্রমে কমছে বা বাড়ছে, যে ভাবেই ভাবুন। আপনাদের জন্যে আমি একটা স্পাইরাল আঁকার চেষ্টা করলাম, দু-ঘন্টা ধরে। পারলাম না। এটা শেষে নেটে গিয়ে নামিয়ে একটু বদলে নিলাম কপিরাইট বাঁচাতে। কিন্তু এরাই বা আঁকল কী করে? যন্ত্র দিয়ে মনে হয়। বা হাতে এঁকে স্ক্যান করে? কে জানে বাপু? হাতে আঁকা যাক আর না-যাক, প্রকৃতি ভীষণ স্পাইরালপরায়ণ। একটা শাঁখের মুখ, যা শামুকের খোলা, বা জল বা হাওয়ার ঘূর্ণী। স্পাইরাল আর ফিবোনাক্কি বা ফিবোনাচি সংখ্যার সিরিজ নিয়ে ভারি মজা আছে। ফিবোনাক্কি সিরিজ মানে ০, ১, ১, ২, ৩, ৫, ৮, ১৩, ২১ ... এই সংখ্যার সারিটা, যার প্রত্যেকটাকেই পাওয়া যায় আগের দুটোকে যোগ করে। প্রকৃতির প্রায় সমস্ত স্পাইরাইলই তাদের জ্যামিতিতে এই ফিবোনাক্কি সিরিজ মেনে চলে।

যাকগে, সিডিরমের কথায় আসা যাক। সিডিরমের এই স্পাইরাল বরাবর বাইটগুলো পরপর সাজানো থাকে লজিকাল ব্লকের এককে। যার এক একটার মাপ ২৩৫২ বাইট। এর মধ্যে অনেকটা থাকে প্রিঅ্যাম্বল মানে গৌরচন্দ্রিকা — ধরুন মেটাডেটা গোছের, এই ফাইলসিস্টেমে কী তথ্য কী ভাবে আছে সেই বিষয়ে তথ্য। এর সঙ্গে এরর কারেকশন বা ত্রুটি সংশোধন, মানে, তথ্যের আকারটা সঠিক আছে কিনা সেটা মিলিয়ে নেওয়ার কৌশল। আর জেনুইন তথ্য থাকে ২০৪৮ বাইট। সংখ্যাটা কি চেনা লাগছে? এই জেনুইন তথ্যের মধ্যে কিন্তু মামা কাকা মাসি পিসি সমস্ত তথ্যই পড়ছে। গানের অডিয়ো সিডি থাকলে গানের আগের, পরের, দুটো গানের মধ্যের নৈঃশব্দ্য, দুটো ট্র্যাকের মধ্যেকার নিঃশব্দ জায়গাটা — সেটাও তথ্য। দুটো ট্র্যাকের মধ্যে এই নৈঃশব্দ্যটা আবার ডেটা সিডি বা তথ্যের সিডিতে থাকেনা। সিডি পোড়ানোর সময় দেখবেন, অনেক সময় মিনিট আর সেকেন্ডের মাপে গোটাটা বলছে, এই মাপটা হল ১ সেকেন্ড = ৭৫ ব্লক। প্রত্যেকটা সিডিরমের গোড়ায় থাকে ফাঁকা ষোলোটা ব্লক। ফাঁকা কারণ এগুলো নিয়ে আইএসও স্ট্যান্ডার্ডে কিছু বলে দেওয়া হয়নি। বুট-সিডি মানে যা দিয়ে বুট করা যায়, কখনো পুরো ওএস, কখনো থাকে ওএস চালু করার ন্যূনতম দরকারি ফাইলগুলো। বুট সিডির বেলায় এই ষোলোটা ব্লক ব্যবহার করা হয়। অন্য কাজেও কেউ ব্যবহার করতে পারে। এরপর আসে প্রাইমারি ভলিউম ডেসক্রিপটর। একটা সিডিকে একটা সেটের একটা ভলিউম হিশেবে ভাবুন — এই প্রাইমারি ভলিউম ডেসক্রিপটরে থাকে ওই সেটটার পরিচিতি, ভলিউমটার পরিচিতি, প্রকাশক এবং তথ্যটা যে সিডির জন্যে প্রস্তুত করেছে, তার পরিচিতি। সব মেশিনেই সিডিটা যাতে পড়া যায়, প্রাইমারি ভলিউম ডেসক্রিপটরে লেখা হয় শুধু বড় হাতের অক্ষর, সংখ্যা, আর অল্প কয়েকটা নির্দিষ্ট যতিচিহ্ন।

সিডিরম ফাইলসিস্টেমে রুট ডিরেক্টরি এবং তার মধ্যেকার ডিরেক্টরিগুলো ভরা থাকে বাস্তব ফাইল নয়, তাদের এন্ট্রি দিয়ে। খেয়াল রাখুন, এটা মেটাডেটা। এর শেষ ডিরেক্টরির শেষ এন্ট্রির মাথায় থাকে একটা বিশেষ বিট যা দিয়ে বোঝা যায়, এখানে ডিরেক্টরি এন্ট্রি শেষ হল। ডিরেক্টরি এন্ট্রিতে থাকে একটা ফাইলের ঠিকানা, সাইজ, তারিখ-সময়, এবং নাম। ফাইলের ঠিকানা মানে ফাইলসিস্টেমে শুরুর ব্লকের নম্বর। ফাইলের নামের মধ্যে একটা মূল নাম বা বেসনেম, তারপরে একটা '.', তারপরে এক্সটেনশন, তারপরে একটা সেমিকোলন বা ';', এবং সবার শেষে একটা বাইনারি ভার্শন নাম্বার। বেসনেম এবং এক্সটেনশন অংশে ছোটহাতের বা বড়হাতের বর্ণমালা এবং শূন্য থেকে নয় অব্দি অঙ্ক ব্যবহার করা যাবে। বেসনেমে আটটা অব্দি বর্ণ ব্যবহার করা যাবে, এবং এক্সটেনশনে তিন অব্দি। আপনার কি একটুও চেনা লাগছে ফাইলনামের এই ৮.৩ হিশেবটা?

ডিরেক্টরি এন্ট্রিতে তারিখ আর সময়ের ব্যাপারটায় একটু মজা আছে। আমরা Y2K বা ২০০০ সালে কম্পিউটার সিস্টেমে সময় এবং হিশেব রাখার সমস্যা নিয়ে বহ্বারম্ভে লঘুক্রিয়ার গল্প সবাই জানি। ঠিক এরকমই একটা সমস্যা আছে সিডিতেও। ডিরেক্টরি এন্ট্রিতে ফাইলের সময়ের ফিল্ডটায় একটা করে বাইট করে দেওয়া থাকে বছর, মাস, দিন, ঘন্টা, মিনিট, সেকেন্ড, আর টাইমজোন এই প্রতিটির জন্যে। টাইমজোন বা সময়এলাকা মানে ভৌগোলিক অবস্থান, গ্রিনউইচের তুলনায়। যেভাবে আমরা একটা দেশের সময় থেকে আর একটা দেশের সময়কে আলাদা করি। এবার, মজাটা ওই বছরের বাইট-টাতে। সময় গোনা শুরু হয়েছিল, আইএসও কমিটির সিদ্ধান্ত মোতাবেক, ১৯০০ থেকে। এরপর ২৫৬ বছর অব্দি, মানে ২১৫৫ সাল অব্দি সিডিতে বছরসংখ্যা লেখা যাবে। তার পরে আবার এটা ফিরে যাবে ১৯০০-তে। তবে কোনো ভয় নেই, ২১৫৫ অব্দি কেন ২০০৫-অব্দিই সিডি থাকে কিনা সন্দেহ। পশ্চিমে সিডি এখন বাতিল প্রকৌশল। এখানেও, যাদেরই একটু পয়সা আছে, ডিভিডি ড্রাইভ ব্যবহার করছে। ডিভিডি বার্নারের দামটা বড্ড বেশি, তবে এই পাঠমালাটা চলতে চলতেও সেটা কমছে। আচ্ছা, বলুন তো, কেন ২১৫৫ সালের পরে গিয়েই সমস্যাটা হচ্ছে? ওই বিশেষ সংখ্যাটাই কেন?

ডিরেক্টরি এন্ট্রিগুলো রাখা থাকে বর্ণানুক্রমে, পরপর অক্ষর অনুযায়ী। শুধু প্রথম দুটো এন্ট্রি বাদ দিয়ে। প্রথমটা হল ডিরেক্টরিটা নিজেই। ঠিক গ্নু-লিনাক্সে যেমন, '.' মানে সেই ডিরেক্টরিটা নিজেই, আর '..' মানে জনিতৃ ডিরেক্টরিটা, মানে, যে ডিরেক্টরির সাবডিরেক্টরি এই ডিরেক্টরিটা। ডিরেক্টরি এন্ট্রিগুলোর প্রথম দুটো সবসময়েই থাকে এই দুটো এন্ট্রি। একটা ডিরেক্টরিতে কতগুলো এন্ট্রি থাকতে পারবে তার কোনো নিষেধ নেই, কিন্তু কটা স্তর অব্দি নামা যাবে, তার নিয়ম আছে। এই ডিরেক্টরির মধ্যে সাবডিরেক্টরি, মা থেকে ছানা, এই ভাবে নামা যাবে আটটা স্তর অব্দি, এইট-লেভেল-নেস্টিং — একে বলে নেস্টিং (nesting)। ডিরেক্টরি এন্ট্রির একদম শেষে এসে আর একটা জিনিষ থাকতে পারে, তার নাম 'সিস্টেম ইউজ' (System Use)। এই ফিল্ডটা সবসময় সব সিস্টেমে থাকেনা, এবং অনেক সময় আলাদা আলাদা সিস্টেম এদের আলাদা আলাদা রকমে ব্যবহার করে। এই নেস্টিং আর ফাইলনামের কাঠামোর বিধিনিষেধ আর 'সিস্টেম-ইউজ' নামে ফিল্ডটার কথায় আসছি একটু বাদেই। সিডিরমের শুরুতেই থাকা এই ডিরেক্টরি এন্ট্রিগুলো শেষ হলে অবশেষে শুরু হয় বাস্তব ফাইলগুলো। তাদের রাখা থাকে পরপর কনটিগুয়াস ব্লকে। যেমন একটু আগেই বললাম, ডিরেক্টরি এন্ট্রিতে দেওয়া ফাইলের শুরুর ব্লকের ঠিকানা আর ফাইলের সাইজ দিয়ে এবার ফাইলটাকে পুরোপুরি বুঝে নিতে পারে সিস্টেম। খুব ছোট করে, খুব সরল করে, এই হল আইএসও ৯৬৬০ ফাইলব্যবস্থার কাঠামো। এর নিয়মকানুনের খুঁটিনাটিগুলো এবার জানব আমরা।

> শুনুন, একদম বাইরের একটা মজা আপনাদের না-শুনিয়ে পারছিনা, এইমাত্র, এই শৈত্যতরঙ্গাক্রান্ত সকালে, একটা অটো ঢুকল আমাদের পাড়ায়। প্রত্যেকটা মফস্বলের কিছু নিজস্ব ভাঁড়ামি থাকে, যারা নিকটতম মহানগরের ভাঁড়ামি থেকে স্বতন্ত্র। সেরকমই একটা, পুষ্পপ্রদর্শনী, মধ্যমগ্রামে হর বছর হয়। ওসব ডেসিবেল ছাড়ুন, মাইক নিয়ে প্রবল চেঁচাচ্ছে। একবার ভাবলাম বাইরে গিয়ে জিগেশ করি, পুষ্পকে যে প্রদর্শন করবেন, পুষ্পর বরের কোনো আপত্তি নেই তো? এর মধ্যেই এল ক্লাইম্যাক্সটা। এরা চেঁচানো ছেড়ে গান বাজানো শুরু করল, অটোটা পাড়া পেরিয়ে যেতে যেতে গানটার বেশ কয়েকলাইন শোনা গেল। কী গান হতে পারে, একবার মনে মনে কল্পনা করুন, যদি পারেন, এইমাত্র শীতের সকালে আমায় যে গরম মাছভাজাটা দেওয়া হল, কাল আমার জলের ফিল্টার পরিস্কার করার বখশিশ হিশেবে, সেটা আপনাকে দিয়ে দেব। গানটা কী, পারলেন ভাবতে? দেবব্রতর 'পুষ্প দিয়ে মারো যারে —'। পাগলা মাছ খাবি তো মাটি খা।

আইএসও ৯৬৬০-র তিনটে পর্যায় বা লেভেল আছে। লেভেল ওয়ানে বিধিনিষেধ খুব বেশি। ফাইলনামের ওই ৮.৩ কাঠামোটা রাখতেই হবে, বর্ণমালার নিয়মও মানতে হবে। ফাইলদেরও একদম নিখুঁতভাবে পরপর সন্নিহিত বা কনটিগুয়াস হতে হবে। এছাড়া আরো আছে। ডিরেক্টরি নামের কোনো এক্সটেনশন থাকবে না, তাদের হতে হবে আট অক্ষরের মধ্যে। এটা এসেছিল, বুঝতে পারছেন, এমএস-ডস ফাইলসিস্টেমের সঙ্গে মিলিয়ে, সেখানে ডিরেক্টরিদের কোনো পদবি দরকার পড়েনা, পদবি হয় শুধু পাঁচপেঁচি আমফাইলদের। দু নম্বর লেভেলে নামের দৈর্ঘের নিষেধটা একটু শিথিল। সেখানে নামের অক্ষরের সংখ্যা হতে পারে ৩১ অব্দি, কিন্তু সেই বর্ণগুলো আসবে নিয়ম মেনে। তিন নম্বর লেভেলে ফাইলগুলোকে আর নিখুঁত ভাবে কনটিগুয়াস না-হলেও চলে, একটা ফাইল একাধিক অংশে ভাগ করা থাকতে পারে, তার এক একটা অংশের ব্লকগুলো কনটিগুয়াস হলেই হবে। এই রকম আরো কিছু।

আইএসও ৯৬৬০ প্রথার এই বিধিনিষেধগুলোকে বদলাতে এল রকরিজ-এক্সটেনশন (Rock Ridge Extension)। রকরিজ নামটা এসেছে জিন ওয়াইল্ডারের 'ব্লেজিং স্যাডলস' নামে একটা সিনেমার একটা শহরের নাম থেকে, কারণ, যে কমিটি এই প্রথাটা স্থির করে তাদের একজনের খুব পছন্দ ছিল সিনেমাটা। ফাইলের মালিকানা এবং অনুমতির যে কাঠামো আলোচনা করেছি আমরা সাত নম্বর দিনে, রকরিজ এক্সটেনশন ব্যবহার করলে সেটা একটা সিডিরম ফাইলসিস্টেমেও বজায় রাখা যায়। এমনকি রাখা যায় মেজর এবং মাইনর ডিভাইস নম্বর আর সিম্বলিক লিংক। এগুলোর কথা কি মনে করতে পারছেন। ফাইলনামের দৈর্ঘ এবং বর্ণমালা সংক্রান্ত কোনো বিধিনিষেধ থাকেনা। আচ্ছা, ফাইলনামের ৮.৩ ছকে যেমন এমএস-ডসের ছায়াপাত ছিল, এখানে সেরকম আর কিছু খুঁজে পাচ্ছেন না? না? সত্যি, কী ভালো আপনার অবজার্ভেশন। রকরিজ দিয়ে আরো কিছু করা যায়। তার মধ্যে সবচেয়ে জরুরি হল পুনঃস্থাপন বা 'রিলোকেশন' (relocation)। যে রিলোকেশন দিয়ে ওই আট স্তর অব্দি নেস্টিং-এর নিষেধটাকে এড়ানো যায়। একটা ডিরেক্টরিকে আপনি রিলোকেশন করলেন মানেই, এবার তার মধ্যে আবার আটটা স্তর অব্দি নেমে যেতে পারবেন। এই ভাবে চলবে। যে সিস্টেম রকরিজ চেনেনা সে এই সিডিরমটাকে রকরিজ বাদ দিয়ে এমনি সিডিরমের মত করেই পড়ে। আপনি কি এখনো বুঝতে পারছেন না, এই রকরিজ এক্সটেনশন কেন আনা, কাদের আনা? আইএসও ৯৬৬০ আসার পরপরই ইউনিক্সের বিভিন্ন ঘরানার লোকেরা মিলে যখন রকরিজ এক্সটেনশন প্রথা চালু করে তখনো কিন্তু গ্নু-লিনাক্সের জন্ম হয়নি, কিন্তু নিয়মকানুনগুলো স্বাভাবিকভাবেই গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সের জন্যেই প্রযোজ্য। একটা জিনিষ খেয়াল করুন, রকরিজ যে বিধিনিষেধগুলোকে ভাঙল তার মধ্যে নামের দৈর্ঘের ফ্যাচাংটা কিন্তু শুধু ইউনিক্সের না। উইনডোজ ৯৫ এবং তার পরবর্তী উইনডোজ গুলোতেও এটা একটা অসুবিধে। কারণ, সেখানেও নামের দৈর্ঘের কোনো নিষেধ নেই। তবে, আগেই তো বলেছি, উইন্ডোজে এই ৮.৩ ছকটা এখনো একইসঙ্গে আছে এবং নেই। যাইহোক, এই খিঁচকেল নিয়মগুলো এড়ানোর জন্যেই এল রকরিজ, এবং রকরিজ ছাড়াও আরো একটা জিনিষ, তার নাম জলিয়েট এক্সটেনশন (Joliet Extension)। জলিয়েট প্রথায় লম্বা ফাইলনাম আনা গেল, বর্ণমালার বিধিনিষেধ প্রায় নেই হয়ে গেল, আটটার বেশি নেস্টিং করা গেল, এবং সবচেয়ে মজার কথা ডিরেক্টরি নামেরও এক্সটেনশন দেওয়া গেল। এই শেষটা খুব অদ্ভুত। কারণ, উইনডোজে ডিরেক্টরি নামের এক্সটেনশন ব্যবহার হয়না।

আপনার ম্যানপেজ পড়ার একটা টাস্ক দেওয়া যাক। হার্ডডিস্কে থাকা কোনো একটা ডিরেক্টরি আর ফাইলের সমাহারকে সিডিরমের ফাইলসিস্টেম বানানোর সময়, আগেই বলেছি, একটা উপায় হল 'mkisofs'। এবার, ধরুন আমাদের জিএলটির সিডির জিনিষপত্তরগুলো, কয়েকহাজার ওয়েবপেজ আর কয়েকশো পিডিএফ সহ, রাখা আছে আমার মেশিনের '/arkive' ডিরেক্টরির মধ্যে '/arkive/linux.books' নামে একটা সাবডিরেক্টরিতে। এই গোটা ডিরেক্টরিটাকে আমি একটা ইমেজ বানাই 'glt-iso' নামে। এই ইমেজটাকে বানানোর জন্যে এর জন্যে সচরাচর কমান্ড দিই, '`mkisofs -DJU -joliet-long -o glt.iso /arkive/linux.books`'। আপনি ম্যানপেজ পড়ে দেখুন তো, এই বাড়তি অপশানগুলো আমি কেন দিই? একটা কথা বলে রাখি, নামের বা নেস্টিং-এর কোনো বিধিনিষেধ মানা হয়নি, এবং আমি চাই আমাদের জিএলটির সিডিটাকে গ্নু-লিনাক্স বা উইনডোজ যে কোনো অপারেটিং সিস্টেম থেকেই সমানভাবে পড়া যাক। এর পরে, ইমেজ বানানো যেই হয়ে গেল, সিডিতে পোড়াতে হয় ইমেজটা। তখন ব্যবহার করি 'cdrecord' বলে একটা কমান্ড। সচরাচর স্বাভাবিক অপশানসহ কমান্ডটা হয়, '`cdrecord dev=0,0,0 speed=4 -v -dao -fs 64m -data glt.iso`'। এখানেই বা অপশানগুলোর মানে কী? বার করুন তো, ম্যান পড়ে। আগেই তো বলেছি, সিডিরেকর্ড ব্যবহার করার আগে রুট হয়ে নেবেন, বা বার্নার ড্রাইভের ডিভাইস ফাইলের উপর ইউজারের ব্যবহারের অনুমতি বানিয়ে নেবেন। আর, ডিভাইস ফাইল চেনার কমান্ড তো আগেই বলেছি, '`cdrecord scanbus`', এটাও রুট হয়ে। এবার বলুন তো, '`-V "glt-mad@ilug-cal.org"`' অপশানটা 'mkisofs'-এর কমান্ডটার মধ্যে যোগ করলে তার মানে কী হত?

৩.৭।। দুম্বো সাইজের ফাইল আর ফাইলব্যবস্থা

ছটা ফাইলসিস্টেম বুড়ি ছুঁয়ে আসা গেল। এছাড়াও অন্য ফাইলসিস্টেম ব্যবহার করা যায় গ্নু-লিনাক্সে। কিন্তু সেগুলো অনেক বিশেষ ধরনের প্রয়োজন। আমাদের পাঠমালার এক্তিয়ারের বাইরে। যদি তেমন দরকার পড়ে, কোলকাতা লাগ-এ (ilug-cal@ilug-cal.org) লিখতে পারেন। 'cat /proc/filesystems' করে আমরা যে ফাইলসিস্টেমগুলোর তালিকা আমরা পেয়েছিলাম, তার বেশ কয়েকটার আলোচনা হল, কিন্তু কতকগুলোর হয়নি। সেগুলোর বিষয়ে প্রাথমিক ধারণা আপনার সিস্টেমের ভিতরেই কোথায় পাওয়া যাবে, মনে করতে পারছেন? যাকগে, কাজের কথায় যাওয়া যাক। আগেই বলেছিলাম এবার আমরা যাব ফাইলসিস্টেম কত বড় সাইজের অব্দি হতে পারে, এবং তাতে একটা একক ফাইল কত বড় সাইজের অব্দি হতে পারে, তার আলোচনায়। এই হিশেবটা কিন্তু রোজই বদলাচ্ছে। রোজই নতুন প্রকৌশল আসছে, পুরোনো প্রকৌশল বাতিল হচ্ছে। বদলাচ্ছে সফটওয়ার এবং হার্ডওয়ার দুটো স্তরেই। তবু, একটা আন্দাজ করা যাক। বড় ফাইলব্যবস্থার এবং ফাইলের যে বিকট হিশেব এখুনি আসছে, সেই সাইজের কোনো কিছু আমি দেখা বা চেনা তো দূরের কথা, ভেবেও উঠতে পারিনা। হিশেবটা নেট থেকে পাওয়া, এবং তাও একদম নতুন না।

হিশেবটা দেওয়ার আগে, বিট বাইট এবং তাদের বৃহত্তর এককগুলো নিয়ে দু-একটা কথা বলে নিই। শূন্যয় বলেছি, পরেও এসেছে। কিন্তু তখন আমরা বিট বাইটের উপর কিলো মেগা গিগা নিয়েই খুশি ছিলাম, টেরা-পেটা-এক্সা-জিটা-ইয়োটার বিকটতায় পৌঁছনোর দরকার পড়েনি। এই পরিমাপগুলোর স্ট্যান্ডার্ড নিয়েও কিছু কথা বলার আছে। আমরা একাধিকবার বলেছি, কম্পিউটারের তথ্যের পরিমাপটা হয় বাইনারি পদ্ধতিতে। কিন্তু মজার কথা হল, এর গুণিতক বা মাল্টিপলগুলোর জন্যে যে প্রাকশব্দ বা প্রিফিক্স আমরা ব্যবহার করি, যেমন, কিলো, সেটা কিন্তু মেট্রিক পদ্ধতির। কিলোবাইট মানে আমরা ধরছি ১০২৪ বাইট, অথচ, মেট্রিক ব্যবস্থায় কিলো হল ১০০০। কারণ মেট্রিক পদ্ধতি হল ডেসিমাল বা দশমিক। এখানে, আত্মীয়তাটা এইরকম যে, ডেসিমাল সংখ্যা ১০০০ বা ১০^৩-এর সবচেয়ে কাছে বাইনারি প্রথায় যে সংখ্যাটায় পৌঁছচ্ছি সেটা হল ১০২৪ বা ২^১০। ('^' দিয়ে আমরা পাওয়ার বা সূচক বুঝিয়েছি, ২^৩ মানে ২×২×২)। তাই, মেট্রিক কিলো শব্দটা যখন বাইনারিতে আসছে, তার মান ১০০০ থেকে বদলে ১০২৪ হয়ে যাচ্ছে। আবার কিলো বলতে দশমিক কিলোও চালু। তাতে লেখা হত ১ কিলোবাইট (বাইনারি) মানে ১০২৪ বাইট, ১ কিলোবাইট (দশমিক) মানে ১০০০ বাইট। কিন্তু এতে ভজকটোটা বেড়েছে বই কমেনি। '৯৮ সালের ডিসেম্বরে আইইসি (IEC — **I**nternational-**E**lectrotechnical-**C**ommission) একটা নতুন আইইসি স্ট্যান্ডার্ড চালু করে। তাতে বলা হয়, মেট্রিক ওই প্রাকশব্দ বা প্রিফিক্স বদলে বাইনারি প্রক্রিয়ার গুণিতকগুলোর জন্যে নতুন কিছু প্রাকশব্দ ব্যবহার করতে হবে। এই নতুন প্রাকশব্দগুলো তৈরি হবে মেট্রিক প্রাকশব্দের প্রথম দুই বর্ণ আর 'বাইনারি' এই শব্দটার প্রথম দুই বর্ণ মিলিয়ে। যেমন কিলোবাইট শব্দটা এখন থেকে হবে কিবিবাইট। কিবি (Kibi) মানে, কিলো (Kilo) থেকে আসছে কি (Ki) আর বাইনারি (Binary) থেকে আসছে বি (bi)। মেগাবাইটের 'মেগা' বদলে ব্যবহার করতে হবে মেবিবাইট, গিগাবাইটের 'গিগা' বদলে ব্যবহার করতে হবে গিবিবাইট, ইত্যাদি। কিন্তু এখনো আইইসি ব্যবস্থা তেমন জনপ্রিয় হয়নি। আমরা তাই সাবেক বাইনারি রকমেই হিশেবটা দিচ্ছি।

এক বিট (bit) = ০ বা ১, একক লেখা হয় 'b' দিয়ে।
এক বাইট (byte) = ৮ বিট, একক লেখা হয় 'B' দিয়ে।
এক কিলোবাইট (Kilobyte) = ১০২৪ বাইট = ২^১০ বাইট, একক লেখা হয় 'KB' দিয়ে।
এক মেগাবাইট (Megabyte) = ১০২৪ কিলোবাইট = ২^২০ বাইট, একক লেখা হয় 'MB' দিয়ে।
এক গিগাবাইট (Gigabyte) = ১০২৪ মেগাবাইট = ২^৩০ বাইট, একক লেখা হয় 'GB' দিয়ে।
এক টেরাবাইট (Terabyte) = ১০২৪ গিগাবাইট ২^৪০ বাইট, একক লেখা হয় 'TB' দিয়ে।
এক পেটাবাইট (Petabyte) = ১০২৪ টেরাবাইট = ২^৫০ বাইট, একক লেখা হয় 'PB' দিয়ে।
এক এক্সাবাইট (Exabyte) = ১০২৪ পেটাবাইট = ২^৬০ বাইট, একক লেখা হয় 'EB' দিয়ে।
এক জিটাবাইট (Zettabyte) = ১০২৪ এক্সাবাইট = ২^৭০ বাইট, একক লেখা হয় 'ZB' দিয়ে।
এক ইয়োটাবাইট (Yottabyte) = ১০২৪ জিটাবাইট = ২^৮০ বাইট, একক লেখা হয় 'YB' দিয়ে।

এবার আসুন, এতাবৎ মহতী এককের বলে বলীয়ান হয়ে আসুন আমরা সেইসব জগদ্দল ফাইলব্যবস্থা আর ফাইল নিয়ে নাড়াচাড়া করি, ভেজানো মুগ আর বাদাম রেখেছেন তো? একদম প্রথমে গ্নু-লিনাক্সে দুই জিবি অব্দি সাইজের

ফাইল বানানোর সংস্থান ছিল। তারপরেই পরিস্থিতিটা খুব দ্রুত বদলে গেল। তখনো মাল্টিমিডিয়া সেভাবে সিনে আসেনি। মাল্টিমিডিয়া আর গেমস-এর দৈত্যাকার সাইজের ফাইল নিয়ে কাজের প্রয়োজন এসে গেল। আর খুব বড় বড় সাইজের ডেটাবেস বা তথ্যভাণ্ডারও এর আগে অব্দি সেভাবে ছিলনা। যা প্রায় প্রাত্যহিক হয়ে দাঁড়াল। এছাড়া এরপর যখন বড় বড় সার্ভারে গ্নু-লিনাক্স দিয়ে কাজ হতে শুরু করল, কারনেল আর সি লাইব্রেরিগুলোকেও সেইমত বদলে নেওয়া হল, যাতে দুই জিবির চেয়ে বড় সাইজের ফাইলও বানানো এবং ব্যবহার করা যায়। এককথায় এর নাম এলএফএস (LFS — **L**arge-**F**ile-**S**upport)। এখন প্রায় সবগুলো ফাইলব্যবস্থাতেই এলএফএস চালু আছে, যাতে সিস্টেমে খুব বড় বড় ফাইল নিয়েও কাজ করা যায়।

ফাইলসিস্টেম	ফাইল-সাইজের সীমা	ফাইলসিস্টেম-সাইজের সীমা
ইএক্সটিটু বা থ্রি, ১ কেবি ব্লক	১৬ জিবি (GB)	২ টিবি (TB)
ইএক্সটিটু বা থ্রি, ৪ কেবি ব্লক	২ টিবি (TB)	১৬ টিবি (TB)
রাইজারএফএস ৩.৫	৪ জিবি (GB)	১৬ টিবি (TB)
এক্সএফএস	৮ ইবি (EB)	৮ ইবি (EB)
জেএফএস, চার কেবি ব্লক	৮ ইবি (EB)	৪ পিবি (PB)

এর বাইরে গ্নু-লিনাক্স কারনেলের কিছু আভ্যন্তরীন ফাইলসাইজ সীমা আছে। ৩২-বিট সিস্টেমে একটা ফাইল বা ব্লক ডিভাইসের সর্বোচ্চ সাইজ হতে পারে ২ টিবি (TB)। তবে লজিকাল ভলিউম ম্যানেজার বা এলভিএম (LVM — **L**ogical-**V**olume-**M**anager) ব্যবহার করে এর চেয়েও বড় ফাইলব্যবস্থা বানানো যায়। ৬৪-বিট সিস্টেমে এই সীমাটা হল ৮ ইবি (EB)। উপরে আমরা কয়েকটা মাত্র ফাইলসিস্টেমের উদাহরণ দিলাম, এরকম সব ফাইলসিস্টেমেরই সাইজের একটা নিজস্ব সর্বোচ্চ সীমা থাকে, তাতে ব্যবহার্য ফাইলের সাইজেরও থাকে। এসব আমাদের নিজের পিসির স্ট্যান্ড অ্যালোন সিস্টেমে কারুর লাগে বলে মনে হয়না। আমার মেশিনে সবচেয়ে বড় ফাইল বানিয়েছিলাম, টেন কমান্ডমেন্টস এর চারটে ভিসিডি থেকে একটা এভিআই (*.avi) ফাইল, এমপ্লেয়ারে ক্যাট করে ওসব করা যায়। তার সাইজ ছিল আড়াই জিবি মানে ২.৬ গিগাবাইট। এটাই আমার টাইটানমানসের চূড়ান্ত। আমাদের জিএলটির অশোকদা আর লাগের তথাগত ডিভিডি এনকোড ডিকোড করে, ওদের লাগতে পারে।

৪।। তুমি যে আমার, মাউন্ট, তুমি যে আমার

'মাউন্ট করা' — এই কথাটার হুবহু সংজ্ঞাটা হল একটা পার্টিশনে থাকা একটা ফাইলসিস্টেম (তথ্য-কাঠামো তথা ডিরেক্টরি এবং ফাইল লেখার/রাখার/পড়ার ব্যবস্থা এই অর্থে, যেমন এক্সএফএস বা ইএক্সটিথ্রি) একটা ঐক্যবদ্ধ ফাইলসিস্টেমে (একটা চালু কেজো অপারেটিং সিস্টেমের মোট ডিরেক্টরি এবং ফাইলের হায়েরার্কিন্যস্ত ব্যবস্থা এই অর্থে, যেমন আমার মেশিনের সুজে বা স্ল্যাকওয়ার বা উইন্ডোজ) যুক্ত হয়ে পড়া। এই শেষ বাক্যে আরো একবার 'ফাইলসিস্টেম' শব্দটার দুটো অর্থকে আলাদা করে দিলাম। মোদ্দা কথায়, একটা চালু ওস-এ একটা পার্টিশনকে যুক্ত করে নেওয়াই হল মাউন্ট। ওএস-টার নিজের একটা হায়েরার্কিন্যস্ত ডিরেক্টরিব্যবস্থা আছে। সেখানে স্পষ্ট একটা রুট ডিরেক্টরি আছে। তার থেকে শিরা-উপশিরার মত করে চারদিকে ছড়িয়ে পড়া ডিরেক্টরি সাবডিরেক্টরি আর তাদের পেটের মধ্যে বোঝাই ফাইলের একটা হায়েরার্কিন্যস্ত ডিরেক্টরিব্যবস্থা আছে। আবার, মাউন্টনীয় একটা পার্টিশনের নিজের ফাইলসিস্টেমের মধ্যেও একটা হায়েরার্কিন্যস্ত ডিরেক্টরি ব্যবস্থা আছে। পার্টিশনের ফাইলব্যবস্থাটুকুর আবার নিজের মত করে একটা রুট ডিরেক্টরিও আছে। সেই রুট ডিরেক্টরি থেকে আবার শাখা-প্রশাখার মত বেরিয়ে এসেছে পার্টিশনের নিজের ফাইলসিস্টেমের নানা ডিরেক্টরি সাবডিরেক্টরি এবং তাদের মধ্যে থাকা ফাইল। এবার, যখনি ওএস-এর রুট ডিরেক্টরির মধ্যে কোনো একটা ডিরেক্টরিতে বা সাবডিরেক্টরিতে কোনো একটা পার্টিশন মাউন্ট হয়, পার্টিশনের ফাইলসিস্টেমের রুট ডিরেক্টরিটা ওএস-এর ঐক্যবদ্ধ মূল ফাইলসিস্টেমের রুট ডিরেক্টরিতে একটা সাবডিরেক্টরি হয়ে যায়। সাত নম্বর দিনের ৪ নম্বর সেকশনে সম্পূর্ণ ফাইলসিস্টেমটা বোঝানোর জন্যে একটা বড় ছবি দেওয়া ছিল। যেখানে সুজে ওএস-এর গোটা ঐক্যবদ্ধ ফাইলসিস্টেমটা দেখিয়েছিলাম। সুজের মূল রুট পার্টিশন '/', তার মধ্যে একটা ডিরেক্টরি '/mnt', তার মধ্যে চারটে সাবডিরেক্টরি '/mnt/arkive', '/mnt/slackware',

‘/mnt/windows/c’, এবং ‘/mnt/windows/d’। যেখানে চারটে আলাদা আলাদা পার্টিশন মাউন্ট হয়, ‘/dev/hdb5’, ‘/dev/hda6’, ‘/dev/hda1’, এবং ‘/dev/hda5’। আর সুজের নিজের ‘/’ ডিরেক্টরিতে মাউন্ট হয় তার রুট পার্টিশন ‘/dev/hdb3’। সোয়াপ আর বুটকে ছেড়ে দিলে এই পাঁচটা মূল পার্টিশনকে নিয়ে কাজ করে সুজে ওএস। ছবিটা একবার দেখুন, এবং নিজেই বোঝার চেষ্টা করুন। একটা জিনিষ খেয়াল করুন, একটা পার্টিশনে একটা রাইজার, ইএক্সটি, এক্সএফএস, ফ্যাটথার্টিটু গোছের ফাইলব্যবস্থা থাকতে পারে এবং থাকে ওএস নিরপেক্ষ রকমে। যখন যে ওএস তাকে মাউন্ট করে, তার ঐক্যবদ্ধ ফাইলসিস্টেমে জুড়ে যায় পার্টিশনটা। ঐক্যবদ্ধ ফাইলসিস্টেম মানেই সেটা একটা নির্দিষ্ট ওএস-এর, একটা জ্যান্ত কেজো জিনিষ। যা একটা পার্টিশনের ফাইলসিস্টেমের বেলায় সত্যি নয়। মেশিন থেকে আলাদা করে, ওএস থেকে আলাদা করে একটা হার্ডডিস্কের একটা পার্টিশনে একটা ফাইলসিস্টেম বানিয়ে রাখা হয়, কখন কোন ওএস তাকে মাউন্ট করবে, ব্যবহার করবে, জ্যান্ত করে তুলবে।

আজকের আলোচনার ১ নম্বর সেকশনের দ্বিতীয় ছবিটা, মানে মাউন্টের ছবিটা দেখুন এবার। আমার মেশিনের সুজে সিস্টেমে ‘/dev/hda6’ পার্টিশনটা ‘/mnt/slackware’ ডিরেক্টরিতে মাউন্ট করা অবস্থায় ওই ডিরেক্টরিতে দাঁড়িয়ে ‘`ls`’ দেখাবে ‘/mnt/slackware/bin’, ‘/mnt/slackware/boot’, ‘/mnt/slackware/dev’, ‘/mnt/slackware/etc’ ইত্যাদি ডিরেক্টরিগুলো। তার মানে, ওই ছবির একদম নিচের ডানদিকের নলপথের স্টপকক্টা খোলা এখন খোলা আছে। তাই নলপথ বা মাউন্টপয়েন্ট ‘/mnt/slackware’ বেয়ে তথ্য চলাচল করছে। এই ডিরেক্টরিগুলো স্ল্যাকওয়ার পার্টিশনের রুট ডিরেক্টরির মধ্যেকার সাবডিরেক্টরি। স্ল্যাকওয়ার অপারেটিং সিস্টেমে বুট করলে এদের স্ল্যাকওয়ার সিস্টেমের রুট পার্টিশনের সাবডিরেক্টরি হিশেবেই দেখাত। তখন, স্ল্যাকওয়ার অপারেটিং সিস্টেমে, ‘/home/dd’ ডিরেক্টরিতে কোনো ফাইল আমি লিখে থাকলে এখন সুজেতে তাকে পাব ‘/mnt/slackware/home/dd’ ডিরেক্টরির মধ্যে একই চেহারায়। কিন্তু মাউন্ট না-করা অবস্থায়, যখন ওই স্টপকক্টা বন্ধ আছে, পার্টিশনটা বিচ্ছিন্ন হয়ে গেছে মূল ঐক্যবদ্ধ ফাইলব্যবস্থা থেকে, তখন আমি ওই একই ‘/mnt/slackware’ ডিরেক্টরিতে দাঁড়িয়ে একই ‘`ls`’ কমান্ড দিলে দেখাবে কিছুই নেই। কারণ তখন ওই পার্টিশনের নিজস্ব ফাইলব্যবস্থা মূল ঐক্যবদ্ধ ডিরেক্টরিন্যস্ত ফাইলব্যবস্থায় যুক্ত নেই, মানে মাউন্ট করা নেই। যখন ‘/mnt/windows/c’ আর ‘/mnt/windows/d’ নলপথের স্টপকক দুটো খোলা আছে, মানে ‘/dev/hda1’ আর ‘/dev/hda2’ পার্টিশনদুটো মাউন্ট করা আছে, তখন ‘/mnt/windows/c’ আর ‘/mnt/windows/d’ ডিরেক্টরিদুটোয় পাব উইনডোজের সি-ড্রাইভ আর ডি-ড্রাইভের গোটা ফাইলব্যবস্থা। স্টপকক বন্ধ মানে আনমাউন্ট হলেই তারা হারিয়ে যাবে। তখন ‘`ls`’ কমান্ড ‘/mnt/windows/c’ আর ‘/mnt/windows/d’ অব্দি এসে থেমে যাবে। ভিতরে ফাঁকা। একই ঘটনা ঘটবে ‘/dev/hdb5’ পার্টিশন মাউন্ট করা থাকলে ‘/mnt/arkive’ ডিরেক্টরিতে। মাউন্টগুলো করা হবে কি হবে-না, কোথায় হবে, কখন হবে, সেগুলো আপনার ইচ্ছে অনুযায়ী ঘটবে। আপনি যদি নিজের কোনো ইচ্ছে আলাদা করে না জানিয়ে থাকেন, তাহলে সেটা হবে সিস্টেমের ডিফল্ট বা চালু পথে। অর্থাৎ, মূল ঐকবদ্ধ ফাইলব্যবস্থা কোনো একটা পার্টিশনকে যে কোনো সময় ‘তুমি যে আমার’ করে নিতে পারে, মাউন্ট করে। জাহাজে ভাসমান নাবিকদের মত। যে বন্দরে মাউন্ট করছে, সেই ডিরেক্টরিটাই মাউন্ট-পয়েন্ট।

একটা পার্টিশনকে আপনি যে কোনো ডিরেক্টরিতে মাউন্ট করতে পারেন, ‘mount’ কমান্ড দিয়ে। আমার সিস্টেমে অটোমাউন্ট অফ করা আছে, আগেই বলেছি, রুট আর বুট ছাড়া আরে কোনো পার্টিশনই সিস্টেম নিজে মাউন্ট করেনা। এবার, আমার ‘/arkive/linux.books’ ডিরেক্টরিতে কোনো একটা ফাইল পড়তে চাইলে তাকে পাব কী করে, মাউন্ট তো করা নেই। তার মানে এখন ‘`ls /arkive`’ মারলে শূন্য দেখাবে। এই ‘/arkive’ ডিরেক্টরিতে এবার ‘/dev/hdb5’ পার্টিশনটা মাউন্ট করার কমান্ড দিতে হয়, ‘`mount /dev/hdb5 /arkive`’। কমান্ডটা রুট হয়ে দিতে হয়, যদিনা অন্য কোনো বিশেষ ব্যবস্থা করা থাকে ‘/etc/fstab’ ফাইলে। আমার সিস্টেমে করা আছে, তার কথায় আসছি। এবার এই গোটা কমান্ডটার কাঠামোটা খেয়াল করুন। প্রথমে ‘mount’ কমান্ড, তারপর পার্টিশনের নাম, তারপর মাউন্ট করার ডিরেক্টরি। এটা মাথায় রাখবেন — মাউন্ট করা হয় একটা পার্টিশনকে, করা হয় একটা ডিরেক্টরিতে। এখন থেকে ওই পার্টিশনের ফাইলদের ওই ডিরেক্টরিতে পাওয়া যাবে।

কমান্ডের আদত কাঠামোটা আর একটু বড়, ‘mount -t <fstype> -o <options> <device> <dir>’। এর মধ্যে আমরা প্রথম দুটো অংশ বাদ দিয়েছি, ‘-t’-এর পরে ‘<fstype>’ অংশে দিতে হয় ফাইলসিস্টেমের মানে তথ্যকাঠামোর নাম। যেমন আমরা জানি আমাদের ‘/dev/hdb5’ পার্টিশনটায় ফাইলসিস্টেম হল রাইজারএফএস

(reiserfs), এখানে সেই নামটা দিতে হত। সচরাচর এটা না-দিলেও চলে, নিজেই বুঝে নেয়, '/etc/fstab' ফাইলে দেওয়া থাকলে তো কথাই নেই। আর "-o'-এর পরে '<options>' অংশে বিভিন্ন রকম অপশান দেওয়া যায়। যেমন, পার্টিশনটা মাউন্ট করার পর আমরা শুধু ফাইল পড়ব, নাকি পড়া/লেখা দুই-ই করব। কিছু না-দিলে মাউন্ট হয় ডিফল্ট রকমে, 'rw', লেখা পড়া দুই-ই চলে। রিড-অনলি রকমে মাউন্ট করলে, কোনো ফাইল লিখতে না-চাইলে অপশান দিতে হত 'ro' মানে রিড-অনলি। কমান্ড হত, '`mount -o ro /dev/hdb5 /arkive`'। '<device>' মানে পার্টিশনের ডিভাইস ফাইল, এখানে '/dev/hdb5'। '<dir>' মানে ডিরেক্টরি বা মাউন্টপয়েন্ট, এখানে '/arkive'। আরো বুঝতে চাইলে ম্যানপেজ আছে। এইভাবে আমরা মাউন্ট করতে পারি যে কোনো পার্টিশনে, এমনকি একাধিক জায়গাতেও মাউন্ট করা যায় একই পার্টিশন ডিভাইসকে। বা মাউন্ট করা যায় একটা ইমেজ ফাইলকেও। সেসব এখনি করবেন না, ধীরে ধীরে শিখে নিন, হাউটু পড়ুন।

একটু এবার '/etc/fstab' ফাইলটার সঙ্গে পরিচিত হয়ে নিন। আমার মেশিনের সুজে সিস্টেমের '/etc/fstab' ফাইলটা তুলে দিলাম এখানে। আমার মেশিনে আরো একটা 'fstab' ফাইল আছে, '/mnt/slackware/etc/fstab'। এই যে ঠিকানাটা, এটা সুজে সিস্টেমের মধ্যে থেকে যখন দেখি ফাইলটাকে, স্ল্যাকওয়ার থেকে দেখলে, এটাই '/etc/fstab' হয়ে ওঠে। কারণ তখন তো '/' ডিরেক্টরি মাউন্ট হওয়ার পার্টিশনটাই বদলে যায়।

```
#device     mountpoint          fs         options               dump  fsck
/dev/hdb3   /                   xfs        defaults              1     1
/dev/hdb1   /boot               xfs        defaults              1     2
/dev/hdb2   swap                swap       pri=42                0     0
/dev/hda1   /mnt/windows/c      vfat       rw,noauto,noexec      0     3
/dev/hda5   /mnt/windows/d      vfat       rw,noauto,noexec      0     3
/dev/hda6   /mnt/slackware      reiserfs   rw,noauto,user,exec   0     3
/dev/hdb5   /mnt/arkive         reiserfs   rw,noauto,exec        0     3
/dev/cdrom  /media/cdrom        auto       ro,noauto,user,exec   0     0
/dev/cdwri  /media/cdwri        auto       ro,noauto,user,exec   0     0
/dev/fd0    /media/floppy       auto       noauto,user,sync      0     0
devpts      /dev/pts            devpts     mode=0620,gid=5       0     0
proc        /proc               proc       defaults              0     0
usbdevfs    /proc/bus/usb       usbdevfs   noauto                0     0
```

এই ফাইলটা কিন্তু হুবহু আমার সুজে সিস্টেমের '/etc/fstab' ফাইলটা নয়। একটু একটু বদলেছি। শুরুর যে লাইনটা, '#' দিয়ে শুরু, ওটা সুজেতে থাকেনা, অন্য ডিস্ট্রোয় থাকে, ফাইলটা বোঝার সুবিধের জন্যে যোগ করেছি। '#' চিহ্নটা কোনো লাইনের গোড়াতেই থাকা মানে, সিস্টেমকে বলে দেওয়া, এই লাইনটা তোমাকে পড়তে হবেনা। এটা শুধু জ্যান্ত ইউজারদের পড়ার। আপনি যাতে বুঝতে পারেন, ফাইলে কী আছে। লাইনটা পড়ে দেখুন, শেষদুটো এন্ট্রি 'dump' আর 'fsck' বাদ দিয়ে আর সবগুলোই বুঝতে পারবেন। 'fsck'-টাও, যেন একটু চেনাচেনা লাগবে, কোথায় যেন দেখেছিলেন? দেখেছেন একাধিকবার, সাত নম্বর দিনের ৪.২ নম্বর সেকশনে, আজকের ৩.১ নম্বর সেকশনে, নানা প্রসঙ্গে। এত খারাপ মেমরি নিয়ে চলেন কী করে? হাতে একটু টাকা এলে সিস্টেম আপডেট করে নিন। আর একদম শেষ তিন লাইন আপনি আপাতত গয়ায় পিণ্ডদান করে দিন, ইনস্টল করার সময়ে সিস্টেমই ওগুলো লিখেছে, এই পাঠমালাতে আসবে না ওদের আলোচনা, আর আপনার আপাতত কাজেও লাগবে না, যখন লাগবে নিজেই শিখে নিতে পারবেন। এইমাত্র যে 'mount' কমান্ডের আলোচনা করলাম আমরা তার '<options>' অংশগুলোকে এবার '`man mount`' করে মিলিয়ে নিন। দেখুন, প্রত্যেকটাই আপনি হুবহু বুঝতে পারছেন। শুধু 'mount'-এর না, সঙ্গে 'fstab'-এর ম্যানপেজটাও পড়ে নিন। একটু আগে '`mount /dev/hdb5 /arkive`' কমান্ড দিয়ে আমরা যে মাউন্ট করছিলাম '/dev/hdb5' নাবিককে, '/arkive' বন্দরে, সেই কমান্ডটা আমি ছোট করে '`mount /dev/hdb5`' বা '`mount /arkive`' দিতে পারতাম। তাতেও একই কাজ হত, কারণ 'mount' অন্য প্রয়োজনীয় তথ্যগুলোকে 'fstab' ফাইল থেকে নিজেই পড়ে নিত।

'/etc/fstab' ফাইলের প্রথম মানে 'device' স্তম্ভটা তো সহজেই বুঝতে পারছেন। পার্টিশন ডিভাইসগুলোর নাম। দ্বিতীয় ফিল্ড 'mountpoint' হল সেই ডিরেক্টরি যেখানে মাউন্ট করা হবে। আজকের আলোচনার দুই নম্বর মানে বাড়ির প্ল্যানের মত দেখতে মাউন্টের ছবিটার সঙ্গে মিলিয়ে নিন। এর প্রথম এন্ট্রিটা রুট ডিরেক্টরি মানে '/' হওয়াটাই

প্রথা। তৃতীয় ফিল্ড 'fs' মানে ফাইলসিস্টেম। তথ্যকে ডিরেক্টরি এবং ফাইলে লেখা/রাখা/পড়ার ব্যবস্থা। মানে, ইএক্সটি জেএফএস এক্সএফএস ফ্যাটথার্টিটু ইত্যাদি যাদের পিন্ডি আমরা আজকেই চটকালাম। এই স্তম্ভে দেখুন উইনডোজ পার্টিশন দুটোর ফাইলসিস্টেমের নাম 'vfat'। এটা হল উইন্ডোজ ফ্যাটথার্টিটুর গ্নু-লিনাক্স বিকল্প। উইন্ডোজের পার্টিশনগুলোকে গ্নু-লিনাক্স ওই সিস্টেম দিয়েই পড়ে। ফাইলের নিচের দিকে ৯/১০/১১ লাইনে তিনটে ডিভাইস, '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0', সিডি-রম, সিডি-বার্নার, ফ্লপি। এই তিনটেই কিন্তু লিংক ফাইল। নিজে '`ls -l`' করে দেখুন, আপনি তো জানেন, কী করে দেখতে হয়। আপনার সিস্টেমে যদি সিডি পোড়ানোর ড্রাইভ না-থাকে, তাহলে '/dev/cdwri'-টা থাকবে না। বা এগুলো অন্য কোনো নামেও থাকতে পারে। যেমন সুজে নিজে নিজে একটা বিচ্ছিরি লম্বা নাম দেয়, '/dev/cdrecorder', লাইন ভেঙে যায়, ফাইলটা দেখতে বিচ্ছিরি লাগে। এই নামটা কি কিউট না? আমার দেওয়া। এবং এই '/dev/cdrecorder' নামটাও সুজে বা অন্য কোনো ওএস-ই নিজে নিজে দেয়না। পাঁচ নম্বর দিনের ৬ নম্বর সেকশনে আমরা '/etc/lilo.conf' ফাইলটা নিয়ে আলোচনা করেছিলাম। তাতে একটা লাইন দেখুন, সুজে আর স্ল্যাকওয়ার দুটো অংশেই একবার করে আছে। 'append = "hdc=ide-scsi"'। সুজে লাইনটায় আরো একটা অংশ আছে 'splash=0', এই অংশটা ভুলে যান, সুজে ব্যবহার করলেই লাগবে কেবল। বুট করার সময় কাদাকাদা রোমান্টিক নীল রঙে পঞ্চশর ভস্ম গোটা স্ক্রিনে ছড়িয়ে দেওয়ার অভ্যেস আছে সুজের, ছবিতে ছবিতে ছয়লাপ করে দেওয়ার, সেইসব কায়দা বন্ধ করার আদেশ। আর 'append = "hdc=ide-scsi"' অংশটা দিয়ে সুজে আর স্ল্যাকওয়ার দুটো সিস্টেমকেই বলা আছে, '/dev/hdc' নামের যে ডিভাইসটা, যদিও সেটা আইডিই ডিভাইস, সেটাকে স্কাসি ডিভাইস হিশেবে পড়ো। আগেও বলেছি, এটাই স্কাসি-এমুলেশন, সিস্টেমে সিডি পোড়াতে চাইলে করতে হয়, সিডি-ড্রাইভটা যদি বার্নার-ড্রাইভ হয়। তখনই ওই '/dev/cdrecorder' লিংকটা সিস্টেমের ডিভাইস ফাইলে বানানো হয়, যা আপনার '/dev/hdc' ডিভাইসটাকে আঙুল দেখায়। এখন থেকে যখনই আপনি লিংকটায় কিছু করছেন, লিখছেন বা পড়ছেন, আসলে করছেন সেটা আপনার বার্নার ডিভাইসটাকে। আমরা তো এখন নিজেরাই লিংক বানিয়ে দিতে জানি, যেভাবে '/dev/cdwri' বানানো। '/etc/fstab' ফাইলের '/dev/cdrom', '/dev/cdwri' আর '/dev/fd0' এই তিনটে ডিভাইসেই দেখুন ফাইলসিস্টেম করা আছে 'auto'। মানে আপনি সিস্টেমের বিচারবুদ্ধির উপর ছেড়ে দিচ্ছেন। রিমুভেবল মানে বার-করে-ফেলা-যায়-এমন ডিভাইসে, সিডি ফ্লপি ইত্যাদি, এটাই চালু প্রথা। ম্যানপেজ পড়ে দেখুন আপনি এখানে জলিয়েট রকরিজ ইত্যাদিও করতে পারেন।

'fstab' ফাইলে এর পরের ফিল্ড 'options'। একটাই কথা বলার, সুস্বাস্থের জন্যে খেজুরের কোনো তুলনা হয়না, শুধু আমার মত হাইপারটেনস ডায়াবেটিক বৃদ্ধ হলে, অপর্যাপ্ত ব্যবহারের আগে একবার পোস্ট-পন্ড্রিয়াল চেক করে নেওয়া ভালো। পঞ্চম মানে 'dump' স্তম্ভটায় আসতে পারে ১ বা ০। ডাম্প হল একটা ইউটিলিটি সফটওয়ার বা ব্যবহারিক প্রোগ্রামের নাম। ব্যাক-আপ করার। এই স্তম্ভে ১ থাকা মানে ডাম্প সেই পার্টিশনকে ডাম্প করবে, ০ থাকা মানে করবে না, যখন আপনি '`dump`' চালাবেন। এবং বুট করার সময় সিস্টেম নিজেই পার্টিশনটাকে মাউন্ট করে নেবে যদি এখানে ১ থাকে, ০ থাকলে করবে না। আপনি নিজে 'dump' চালালে, তখন কোন পার্টিশনকে কী করতে হবে সেটা 'dump' এই 'fstab' ফাইল থেকে পড়ে নেবে। 'fstab' ফাইলের শেষ বা ষষ্ঠ ফিল্ড 'fsck'। এই ডিভাইসের ফাইলসিস্টেমকে চেক করতে হবে কিনা, তার হাল-হকিকত যাচাই করে নিতে হবে কিনা, এবং চেক করতে হলে, কার পরে কাকে চেক করা হবে সেইটাই বলে দিচ্ছে এই ফিল্ডের সংখ্যাগুলো। গ্নু-লিনাক্স বুট করার সময়, যে ডিভাইসে এই ফিল্ডের মান ০ করা নেই, তাদের চেক করে নেয়। কোনো দুর্নীতি ঘটেছে কিনা, কোথাও কোনো তথ্যকাঠামো ঘেঁটে গেছে কিনা, কোরাপশন হয়েছে কিনা ফাইলসিস্টেমে। এই ফিল্ডে দেওয়া সংখ্যাটা মেনে পরপর চেক করে সিস্টেম। যেমন, সচরাচর রুট পার্টিশনের থাকে ১। তাই, প্রথম চেক করে একেই। তারপর ২, তারপর ৩, এইভাবে। আপনার মেশিনে যখন গ্নু-লিনাক্স ইনস্টল হয়েছিল, তখনই প্রাথমিক 'fstab' ফাইলটাকে বানিয়ে '/etc' ডিরেক্টরিতে রেখেছিল সিস্টেম। প্রতিবার তার প্রয়োজন অনুযায়ী সেটাকে পড়ে। ফাইলে যদি কোনো বদল ঘটান, ইচ্ছে বা প্রয়োজন মত, সেই বদল সহ তাকে পড়বে সিস্টেম। হার্ডডিস্কের পার্টিশন-ছকে কোনো ছোটতম বদলও যদি করেন, লিখে দিতে হবে এই ফাইলে। মাউন্ট করার ছকে কোনো বদল যদি ঘটাতে চান, সেটাও লিখে দিতে হবে। এখানের ফাইলটায় দেখুন, চার থেকে সাত নম্বর লাইনে, 'options'-এ, 'rw,noauto,user,exec' অংশটা

আমার যোগ করা। 'rw' মানে ফাইল পড়া আর লেখা দুটোই যাতে করা যায়। 'noauto' মানে আপনা থেকে সিস্টেম মাউন্ট করবে না, আলাদা করে কমান্ড দিয়ে মাউন্ট করতে হবে। 'user' দিয়ে বোঝাচ্ছে, শুধু রুট না, অন্য কোনো ইউজারও মাউন্ট করতে পারবে। '/dev/hdb5' পার্টিশনে দেখুন, 'user' নেই, যাতে কখনো ভুলবশত এই পার্টিশনের কোনো ফাইল কেউ বদলে না-ফেলে। 'exec' মানে বাইনারি ফাইল চালানো যাবে। মাউন্টের প্রতিটি অপশানেরই ঠিক উল্টোটা হল সেই অপশানের আগে 'no' যোগ করা, যেমন 'exec' আর 'noexec'।

এবার আমার মেশিনের '/mnt/slackware/etc' ডিরেক্টরি থেকে আর একটা 'fstab' ফাইল। এটা স্ল্যাকওয়ারের। পড়ে দেখুন, কিছু বুঝতে পারেন কিনা। আগেরটার সঙ্গে তফাতগুলো কোথায় ঘটছে, এবং কেন।

```
#device      mountpoint       fs          options            dump  fsck
/dev/hdb2    swap             swap        defaults           0     0
/dev/hda6    /                reiserfs    defaults           1     1
/dev/hda1    /mnt/windows/c   vfat        noauto,user,rw     1     0
/dev/hda5    /mnt/windows/d   vfat        noauto,user,rw     1     0
/dev/hdb5    /mnt/arkive      reiserfs    noauto,user,ro     0     0
/dev/cdrom   /mnt/cdrom       iso9660     ro,noauto          0     0
/dev/cdwri   /mnt/cdwri       iso9660     ro,noauto          0     0
/dev/fd0     /mnt/floppy      auto        noauto,owner       0     0
devpts       /dev/pts         devpts      gid=5,mode=620     0     0
proc         /proc            proc        defaults           0     0
```

তাকান, এখানেও একটা '/' পার্টিশন আছে, কিন্তু তার ডিভাইসটা আগের ফাইল থেকে বদলে গেছে, এখন রুট পার্টিশন হয়েছে '/dev/hda6'। কারণ, স্ল্যাকওয়ার সিস্টেম যখন বুট করে, তার রুট পার্টিশন হয় ওটাই। তখন সেই রুট পার্টিশনে যেটা '/etc' ডিরেক্টরি, সেই ডিরেক্টরির '/etc/fstab' ফাইলটাকে তার স্বাভাবিক জায়গাতেই পাই। যাকে সুজে দিয়ে বুট করার পর পাই '/mnt/slackware/etc' ডিরেক্টরির ভিতর '/mnt/slackware/etc/fstab' ফাইল হিশেবে। সুজের 'fstab'-এর সঙ্গে মিলিয়ে দেখুন, সোয়াপ পার্টিশন একই আছে। সুজে আর স্ল্যাকওয়ার দুটো অপারেটিং সিস্টেমই এক সোয়াপ পার্টিশন ব্যবহার করে। যখন সুজে দিয়ে বুট করি তখন সুজে, যখন স্ল্যাকওয়ার দিয়ে বুট করি তখন স্ল্যাকওয়ার। এবং সুজের যেটা রুট পার্টিশন, '/dev/hdb3', সেটা এই 'fstab' ফাইলে উল্লিখিতই নেই, তার কারণটা তো আগেই বলেছি, স্ল্যাকওয়ারে যে কারনেলটা আমি এখন ব্যবহার করি তার মধ্যে এক্সএফএস ড্রাইভারটা নেই, তাই সে মাউন্ট করতে পারেনা ওই পার্টিশন। ঠিক সেই একই কারণে '/dev/hdb1' পার্টিশনটাও নেই। দুটো লাইন তাই কমে গেছে স্ল্যাকওয়ারের এই 'fstab' ফাইলে। অন্য তফাতগুলো নিজেই বুঝে নিতে পারবেন। স্ল্যাকওয়ার অনেক বিশুদ্ধতাপন্থী, তাই ডিফল্ট রকমে সে সিডিরমে শুধু 'iso9660' ফাইলব্যবস্থার মত করেই পড়ে, দরকার পড়লে রকরিজ বা জলিয়েট রকমে পড়ানো যায়। এই টুকিটাকিগুলো আপনাকে পড়তে হবে ডিস্ট্রিবিউশনের নিজের ডকুমেন্টেশন থেকে। খেয়াল রাখবেন, ডিস্ট্রো থেকে ডিস্ট্রোয় কিছু তফাত থাকেই।

৫।। একটু ছোট্ট কুপথ — ফাইল বদলানো

একটু কুপথে ঘুরে আসি, ডাইগ্রেস করে আসি, তারপর শুরু হবে আমাদের ফাইলসিস্টেম আলোচনার লাস্ট ল্যাপ। আগের সেকশনে বারবার বলছিলাম 'fstab' ফাইল বদলানোর কথা। একটা জিনিষ বদলে অন্য জিনিষ লিখে দিতে হবে। ইত্যাদি। কিন্তু লিখবেন কী করে? আমাদের আলোচনার শূন্য নম্বর দিনে আমরা প্রচুর কথা বলেছিলাম বিশুদ্ধ টেক্সট এডিটর আর ওয়ার্ড প্রসেসর নিয়ে। তারপর সেই আলোচনা আবার এসেছিল কমান্ড এডিটরের প্রসঙ্গে। এখন সেই কমান্ড এডিটরদের প্রয়োজন পড়বে। বিল জয়ের এক্স (ex) তথা ভিআই (vi) এডিটর আর রিচার্ড স্টলম্যানের ইম্যাক্সের (emacs) কথা মনে আছে? ভিআই এখন আরো উন্নত হয়েছে, এখন তার নাম ভিম (vim), ভিআই-ইমপ্রুভড, উন্নতি এবং বলশালীতা বোঝাতে দীর্ঘ-ঈকার লাগিয়ে ভীম বলা যায়। তাতে দ্বিতীয় পাণ্ডবের মত আগ্রাসী হয়ে ওঠেনা, র‍্যাম সিপিইউ ইত্যাদি রসদের খিদে ভীমের অত্যন্ত কম। এ ব্যাপারে আপনি লাগ-এর অরিজিতকে মেল করতে পারেন, ভীমগীতা গেয়ে ও বেশ নাম কিনেছে। আমি নিজে এই কাজে ইম্যাক্স ব্যবহার করি। ধরুন 'fstab' ফাইলটা বদলাতে হবে। '`emacs /etc/fstab`' কমান্ড দিয়ে ফাইলটাকে খুলি, যা দরকার এডিট করি। তারপর 'Ctrl-X, Ctrl-S' মেরে সেভ করি। 'Ctrl-X, Ctrl-S' মানে 'Ctrl' সুইচটা টিপে রেখে একবার 'X' সুইচটা

টেপা, তারপর একবার 'S' সুইচটা টেপা, তারপর 'Ctrl' সুইচটা ছেড়ে দেওয়া। এবং কাজ শেষ হলে 'Ctrl-X, Ctrl-C' মেরে বেরিয়ে আসি। এটা কিন্তু রুট হয়ে করতে হয়। সি শিখতে গিয়ে যে মকশো করার প্রোগ্রাম বানাতে হয়, সেগুলো লিখি এবং কম্পাইল করি ইম্যাক্সে। আমার কাছে এর খুব কোনো উদ্দেশ্য নেই, ডিপ্রেশন কাটানো ছাড়া। রান্না করা আর প্রোগ্রাম লেখা এ দুটোই দেখেছি খুব কাজ দেয়। তবে রান্না করতে গেলেই বাজার করা লাগে, যেটা বোর কাজ এবং খরচেরও। আর প্রোগ্রাম লিখতে কম্পাইল করতে চালাতে আলাদা কোনো খরচই নেই, টেবিলে বসেই করা যায়। ইম্যাক্সের কাছে এই কাজটুকু হল একটা তিমি মাছের পাখনা নাড়ানোর মত। কিন্তু ইম্যাক্স ঠিক ভাবে শেখাটা নিজেই বিরাট একটা কাজ। ইম্যাক্সে ঢুকে আপনি যদি 'Ctrl-H' মারেন তাহলে আপনি ইম্যাক্সের হেল্পে ঢুকতে পারবেন, ঢুকে পড়তে শুরু করুন, যখন যতটুকু দরকার। একসঙ্গে গোটাটা পড়া একমাত্র কোনো নিরক্ষরের পক্ষেই সম্ভব, শুধু পেজ ডাউন করে গেলেই হয়। সুন্দর একটা টিউটোরিয়াল হয় ইম্যাক্সের, তাতে ঢোকার উপায় দেখুন ও নিজেই বলে দেবে, '`emacs`' দিয়ে এন্টার মারুন। এই ভিম আর ইম্যাক্স ছাড়াও জো (joe) আছে, সাতকেলে এড (ed) আছে। এদের শেখা শুরু করার জন্যে ম্যানপেজ আছে। ইনফো (`info`) আছে।

এবার ধরুন, আপনি আপনার সিস্টেমে দেখলেন কোনো কোনোটা নেই, এদিকে আপনি সেটা দিয়ে কাজ করতে চান। তখন আপনার ডিস্ট্রিবিউশন সিডি, যা দিয়ে আপনি ইনস্টল করেছিলেন, সেটায় খুঁজে দেখুন। এরপরেও হয়ত দেখলেন নেই, বা যে কাজটা করার জন্যে যে প্যাকেজের যে ভার্শনটা চাইছেন সেটা নেই। একটু আগে যেমন 'rman'-এর কথা বললাম, সেটা সব ডিস্ট্রোয় থাকেনা। বা সমগ্র হাউটু-টা (howto)। বা অর্বুদ অর্বুদ গ্নু-লিনাক্সীর প্রাণের চেয়ে প্রিয় এমপ্লেয়ার। সেগুলো কী করে পাবেন? নেটে যদি পেতে চান, তার একটা ভালো জায়গা 'www.freshmeat.net'। বা 'www.sourceforge.net' । এই সাইটে গিয়ে আপনার সাধের প্যাকেজটার নাম দিয়ে সার্চ দিন। তারপর হাইপার লিংক দিয়ে সেটাকে ডাউনলোড করুন। এই ইনস্টলার ফাইলগুলো সচরাচর '*.rpm' বা '*.tar.bz2' বা '*.tar.gz' বা '*.tgz' হয়।

এই আরপিএম ব্যাপারটা চালু করেছিল রেডহ্যাট, আরপিএম এসেছে রেডহ্যাট প্যাকেজ ম্যানেজমেন্ট কথাটা থেকে। পরে এই ধরনের ফরম্যাট ম্যানড্রেক বা সুজে বা অন্য ডিস্ট্রোরাও অনেকেই ব্যবহার করে। এই '*.rpm' ফাইলগুলো ইনস্টল করার কমান্ড '`rpm -i`'। 'rpm' কমান্ডের ম্যানুয়াল পড়ে নিন। যেমন আপনি রুট হয়ে কমান্ড দিতে পারেন '`rpm -Uvh mundu.rpm`'। এটা আপনার সিস্টেমে 'mundu.rpm' প্যাকেজটাকে ইনস্টল করে দেবে, মানে 'mundu' নামের প্রোগ্রামটাকে, আগেই ইনস্টল করা থাকলে আপডেট করে দেবে। যারা একটা কার্যোপযোগী মুণ্ডের অভাবে ভোগেন, 'mundu.rpm' প্যাকেজটা ইনস্টল করে নিলে তাদের খুবই কাজ দেয় বলে শুনেছি। '*.tgz' হল স্ল্যাকওয়ারের নিজের প্যাকেজ বানানোর স্টাইল। অন্য ডিস্ট্রোতেও ব্যবহার করা যায়। একধরনের প্যাকেজ থেকে অন্য ধরনের প্যাকেজে বদলে নেওয়ার একটা ভালো প্রোগ্রাম আছে, তার নাম '`alien`'। '*.deb' হল ডেবিয়ান ডিস্ট্রোর প্যাকেজ বানানোর রকম। যে ডিস্ট্রো আপনি ব্যবহার করছেন তার থেকে ভালো করে পড়ে নিন।

এই যে একটা প্রোগ্রাম আপনি ইনস্টল করছেন, এর মানে কী? প্রোগ্রামটার জন্যে প্রথমে লেখা হয়েছিল কিছু কোড। কিন্তু কোড ফাইলটা একটা টেক্সট ফাইল। সে নিজে চলেনা। কম্পাইলার দিয়ে কোড ফাইলটা কম্পাইল করে তৈরি হয় বাইনারি বা এক্সিকিউটেবল ফাইল। মানে, যে ফাইল চালানো যায়। প্রাথমিক কোডকে কম্পাইল করে তৈরি করে বাইনারিটাকে বলে প্রিকম্পাইলড বাইনারি। এই আরপিএম ব্যাপারটা হল তাই। এর মধ্যে একটা প্রিকম্পাইলড বাইনারি থাকে। তাকে তার সঠিক জায়গায় রেখে দেওয়া, এবং তার প্রয়োজনীয় লাইব্রেরি ফাইলগুলোর সঙ্গে লিংক করে দেওয়াই মূলত কাজ একটা আরপিএম প্যাকেজের। অন্যদিকে, গ্নু-লিনাক্স সিস্টেমে জিসিসি কম্পাইলারটা যেহেতু এমনিতেই দেওয়া আছে, তাই, মূল কোডটাকে নিজের মেশিনে চালিয়ে কম্পাইল করে নেওয়াই যায়। জিসিসি (gcc) মানে একটা কম্পাইলারের একটা গোটা সংগ্রহ। জিসিসি দিয়ে মূল কোডটাকে কম্পাইল করে নিতে পারেন। কোডগুলো সচরাচর দেওয়া থাকে '*.tar.bz2' বা '*.tar.gz' ফাইলে। '*.tar.*' মানে সিন্দুক ফাইল, টার প্যাকেজ দিয়ে বানানো, আগেই বলেছি। 'bz2' মানে তাকে কোঁকড়ানো হয়েছে 'bzip2' প্রোগ্রাম দিয়ে, আর 'gz' মানে কোঁকড়ানো হয়েছে 'gzip' দিয়ে। ম্যানপেজ পড়ে নিন। নানা ভাবে এদের নিয়ে কাজ করা যায়, ইউনিক্স তথা গ্নু-লিনাক্সের দর্শনই তাই, কোনো একটা কাজ সবসময়েই একাধিক ভাবে করা যায়। 'tar' দিয়ে সরাসরি এই 'bz2' বা 'gz' ফরম্যাটে বানানো কোঁকড়ানো সিন্দুক খোলার উপায় আছে। ধরুন ওই 'mundu.rpm' এবার আপনি আর 'rpm'

মানে প্রিকম্পাইলড বাইনারিতে না-নামিয়ে কোঁকড়ানো সিন্দুকে নামিয়েছেন, 'mundu.tar.bz2'। আগেরবার ইনস্টল করার পর থেকে বহুত গন্ডগোল করছিল, আলো অন্ধকারে যেখানেই যান, আপনার নৃমুণ্ডের আবছায়া তৈরি হচ্ছিল, এবার আপনি আর মুন্ডু নিয়ে ওইসবে রিস্কে যেতে চাননা।

আপনার মুন্ডু যাতে আপনার হার্ডওয়ারের সঙ্গে খাপে খাপে মিলে যায়, আপনার হার্ডওয়ারের সফটওয়ারের প্রতিটি খুঁটিনাটি নিয়ে আপনার সিস্টেমেই মুন্ডুর কোড-ফাইল কম্পাইল করে নিতে চাইছেন। এমনকি খুঁটিনাটি আপনার নিজে দিতেও হবেনা, এই কোড প্যাকেজগুলোর ভিতরে 'configure' আর 'make' বলে দুটো ব্যবস্থা করাই আছে, তারা নিজেরাই আপনার সিস্টেমের খুঁটিনাটি অনুযায়ী ইনস্টলেশনটা তৈরি করে নেবে। আপনার নামানো কোড-প্যাকেজটা যদি 'mundu.tar.bz2' হয় তাহলে এবার কমান্ড দিন, '`tar xvjf mundu.tar.bz2`'। এতে 'mundu.tar.bz2' ফাইলটা কোঁকড়ানো টার-সিন্দুক অবস্থা থেকে বেরিয়ে একটা স্বাভাবিক একটা 'mundu' নামের ডিরেক্টরি তৈরি হয়ে গেল, যে ডিরেক্টরিতে দাঁড়িয়ে আপনি কমান্ডটা দিয়েছেন, তার মধ্যে। এই 'mundu' ডিরেক্টরির মধ্যে মুন্ডু ইনস্টল করার প্রয়োজনীয় কোড, 'configure', 'make' সবই আছে। যদি ফাইলটা 'mundu.tar.gz' হত তাহলে আপনার কমান্ডটা সামান্য একটু বদলে যেত, হত '`tar xvzf mundu.tar.gz`'। এতেও ওই একই ভাবে আপনি 'mundu' বলে একটা ডিরেক্টরি পেতেন, যার মধ্যে ফাইলে বা সাবডিরেক্টরিতে ভরা থাকত আপনার প্রয়োজনীয় কোডগুলো।

এবার ওই 'mundu' ডিরেক্টরিতে ঢুকে আপনি পাবেন একটা 'README' ফাইল বা একটা 'INSTALL' ফাইল বা এই দুটোই। এদের 'less' দিয়ে পড়ে নিলেই আপনি জানতে পারবেন কী করতে হবে। করাটাও এমন হাতি ঘোড়া কিছু না। এরা প্রত্যেকেই ব্যবহার করে জিসিসির মতই ওই অন্য পরশপাথরটা, যার নাম মেক (make)। এই 'make' যাতে কাজ করতে পারে তার জন্যে একটা 'makefile' দেখুন রয়েছে ডিরেক্টরিতে। এবার আপনাকে কমান্ড দিতে হবে, '`./configure`'। এই '.' মানে কারেন্ট ডিরেক্টরি, যে ডিরেক্টরিতে এই মুহূর্তে আপনি আছেন, সেখানে থাকা 'configure' প্রোগ্রামটা চালাও। এই 'configure' প্রোগ্রামটা চালিয়ে ইনস্টল হওয়ার আগে ওই সফটওয়্যার প্যাকেজ এবার সিস্টেমের প্রত্যেকটা প্রয়োজনীয় খুঁটিনাটি কনফিগারেশন জেনে নেবে। তারপর, কনফিগারেশনের কাজ শেষ হলে, কমান্ড প্রম্পট ফেরত এলে কমান্ড দেবেন '`make`'। অনেকসময় কনফিগার প্রোগ্রামটা থাকেনা, সরাসরি মেকফাইল থাকে। তখন মেক দিয়েই কাজ শুরু করতে হয়। এতে ওই রাশিরাশি কোডের ঠিক ঠিক প্রয়োজনীয় অংশ 'make' আপনার সিস্টেমের মানানসই রকমে সাজিয়ে দেবে। এবার 'make' শেষ হলে রুট হয়ে কমান্ড দেবেন '`make install`'। বা, '`./configure && make && make install`' কমান্ড দিয়ে এই গোটা কাজটাই একবারে করে নেওয়া যেত। এই '`&&`' অংশটা ব্যাশকে বলে দিত, আগের কাজটা শেষ হলেই কেবল পরের কাজটায় যেও। যদি এটা একটু ভালো করে বুঝতে চান, সরাসরি ব্যাশের ম্যানপেজ থেকে মিলিয়ে নিন। এরপর বেশ কিছুটা সময় ধরে কনসোলের পর কনসোল জুড়ে রাশিরাশি প্রোগ্রামিং কচকচি ফুটে উঠতে থাকবে। তারপর কমান্ড প্রম্পট ফেরত আসবে। মানে, কম্পাইলেশন হয়ে গেল। এখুনি যান, তাদের প্রত্যেককে একটা করে মেল করে দিন, যারা এতদিন ধরে ব্যাঙ্গবিদ্রূপ করে আসছে যে আপনার কোনো মুন্ডু নেই। মুন্ডু হাত পা যা যা খুশি এভাবে নেট থেকে নামান আর ইনস্টল করে নিন। শুধু টেলিফোন বিলটা মাথায় রাখলেই হল। এবার আমরা যাব আমাদের সেই দীর্ঘ প্রতীক্ষার গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কিতে। শুধু একটা কথা বলতে ভুলে গেছি, ৪ নম্বর সেকশনে, সেটা হল, ওই 'mount' কমান্ডটার একটা বিপরীত কমান্ড আছে 'umount', যা মাউন্টের ঠিক উল্টোটা করে, মানে আনমাউন্ট। আদতে কমান্ডটার নাম ছিল 'unmount', ১৯৭০ অব্দি ইউনিক্স জগতে এই চেহারাতেই তাকে পাওয়া যেত, তার পর কবে 'n'-টা খসে গেছে, দীর্ঘ কঠোর তদন্তের পরেও জানা যায়নি 'n'-এর এন্ড ঘটল কিভাবে, কোন আততায়ীর হাতে।

৬।। গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কি

আগের দুদিন থেকে আজ এই দীর্ঘ আলোচনার মধ্যে দিয়ে আমাদের কাছে হার্ড ডিস্কের ভৌত এবং মূর্ত ভূমি, সেখানে তথ্য তথা ফাইল এবং ডিরেক্টরি ইত্যাদি নানা জাতের তথ্য-সমাহার রাখার প্রক্রিয়া, আর সেই তথ্যকে একটা কেজো চলন্ত কম্পিউটারের কাজের সম্পর্কে মিলিয়ে নেওয়ার জন্যে ফাইল আর ডিরেক্টরি দিয়ে সাজানো একটা

বিমূর্ত ভূমি — এই তিনের আলাদা অস্তিত্ব এবং একই সঙ্গে অঙ্গাঙ্গী সম্পর্ক — এই গোটাটাই এখন আমাদের কাছে পরিচিত। মূর্ত ভূমিটা হল সেই কাঁচা ডিস্ক যা আমরা কিনে আনি। আর রুট ডিরেক্টরি '/' থেকে '/home', '/bin', '/etc' ইত্যাদি সাবডিরেক্টরিতে বিন্যস্ত কেজো কম্পিউটার ভূমিটা হল বিমূর্ত জায়গা যা আমাদের তৈরি করে নিতে হয়। বিমূর্ত এবং সাঙ্কেতিক, বহুবার বলেছি আমরা, অ্যাবস্ট্রাক্ট এবং সিম্বলিক। এই জায়গাটা অত্যন্ত গুরুত্বপূর্ণ, কারণ যে প্রোগ্রামগুলো নিয়ে আমরা কাজ করি তারা কেউ কাজ করে একদম ফাইল নিয়ে, যেমন ধরুন একটা 'emacs' বা 'mkdir', আবার কেউ কাজ করে একদম কাঁচা ভৌত ভূমি নিয়ে যেমন 'fdisk' বা 'fsck'। আমরা যে প্রোগ্রামগুলোকে চিনি এবং আমাদের কম্পিউটার সময়ের প্রায় গোটাটাই যাদের নিয়ে কাজ চলে তারা প্রায় সবাইই কাজ করে বিমূর্ত ভূমিটায়। কিন্তু এই বিমূর্ত ভূমিটা দাঁড়িয়ে আছে তথ্য আর তথ্য-সমাহারের অনেকগুলো প্রক্রিয়ার উপর, যার সম্মিলিত নাম ফাইলসিস্টেম, কাঁচা ডিস্কভূমিতে তথ্য তথা ডিরেক্টরি এবং ফাইল লেখা/রাখা/পড়ার প্রক্রিয়া এবং রকম। ফাইলসিস্টেমের উপর দাঁড়িয়েই এরা কাজ করে, তাই ফাইলসিস্টেমের ভৌত বাস্তবতার সঙ্গে প্রতি মুহূর্তে মিলিয়ে না-নিলে গোটা প্রক্রিয়াটাই বোঝা সম্ভব না।

কাজের ওই বিমূর্ত ভূমিটার সঙ্গে আমরা এখন অনেকটাই পরিচিত। যাকে আমরা ডেকেছি ঐক্যবদ্ধ হায়েরার্কিন্যস্ত ফাইলসিস্টেম। ডিরেক্টরি সাবডিরেক্টরি ফাইলের একটা ক্রমানুসারী সমগ্র, যা তৈরি হয়েছে বিভিন্ন পার্টিশনে তথ্য রাখার বিভিন্ন প্রক্রিয়া মানে বিভিন্ন ফাইলসিস্টেম নিয়ে, একটা কেজো চালু জ্যান্ত ওএস যাকে ঘিরে কাজ করে। এখন থেকে আগের বাক্যে ফাইলসিস্টেম কথাটার দ্বিতীয় ব্যবহারটাকে আমরা আপাতত ভুলে যাব। যেখান থেকে শুরু করেছিলাম ছয় নম্বর দিনে, সেখানে ফেরত যাব। ফাইলসিস্টেম মানে ডিরেক্টরি আর ফাইলের ওই ঐক্যবদ্ধ সমগ্রটা। সেটার হায়েরার্কিটাকেও চিনি আমরা। রুট ডিরেক্টরি মানে '/' থেকে শুরু। তার মধ্যে নানা ডিরেক্টরি। তাদের মধ্যে আবার ডিরেক্টরি, আবার, এই ভাবে চলতেই থাকে। প্রতিটি ডিরেক্টরিতেই থাকতে পারে আরো নিচের ডিরেক্টরি, মানে সাবডিরেক্টরি, যতখুশি, এবং যতখুশি ফাইল। এবার, এই ক্রমটা আছে বলেই, ঐক্যটা আছে বলেই, একটা ফাইলসিস্টেমে প্রতিটা ফাইল এবং ডিরেক্টরির একটা ঠিকানা আছে — একটা বিমূর্ত ঠিকানা, রুট ডিরেক্টরি থেকে ঠিকানাটা শুরু হয়ে শেষ হয় যে ডিরেক্টরিতে ফাইলটা আছে সেই ডিরেক্টরির নামে এসে (মূর্ত ঠিকানাটা, খেয়াল করুন, আর আমাদের আপাতত দরকার পড়বেনা, মানে কত নম্বর সিলিন্ডারের কত নম্বর সেক্টরের কত নম্বর ব্লক ইত্যাদি)। যেমন ধরুন '/mnt/slackware/etc/fstab' ফাইলটার ঠিকানা হল '/' ডিরেক্টরির মধ্যে 'mnt' ডিরেক্টরির মধ্যে 'slackware' ডিরেক্টরির 'etc' ডিরেক্টরির মধ্যে 'fstab' নামের ফাইল। আমি এবং আমার জ্যান্ত ওএস দুজনেই এটাকে চিনে নিতে পারব। এবার আমরা গ্নু-লিনাক্সে ফাইলসিস্টেম হায়েরার্কির ছকটা বুঝব। কেন গ্নু-লিনাক্সের প্রতিটি ডিস্ট্রোতেই একটা মোটামুটি একরক ডিরেক্টরি কাঠামো পাই। এর দুটো চমৎকার ডকুমেন্ট আছে, রাস্টি রাসেল আর ড্যানিয়েল কুইনলানের, 'ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ড', অন্যটা বিন এনগুয়েনের, 'লিনাক্স ফাইলসিস্টেম হায়েরার্কি', পরেরটা লিনাক্স ডকুমেন্টেশন প্রোজেক্ট 'www.tldp.org' থেকে।

ছকটা একদিনে গড়ে ওঠেনি। '৯০-এর দশকের মাঝামাঝি অব্দি, প্রতিটি গ্নু-লিনাক্স ডিস্ট্রো ছকটা বানিয়ে নিত নিজের মত করে। এতে সমস্যা এই যে, একটা ছকে অভ্যস্ত কেউ অন্য ছকে কাজ করতে গেলে ছক বোঝাটাই তার প্রথম কাজ হয়ে দাঁড়ায়। আরো সমস্যা অবলা কোডগুলোর। তাদের বানানো হয়েছে একটা ছকের সঙ্গে মিলিয়ে। এই ডিরেক্টরিতে খোঁজো এই ফাইলটা আছে কিনা, ওই ডিরেক্টরিতে খোঁজো ওই ফাইলটা আছে কিনা, ইত্যাদি। তারা অন্য ছকের কোনো ডিস্ট্রোতে কাজই করবে না। প্রতিটা ডিস্ট্রোর আলাদা প্রোগ্রাম বানাতে হবে। মনে করুন, ইউনিক্স জগতে পোজিক্স আসার প্রসঙ্গটা। পোর্টেবিলিটি বা স্থানান্তরযোগ্যতা। স্থানান্তরযোগ্যতার দিকে তাকিয়েই তৈরি হল একটা ডকুমেন্ট, তার নাম 'লিনাক্স ফাইলসিস্টেম স্ট্রাকচার'। আর ডকুমেন্টটা বানানোর জন্যে তৈরি হল একটা গ্রুপ। এই ডকুমেন্ট আর গ্রুপ দুটোকেই ডাকা হয় এফএসস্ট্যান্ড (FSSTND), ফাইলসিস্টেম-স্ট্যান্ডার্ড। এই কাজটাই বাড়তে বাড়তে পরে ফাইলসিস্টেম হায়েরার্কি স্ট্যান্ডার্ডে পৌঁছয়। কাজটা এখনো সম্পূর্ণ নয়, অনেক না-মেটা মনান্তর আছে। যেসব শেলস্ক্রিপ্ট কোনো একটা বিশেষ রকম মেশিনের সঙ্গে সম্পৃক্ত নয়, যে কোনো মেশিনেই চালানো যায়, তারা কি '/usr/share' ডিরেক্টরিতে থাকবে, '/usr/share' ডিরেক্টরি কি আরো ডিরেক্টরিতে ভাঙা হবে, ইত্যাদি। ডিস্ট্রো থেকে ডিস্ট্রো-তে কিছু তফাত তো আছেই। যেমন, বহিরাঙ্গিক পার্টিশনগুলো, আমার মেশিনে ওই চারটে পার্টিশন, '/dev/hdb5', '/dev/hda6', '/dev/hda1', '/dev/hda5', এরা কোন কোন মাউন্টপয়েন্টে মাউন্ট হবে সেটা

ডিস্ট্রো থেকে ডিস্ট্রোতে বদলায়। এইরকম আরো আছে। একাধিক ডিস্ট্রোয় কাজ করতে গেলেই দেখবেন। এবং এখন আমরা যে ছকটা দেব সেটা যে প্রত্যেকটা ডিস্ট্রো হুবহু মেনে চলে তা কিন্তু নয়। ছকটা একটা মোটামুটি দিশা দেয়।

৬.১।। রুট ডিরেক্টরি বা '/'

এফএসস্ট্যান্ড মানার জন্যে রুট ডিরেক্টরি বা স্ল্যাশ বা '/'-এর মধ্যে কিছু ডিরেক্টরি থাকতেই হবে। এমনকি হুবহু যদি এই ডিরেক্টরিগুলোর কোনোটা নাও থাকে, তখন সেই নামের কোনো সিম্বলিক লিংক বানিয়ে এখানে রাখতেই হবে। ধরুন, আপনি আগে যে ডিস্ট্রোটা ব্যবহার করতেন সেটায় '/lib' বলে একটা লাইব্রেরি ডিরেক্টরি পেতেন। এই ডিস্ট্রোয় ওই ফাইলগুলো হয়ত রাখা আছে '/usr/lib' ডিরেক্টরিতে। এখন আপনার যাতে দিশাহারা অবস্থা না-হয়, সেই জন্যে হবে। এটা যাতে না-হয়, তাই, '/' ডিরেক্টরিতে 'lib' নামে '/usr/lib' ডিরেক্টরির একটা সিম্বলিক লিংক বানিয়ে রাখতে হবে। আপনি এখনো যাতে একই ভাবে '`cd /lib`' করে সেইখানে পৌঁছতে পারেন, যেখানে আপনার কাঙ্খিত ফাইলগুলো আছে। সিম্বলিক লিংক তো তাই করে। নিচের তালিকাটা হল রুট ডিরেক্টরির ন্যূনতম বাসিন্দা। এরা থাকবেই। এদের বাইরে আদতে আরো বহু কিছু থাকে বা থাকতে পারে।

/bin — বিন ডিরেক্টরিতে থাকে অত্যাবশ্যক কমান্ড বাইনারিগুলো, যাদের কমান্ড প্রম্পটে টাইপ করে তারপর এন্টার মেরে আমরা চালাই। আমরা কমান্ড প্রম্পটে কাজ করার জন্যে যে আদেশগুলো নিয়ে এই পাঠমালায় আলোচনা করেছি, তারা প্রায় সকলেই থাকে এই ডিরেক্টরিতে। '/bin'-এর বিশদ আলোচনায় আসছি।

/boot — বুট ডিরেক্টরিতে থাকে সিস্টেম বুট করার জন্যে প্রয়োজনীয় কারনেল ইমেজ '/boot/vmlinuz' ইত্যাদি এবং তার কিছু আনুষঙ্গিক ফাইল। তারা যদি নাও থাকে, থাকে তাদের সিম্বলিক লিংক। কারনেল কাজ করতে গেলে যে মডিউলগুলো লাগবে সেই তালিকাও দেওয়া থাকে এখানে। বুট-প্রক্রিয়ার কনফিগারেশন ফাইলগুলো এখানে থাকেনা, বলেছি আগেই, 'lilo'-র কনফিগারেশন ফাইল 'lilo.conf' থাকে '/etc' ডিরেক্টরিতে। '`lilo`' কমান্ড দিয়ে ইনস্টল হয় যখন, কিছু ফাইল তৈরি হয় বুট ডিরেক্টরিতে। যেমন '/boot/boot.0300' মানে মাস্টার বুট রেকর্ডের ব্যাকআপ, বা '/boot/boot.b' মানে মূল বুট সেক্টর বা তার সিম্বলিক লিংক, দুটোই 'lilo'-র তৈরি।

/dev — ডেভ ডিরেক্টরিতে থাকে রাশিরাশি ডিভাইস ফাইল, যাদের নিয়ে যথেষ্ট আলোচনা হয়েছে।

/etc — ইটিসি বা এটসেট্রা ডিরেক্টরিতে থাকে একটা সিস্টেমের হার্ডওয়ার এবং সফটওয়ার সংক্রান্ত প্রায় সমস্ত কনফিগারেশন ফাইল। এর কিছু ফাইলকে এর মধ্যেই বেশ ধরে ধরে পড়েছি আমরা, বারবার। সিস্টেমের স্নায়ুকেন্দ্র বলা চলে এই '/etc' ডিরেক্টরিকে। আমরা এর কথায় পরে আসছি।

/lib — লিব ডিরেক্টরিতে থাকে লাইব্রেরি ফাইল, এর কথা আগেই বলেছি, প্রোগ্রামগুলো এই লাইব্রেরিদের কাজে লাগায়। প্রতিটি প্রোগ্রামের জন্যে আলাদা আলাদা ফাংশন বা ক্রিয়া না লিখে, সব প্রোগ্রামের জন্যে একত্রে এই লাইব্রেরি ফাংশনগুলোকে তুলে রাখা হয় লিব ডিরেক্টরিতে। আর থাকে কারনেল মডিউলগুলো, মানে, কারনেলের চলার জন্যে ছোট ছোট টুকরো প্রোগ্রাম বানানো থাকে, এক ধরনের ড্রাইভার, এক এক ধরনের উপাদানের জন্যে। ধরুন একটা বিশেষ সাউন্ডকার্ড আপনি ব্যবহার করছেন, তার জন্যে কারনেলে একটা বিশেষ মডিউল লাগিয়ে নিতে হচ্ছে। আবার সাউন্ডকার্ডটা বদলালেন, মডিউলটা বদলে নিতে হল।

/mnt — এমএনটি বা মাউন্ট ডিরেক্টরি হল বহিরাগত পার্টিশনদের বসতে দেওয়ার জেনেরাল শীতলপাটি। সব ডিস্ট্রো এটা একই ভাবে মানেনা। ম্যানড্রেক স্ল্যাকওয়ার খুব মানে। কিন্তু সুজে নানা অন্য অন্য জায়গা বানিয়ে দেয়। যেমন আমার উইনডোজ ডিরেক্টরি দুটোর জন্যে নিজে থেকেই বানিয়ে দিয়েছিল '/data1' আর '/data2' বলে দুটো ডিরেক্টরি, একদম রুট ডিরেক্টরিতে। পরে নিজে বদলে নিতে হয়। তবে খেয়াল রাখবেন, এইসমস্ত যাবতীয় জিনিষ বদলানোর পরেই '/etc/fstab' ফাইল থেকে মিলিয়ে নিন, ঠিক আছে কিনা। নিজের মত মাউন্টপয়েন্ট বানিয়ে নেওয়া যায় '/mnt' ডিরেক্টরিতে, '/mnt/arkive' বা '/mnt/slackware' ইত্যাদি। আর চালু কিছু মাউন্টপয়েন্ট এখানে এমনিতেই থাকে, যদি ডিস্ট্রো সেখানে নিজস্ব কেরদানি না-মারে। যেমন '/mnt/cdrom' বা '/mnt/floppy' ইত্যাদি। সুজেতে সেসব থাকে '/media' ডিরেক্টরির মধ্যে।

/opt — অপশনাল বা ঐচ্ছিক, ডিস্ট্রোর ডিফল্ট ইনস্টলেশনের নিজস্ব কাঠামোর বাইরে থেকে যুক্ত প্যাকেজদের জায়গা। যেমন ওপেনঅফিস, মোজিলা ইত্যাদি প্যাকেজগুলো এখানে থাকে। এফএসস্ট্যান্ড অনুযায়ী, সমস্ত বহিরাগত প্যাকেজদেরই এখানে থাকার কথা। আর এই প্যাকেজটা ইনস্টল করার জন্যে আপনার যে আনুষঙ্গিক ফাইলগুলো থাকবে তাদের থাকার জায়গা এই '/opt' ডিরেক্টরির মধ্যেই ওই প্যাকেজের নামে আলাদা একটা ডিরেক্টরিতে। ধরুন ওই 'mundu' প্যাকেজটা, বাইরে থেকে যোগ করা প্যাকেজ, কারণ, ডিফল্টে মানুষের একটা মুন্ডু এমনিতেই থাকার কথা, আপনি যদি ইনস্টল করেন, সেই ইনস্টলেশনটা ঘটবে '/opt/mundu' ডিরেক্টরির মধ্যে। ইনস্টল করতে গিয়ে যা যা বাড়তি ফন্ট মানে অক্ষর-আকার লাগবে, বা ছবি, বা ডেটাবেস, সে সবই থাকবে এই ডিরেক্টরির ভিতর। এই প্যাকেজ যে বাইনারি ফাইল তৈরি করবে সেটা থাকার কথা '/opt/bin' ডিরেক্টরিতে, ডকুমেন্টেশন থাকার কথা '/opt/doc' ডিরেক্টরিতে, ইত্যাদি। তবে, বেশিরভাগ প্যাকেজই এতসব খাবলা দায়িত্ব পালন করেনা।

/sbin — এর মধ্যে থাকে অত্যাবশ্যক সিস্টেম বাইনারি। গ্নু-লিনাক্সে বাইনারি বা এক্সিকিউটেবল মানে চালানো যায় এমন ফাইলগুলোর দুটো ভাগ আপনারা চেনেন। একটা সাধারণ বাইনারি, যে কোনো ইউজারের ব্যবহারযোগ্য। আর অন্যগুলো হল ভিআইপি বাইনারি। সিস্টেম অ্যাডমিনিস্ট্রেশন বা পরিচালনার বিশেষ বাইনারিগুলো থাকে এই '/sbin' ডিরেক্টরিতে। এর চেয়ে একটু কম গুরুত্বপূর্ণ সিস্টেম বাইনারিরা থাকে '/usr/sbin' ডিরেক্টরিতে বা '/usr/local/sbin' ডিরেক্টরিতে। এই '/sbin' ডিরেক্টরি তথা সিস্টেম বাইনারিদের কথায় আসছি আমরা।

/tmp — অস্থায়ী বা টেম্পোরারি ফাইলদের জায়গা। একটা নির্দিষ্ট সময়ের পরে যাদের আর দরকার পড়বেনা। অনেক প্রোগ্রাম কাজ করার সময় বিশেষ কিছু ফাইল বা ডিরেক্টরি বা প্রোগ্রামকে লক করে দেয়, চলার সময়টুকু জুড়ে অন্য কেউ তাদের স্পর্ষ করতে পারে না। কিম্বা চলার কাজে কিছু অস্থায়ী তথ্য তৈরি করে, প্রোগ্রাম চলা শেষ হয়ে যাওয়ার পরে যাদের আর লাগবে না। এই সবকিছুকে অস্থায়ী ভাবে রাখা হয় এই '/tmp' ডিরেক্টরিতে। '`su`' করে রুট হয়ে নিয়ে এখান থেকে ফাইল ওড়ানোই যায়, কিন্তু কদাচ নয়, যদি-না, হুবহু জানা থাকে, ঠিক কী করছি। সেই মুহূর্তে যে প্রসেসগুলো চলছে সিস্টেমে তাদের অনেকেরই অত্যন্ত জরুরি মালপত্তর এখানে থাকে। গোটা সিস্টেম হ্যাং করে যেতে পারে ক্র্যাশ করে যেতে পারে — গ্নু-লিনাক্স সিস্টেমে যা সত্যিই একটা দুর্ঘটনা, এত কম ঘটে। সত্যিই কম। হাতে গোনা যায়। ওপনঅফিস ১.০ ভার্শনে আমার এনভিডিয়া ড্রাইভারের সঙ্গে কিছু একটা গোলযোগ হচ্ছিল, পরে, এনভিডিয়া ড্রাইভারের একটা পুরোনো ভার্শন ব্যবহার করে যেটা চলে গেল, সেই সময়ের গোটা পাঁচেক হ্যাং আজ পর্যন্ত এই দু-বছরে আমার গ্নু-লিনাক্স সিস্টেমের পাঁচমাত্র হ্যাংদোলন। নিজে লিখতে গিয়ে তার আগের উইনডোজের বছরগুলোর সঙ্গে মিলিয়ে নিজেরই কেমন অবিশ্বাস হচ্ছে। এই '/tmp' ডিরেক্টরির মোট ফাইলের আয়তন সচরাচর কয়েক কিলোবাইটের বেশি হয়না। বুট বা শাটডাউনের সময় সচরাচর এই ডিরেক্টরিকে ফাঁকা করে দেওয়া হয়।

ডিরেক্টরি	সুজে	স্ল্যাকওয়ার
/bin	7.2M	6.2M
/boot	6.7M	2.9M
/dev	428K	293K
/etc	48M	23M
/home	114M	3.1M
/lib	64M	19M
/mnt	2.5K	3.0K
/opt	658M	350M
/proc	2.0K	16K
/root	2.2M	489K
/sbin	11M	6.9M
/tmp	8.4M	13M
/usr	2.5G	1.9G
/var	128M	23M
মোট	3.5G	2.3G

/usr — মূল '/' হায়েরার্কির মধ্যে দ্বিতীয় একটা হায়েরার্কি বা ক্রম। এই '/usr' ডিরেক্টরিতে থাকে গোটা সিস্টেমের মোট ফাইল এবং ডিরেক্টরির সমগ্রের সিংহভাগ। আমরা এখানে সুজে এবং স্ল্যাকওয়ার দুটো ডিস্ট্রোরই রুট ডিরেক্টরিতে থাকা ডিরেক্টরিগুলোর আলাদা আলাদা ভাবে মোট আয়তন আর গোটা সিস্টেমটার মোট আয়তন দিয়েছি, পাশাপাশি। সাধারণ একটা গড় ইনস্টলেশন কখনো কখনো এর থেকে সামান্য বেশি-কম হতে পারে। ডিভলপমেন্টের, মানে কম্পিউটার ভাষার আর প্রোগ্রামিং-এর যে প্যাকেজগুলো এই দুটো সিস্টেমেই ইনস্টল করা আছে, তার সবগুলো সব সিস্টেমে থাকেনা। আর পাবলিশিং সংক্রান্ত প্যাকেজও দু-চারটে বেশি থাকতে পারে। স্বাভাবিক একটা হোম পিসির একটা ইনস্টলেশনের সাইজের আন্দাজটাও এখান থেকে পেতে পারবেন। যে কমান্ড দিয়ে আমি এ তালিকাটা পেয়েছি সেটা হল '/' ডিরেক্টরিতে দাঁড়িয়ে '`du -chs *`', তারপর তাকে আমাদের চেনা কায়দায় রিডাইরেক্ট করে। ম্যানপেজ পড়ে দেখুন এই 'du' কমান্ডটা ভারি চমৎকার ভাবে ডিস্কভূমির ব্যবহারটা দেখায়। এর '-chs' অপশানগুলোর মানে নিজেই দেখে নিন। এছাড়া আর একটা কমান্ডও এই কাজে খুব ব্যবহার হয়, 'df'। তালিকাটায় 'M' মানে মেগাবাইট, 'K' মানে কিলোবাইট, 'G' মানে গিগাবাইট। সুজেতে দেখুন '/usr' জুড়ে আছে গোটা সিস্টেমের মোট ব্যবহৃত ডিস্কভূমির প্রায় একাত্তর শতাংশ আর স্ল্যাকওয়ারে প্রায় তিরাশি শতাংশ। এই '/usr' ডিরেক্টরির আলোচনায় আমরা পরে আসছি।

/var — ভ্যারিয়েবল বা পরিবর্তনশীল তথ্যদের আবাস। সিস্টেম লগিং সংক্রান্ত তথ্যের ফাইল, মেল আর প্রিন্টার স্পুলের ফাইল, এই সব থাকে '/var' ডিরেক্টরিতে। এই '/var' ডিরেক্টরি দুটো সিস্টেম কখনো শেয়ার করতে পারেনা, মানে দুজনেই আলাদা ভাবে একই পার্টিশনকে ব্যবহার করতে পারেনা। কারণ এখানে থাকা তথ্যের গোটাটাই একটা সিস্টেমের একান্ত নিজস্ব। এই ডিরেক্টরির তথ্যগুলো সিস্টেম ব্যবহারের সঙ্গে তাল মিলিয়ে সব সময়েই বদলাচ্ছে।

গ্নু-লিনাক্স ফাইলসিস্টেম স্ট্যান্ডার্ড অনুযায়ী উপরের ডিরেক্টরিগুলো একটা ইনস্টলেশনের রুট ডিরেক্টরিতে বা '/'-এ থাকতে হবে, বা, নিদেন, তাদের লিংক। কিন্তু এগুলো ছাড়াও আরো কিছু ডিরেক্টরি থাকতে পারে। কোনো কোনো ডিস্ট্রোতে এর বাইরেও ডিরেক্টরি থাকে বৈকি, যেমন সুজেতেই '/srv', যেটার মধ্যে সার্ভার সংক্রান্ত তথ্য রাখে। এর মানে, যতটুকুই হোক, গ্নু-লিনাক্স স্ট্যান্ডার্ড থেকে নড়ছে সুজে। উপরের এই অত্যাবশ্যকগুলোর বাইরে নিচের এই অতিরিক্ত ডিরেক্টরিগুলোও রুট ডিরেক্টরিতে থাকতে পারে। এফএসস্ট্যান্ড তার অনুমতি দেয়।

/initrd — '/initrd' ডিরেক্টরি গ্নু-লিনাক্স সিস্টেমকে দেয় বুটলোডার মারফত র‍্যামডিস্ক ব্যবহারের সুযোগ। র‍্যামডিস্ক (RAM Disk) মানে র‍্যান্ডম-অ্যাকসেস-মেমরির একটা বিশেষ এলাকা, র‍্যামডিস্ক বানানোর সফটওয়ার দিয়ে যাকে বানিয়ে নিতে হয়। এই এলাকাটা ঠিক একটা ছোট্ট সাইজের হার্ডডিস্ক পার্টিশনের মত ব্যবহার করে। হার্ডডিস্ককে নকল বা এমুলেট করে এই র‍্যামাঞ্চলটা। সুবিধেটা এই যে হার্ডডিস্ক পার্টিশন থেকে তথ্য পড়ার চেয়ে বহুগুণ বেশি গতিতে এইখান থেকে তথ্য পড়ে নিতে পারে সিপিইউ, দ্রুত তার কাজে ঢুকে যেতে পারে। আর যখনই মেশিন শাটডাউন করছি বা রিবুট করছি তখনই এই র‍্যামডিস্কটা মুছে যাচ্ছে। একটা সঠিক সাইজের অপারেটিং সিস্টেমকে, র‍্যাম-এলাকাটার মধ্যে যেটা ভরে যাবে, এইভাবে র‍্যামে রেখেই কাজ করে যেতে পারে কম্পিউটার। '/initrd' ডিরেক্টরিটা সুযোগ দেয় র‍্যামডিস্ককে রুট ফাইলসিস্টেম হিশেবে মাউন্ট করার, এরপর এখান থেকেই প্রয়োজনীয় প্রোগ্রাম চালানো যায়। তারপর, প্রাথমিক কাজ শেষ হলে, সঠিক হার্ডডিস্ক পার্টিশনকে রুট ডিরেক্টরি হিশেবে মাউন্ট করে নেওয়া যায়। এই ভাবে র‍্যামডিস্ক রাখার সুযোগ করে দিয়ে '/initrd' ডিরেক্টরি সিস্টেম বুটের গোটা কাজটাকে দুটো অংশে ভেঙে দেয়। প্রথম অংশটায় কারনেলে অনেক কম ড্রাইভার কম্পাইল করা থাকলেও অসুবিধে নেই, দ্বিতীয় স্তরে প্রয়োজনীয় মডিউলগুলো '/initrd' ডিরেক্টরি থেকে তুলে নেয় কারনেল। এই ডিরেক্টরির জটিলতায় আর গিয়ে কাজ নেই, আপনার ইচ্ছে থাকলে আপনি নিজেই শিখে নিতে পারবেন। '/initrd' বানানোর যে কমান্ড, 'mkinitrd', তার ম্যানপেজ পড়ুন। আমাদের এখানে তুলে দেওয়া তালিকাটা দেখুন, সুজে রুট ডিরেক্টরির মধ্যে কোনো '/initrd' ডিরেক্টরি নেই। এই '/initrd' ডিরেক্টরিটা ছাড়াই সুজে কিন্তু র‍্যামডিস্ক ব্যবহার করছে। দেখুন তো, পাঁচ নম্বর দিনে তুলে দেওয়া '/etc/lilo.conf' ফাইলটা থেকে তার কোনো হদিশ পান কিনা। এটা পাঠমালার টাস্ক, আমি এটার উত্তর জানাব না।

/home — বারবার ব্যবহার করতে করতে এই ডিরেক্টরিটা আমাদের চেনা হয়ে গেছে। সমস্ত ব্যবহারকারীর নিজের নিজের হোম ডিরেক্টরি বা নিজের ঘর মিলিয়ে তৈরি এই ব্যবহারকারীদের কোয়ার্টার। এই সার্বজনীন ঘরগেরস্থালির মধ্যে একজন বিশেষ ব্যবহারকারী বা ইউজারের হোম ডিরেক্টরির হল '/home/$USER' যার অন্য নাম '~/'। এই দুটোকে কি চিনতে পারলেন? প্রত্যেকটা ইউজারের নিজের ঘরে কী কী ফাইল থাকে তার সঙ্গে মোটামুটি একটা পরিচয় তো আমাদের আগেই হয়ে গেছে। কিছু ডিরেক্টরি, কিছু ফাইল, আর কিছু সেই সতত-অদৃশ্য ডটানন ফাইল এবং ডিরেক্টরি, যাদের মধ্যে রাখা আছে বিভিন্ন ধরনের কনফিগারেশন বা সিস্টেম সেটিং। মনে করতে পারছেন '/etc/skel' ডিরেক্টরির কথা যার স্কেলিটনের বা কঙ্কালের আদলে একজন নতুন ব্যবহারকারীর নতুন হোম ডিরেক্টরি তৈরি করে নেওয়া হয়? এই '/home' ডিরেক্টরিটা, ব্যবহারকারীদের কল্যাণে, বড্ড দ্রুত ভরে যেতে শুরু করে। অনেকসময় তাই এই ডিরেক্টরিটাকে আলাদা একটা পার্টিশনে দিয়ে দেওয়া ভালো, যাতে বেড়ে গিয়ে গিয়ে মূল রুট পার্টিশনের ফাইলগুলোরই দমবন্ধ করে না-দেয়। সেই সমস্যার কথা তো বলেছি আগেই।

/lost+found — দেখুন তো, এই মহাপ্রভুর নামটা আপনার চেনা চেনা লাগছে, কোথায় যেন শুনেছিলেন? ইএক্সটিটু এবং ইএক্সটিথ্রি ফাইলসিস্টেমের ক্র্যাশ-রিকভারি সংক্রান্ত আলোচনা মনে পড়ছে? আমি আগে এটার কথা জানতাম না। তখনো অব্দি আমি ম্যানড্রেক ব্যবহার করছিলাম, একদিন শখ হল, একটু রেডহ্যাট ইনস্টল করে দেখি। প্রথমত, ইনস্টল করতে করতে এবং বুট করতে করতে চোখে প্রায় জল এসে গেছে, তার আগে অব্দি ম্যানড্রেকে এক্সএফএসের ওই দুর্ধর্ষ গতিতে কাজ করে এসেছি, মনে হচ্ছে সবকিছু যেন স্লো-মোশনে ঘটছে, এরপর ইনস্টল হওয়া সিস্টেমে লগ-ইন করে '`ls`' মেরে দেখি, '/lost+found' জাতীয় মেলো-রোমান্টিক নাম, সুখেন দাস কোথায় লাগে? তখনো ফাইলসিস্টেম, তার মেটাডেটা, তার ক্র্যাশ-রিকভারির কৌশল এসব জানতাম না। আপনারা যা জানেন। ক্র্যাশ-রিকভারির সময় ফাইলসিস্টেম চেক করে পুনরাহরিত ঘেঁটে যাওয়া তথ্য রাখা থাকে এই ডিরেক্টরিতে, ইএক্সটিটু ফাইলসিস্টেমে, আগেই বলেছি। অচিরেই রাইজার ফাইলসিস্টেমে রেডহ্যাট ইনস্টল করে আমি মুক্তি পেয়েছিলাম এর থেকে। এর কিছুদিন বাদে রেডহ্যাটটাই উড়িয়ে দিয়েছিলাম। আপনি যদি স্লোমোশন ভালোবাসেন তাহলে ইএক্সটিতে রেডহ্যাট করুন, এবং '/lost+found'-র শোভা দেখুন বসে বসে। অনেকটা সময়ই তো আপনাকে বসে থাকতে হবে, প্রতিবার কমান্ড দিয়ে প্রম্পট ফেরত আসার জন্যে।

/proc — এই ডিরেক্টরির গোটা ফাইলসিস্টেমটাই অন্য প্রতিটি ডিরেক্টরির থেকে একদম আলাদা এই অর্থে যে এটা কোনো বাস্তব ফাইলব্যবস্থাই না, একটা বিমূর্ত এবং সাঙ্কেতিক ফাইলব্যবস্থা। সিস্টেমের প্রসেসগুলোর বিষয়ে তথ্য যোগানোর উদ্দেশ্যে বানানো একটা নকল ফাইলব্যবস্থা। আমরা, কী কী ফাইলব্যবস্থা আমাদের সিস্টেমে ব্যবহার হচ্ছে সেটা জানার জন্যে, '`cat /proc/filesystems`' কমান্ড ব্যবহার করেছিলাম, মনে আছে? কোনো বাস্তব ফাইল '/proc' ডিরেক্টরিতে থাকেনা। থাকে শুধু সিস্টেম কাজ করার সময়ে সিস্টেমের মেমরি, সিস্টেমে কী কী ডিভাইস মাউন্টেড আছে, সিস্টেমের হার্ডওয়ার কনফিগারেশন ইত্যাদি সিস্টেম-তথ্য। '/proc' ডিরেক্টরিতে গিয়ে, এই গোটা পাঠমালা জুড়ে বাস্তব ফাইল নিয়ে আপনি যা যা জেনেছেন সেটা দিয়ে কদাচ বোঝার চেষ্টা করবেন না। যেমন, আপনি জানেন, একটা ফাইলে টেক্সট থাকার সঙ্গে ফাইলের বাইটমাপের সম্পর্ক। এবার '`cat /proc/filesystems`' করে স্ক্রিনে চিহ্নের সংখ্যা গুনে নিয়ে আপনি যদি সেটা দিয়ে '/proc/filesystems' ফাইলটার সাইজের সঙ্গে মেলাতে যান, প্রচণ্ড চমকাবেন। '`ls -sh`' করে দেখুন গোটা '/proc' ডিরেক্টরির যাবতীয় ফাইলেরই সাইজ দেখবেন শূন্য, 'kcore', 'mtrr', আর 'self' বাদ দিয়ে। এবং মজার কথা কী বলুন তো, আপনার অত্যাবশ্যক পার্টিশনগুলো ছাড়া আর কিছু যখন মাউন্ট করা নেই, মানে আমাদের মাউন্টের ছবিতে স্টপককহীন ওই তিনটে চৌবাচ্চা বাদ দিয়ে আর কাউকেই যখন সিস্টেম 'তুমি যে আমার' করেনি, তখন আপনি '`du -chs /proc`' করে যে সাইজ পাবেন সেটা নাটকীয়ভাবে বদলে যাবে, যেই আপনি পার্টিশনগুলো মাউন্ট করে নেবেন। আমার মেশিনের সুজে সিস্টেমে যখন স্ল্যাকওয়ার, আর্কাইভ আর উইনডোজ পার্টিশনগুলো মাউন্ট করা নেই তখন '/proc' ডিরেক্টরির 'du' মানে ডিস্ক-ইউসেজ বা ডিস্ক-ব্যবহার দেখাচ্ছে চুরানববই কিলোবাইট। আর ওগুলো মাউন্ট করা মাত্রই ডিস্ক-ব্যবহার বেড়ে গিয়ে হচ্ছে

দুশোপঁচিশ মেগাবাইট। এবং তখনও যাবতীয় ফাইলের আকার শূন্য, একা 'kcore' ফাইলটাই প্রায় ওই গোটা সাইজটা নিয়ে নিচ্ছে। কিন্তু আর না, আমরা আমাদের পাঠমালার এক্তিয়ার ছাড়িয়ে বেরিয়ে যাচ্ছি।

/root — এটা হল রুট ইউজারের বা সিস্টেম অ্যাডমিনিসট্রেটরের নিজের হোম ডিরেক্টরি। '/' বা রুট ডিরেক্টরির মধ্যে এই রুট বা '/root' নামের ডিরেক্টরিতে গুলিয়ে ফেলবেন না। অনেক আগে ওই '/' ডিরেক্টরিটাই ছিল রুট ব্যবহারকারীর হোম-ডিরেক্টরি। সেখানেই থাকত তার নিজের ফাইলপত্তর। পরে, ঘরকন্নার কাজ একটু পরিপাটী রাখার উদ্দেশ্যে এই '/root' ডিরেক্টরিটা বানানো হয়। এই ডিরেক্টরিটা যেহেতু একজন ব্যবহারকারীর হোম ডিরেক্টরি, একে '/home' ডিরেক্টরির মধ্যেই আর একটা সাবডিরেক্টরি, '/home/root' নামেই তো রাখা যেত? সেটা করা হয়না, তার কারণ, আমাদের মত পিসি ব্যবহারকারীদের জন্যেই শুধু গ্নু-লিনাক্স তৈরি হয়নি। নেটাবদ্ধ বড় বড় সিস্টেমে অনেক সময় ভৌগোলিক দূরত্বে থাকা একটা পার্টিশনকেও মাউন্ট করা হয় '/home' ডিরেক্টরিতে। সেটাও তখন স্টপকক লাগানো চৌবাচ্চা হয়ে যায়। কিন্তু অন্য ব্যবহারকারীদের সাময়িকভাবে বাইরে রেখেও একটা সিস্টেম চালু হলেও, সুপারইউজারকে তো মূল সিস্টেমের অত্যাবশ্যকীয় অংশের মধ্যে থাকতেই হবে। তাই এই ব্যবস্থা। এই সবিশেষ সর্বশক্তিমান ব্যবহারকারীর একটা আলাদা সুপারভাইজর কোয়ার্টার একদম রুট পার্টিশনের মধ্যেই।

গ্নু-লিনাক্স ফাইলসিস্টেম হায়েরার্কির প্রাথমিক ধারণাটা হল আমাদের। রুট ডিরেক্টরি স্ল্যাশ বা '/'-এর মধ্যে কী কী ডিরেক্টরি থাকে সেটা আমরা জানলাম। এবার এই ডিরেক্টরিগুলোর চারটেকে নিয়ে আরো দু-চারটে কথা বলার আছে। '/bin', '/etc', '/sbin' এবং '/usr' — এই চারটে ডিরেক্টরি আলাদা করে আরো একটু বলে নিতে হবে। সেটা দিয়ে আমরা বরং সামনের দিনের আলোচনা শুরু করব। সেটাই আমাদের এই জিএলটি ইশকুল পাঠমালার শেষের আগের দিন। নয় নম্বর।

গ্নু-লিনাক্স ফাইলসিস্টেম বোঝা শেষ করে আমরা ডিরেক্টরি-কাঠামোর হায়েরার্কিতে পৌঁছেছি। কিন্তু এখনো খুব গুরুত্বপূর্ণ একটা জায়গায় একদমই হাত দেওয়া হয়নি। বারবার সেটার প্রসঙ্গ আসছে, এবং পিছলে পিছলে বেরিয়ে যেতে হচ্ছে। সেটা হল কাস্টমাইজেশন। নিজের প্রথা ইচ্ছা অভ্যেশ পছন্দ অনুযায়ী সিস্টেমকে নিজের মত করে বদলে নেওয়া। তার প্রাথমিক উপাদানগুলো অনেক জায়গাতেই গুঁজতে গুঁজতে এগোচ্ছি। শুধু আলাদা করে খেয়াল করাচ্ছি না। পুরোটাই দেখবেন মিলে যাবে, যখন সামনের নয় নম্বর দিনে আমরা এই কাজে হাত দিতে শুরু করব। সিস্টেম ভ্যারিয়েবল এবং কনফিগারেশন ফাইলদের বদলে। এই স্বাদ আপনি একটু পেয়েছেন '/etc/fstab' ফাইলের বেলায়। তবে সেটা কিছুই না। এই পাঠমালার দশ নম্বর দিনে ব্যাশের এবং ব্যাশস্ক্রিপ্টের আলোচনায় গিয়ে দেখবেন এই গোটাটাই মিশে যাচ্ছে একসঙ্গে।

glt-mad@ilug-cal.org