গ্নু-লিনাক্স ইশকুল

জিএলটি মধ্যমগ্রাম
glt-mad@ilug-cal.org

কম্পিউটার, তার হার্ডওয়ার সফটওয়ার, ইতিহাস, অপারেটিং সিস্টেম এবং অ্যাপ্লিকেশনের প্রাথমিক আলোচনা সেরে, অবশেষে আমরা গ্নু-লিনাক্সে পৌঁছচ্ছি এবার, যেখানে পৌঁছনোর কাজেই আমাদের এই জিএলটি ইশকুল আর তার ক্লাস। তবে ভাবার কোনো কারণই নেই যে এখান থেকে আমরা গ্নু-লিনাক্সাধিকারী হয়ে গেলাম। যেটা ম্যাক্সিমাম ঘটতে পারে এই ক্লাস থেকে তা হল, কীভাবে এগোবো তার একটা আন্দাজ — এগোতে হবে নিজে, এগোতে গিয়ে কেলো ঘটবে, সিস্টেম ঘেঁটে যাবে, ইনস্টলেশন নষ্ট হয়ে যাবে — যত প্রভূত পরিমাণে হবে এবং হতে থাকবে এইসব যাবতীয় কেলেঙ্কারি, ততই আসলে শেখাটা এগোবে। আর উইন্ডোজ তো হল জর্জ শার্দুলের মন্তব্যে হলিউড ছবির মত — চুইংগাম চিবোতে চিবোতে অ্যাভারেজ অ্যামেরিকানও যাতে তরতরিয়ে এগোতে পারে। অ্যাভারেজ অ্যামেরিকান বোধহয় পৃথিবীর সবচেয়ে ভয়ঙ্কর অ্যাভারেজ, মূর্খতা অ্যাফর্ড করার জন্যেও তো সম্পদশালী হতে হয়। সেই উইন্ডোজ থেকে যারা গ্নু-লিনাক্সে অভিবাসী হয়, তাদের আরো একটা সমস্যা, ডায়োরিয়ার রিফ্লেক্সকে আদ্যোপান্ত বদলে ফেলতে হয় কনস্টিপেশনের রিফ্লেক্সে। অবশ্য এর একটা চুইংগাম সমাধানও আছে — গুইতে কাজ করা, মাউসের পেট টেপার বাইরে আর প্রায় কোনো ক্রিয়ায় না-গিয়ে। কিন্তু এর মানে, আপনার নাগালের মধ্যে ছিল একটা দৈত্য — যেকোনো কাজ তাকে দিয়ে করাতে পারতেন, না-করে, শুধু তার শিঙে একটু পিঠ চুলকে নিলেন। মাইক্রোসফট আপনাকে এই সুযোগ দিতে পারেনা, ওটা তাদের ব্যবসা — কিন্তু গোটা গ্নু-লিনাক্স কমিউনিটিটা, চিন থেকে মেক্সিকো, আইসল্যান্ড থেকে অস্ট্রেলিয়া, আপনার জন্যে বসে আছে, তাদের করে-ফেলা কাজ এবং আরো কাজের সম্ভাবনা নিয়ে। আপনাকে নেমন্তন্ন করছে, বাইটের পর বাইটে — শেখো, প্রয়োগ করো, সব বই খাতা নোটবই তোমার সামনে খোলা। আপনি শিখতে যান, ঘেঁটে ফেলুন, যদি নেটে ঢোকার পয়সা থাকে তো সোজা লাগ-এর, মানে লিনাক্স-ইউজার-গ্রুপের সাইটে চলে যান, www.ilug-cal.org। আর যদি না-থাকে, ওয়েল অ্যান্ড গুড, জিএলটিতে আসুন, আমরা যেটুকু জানি তাই দিয়ে ঘাঁটা মেশিনকে সিধে করার চেষ্টা চালান, তাতেও না-হলে লাগ-এ, তারপরেও না-হলে নেটে। গোটা গ্নু-লিনাক্স গ্লোব আপনার জন্যে দিনে তেইশ ঘন্টা উনষাট মিনিট ষাট সেকেন্ড জেগে রয়েছে অতন্দ্র। তারপর, যখনই একটু শিখলেন, মনে রাখবেন, আপনার কমিউনিটিতে, ঠিক যেমন আপনি পেলেন, আপনারও দেওয়ার আছে। পশ্চিমের লিনাক্সীদের চেয়েও, বোধহয় আপনার দায়িত্ব একটু বেশি, আপনার চার পাশে এত এত বেশিসংখ্যক তৃতীয় বিশ্বের তৃতীয় ছানারা, যাদের কোনোকিছুরই অধিকার নেই, খাওয়ার অধিকার, পরার অধিকার, কম্পিউটার শেখার অধিকার। সেই গ্নু-লিনাক্স আজ এল, শেষ অব্দি, জিএলটি পাঠমালার এই সিরিজে। আজ প্রথমে বুটপ্রক্রিয়াটা আলতো করে একটু, তারপর লগ-ইন। কম্পিউটার বুট করার পর গ্নু-লিনাক্স সিস্টেমে কাজ শুরু করার আগে নিজের পরিচয় দিয়ে নিতে হয়, যার ভিত্তিতে সিস্টেম ঠিক করে এই ব্যবহারকারীর অধিকার কতদূর অব্দি। কোন ফাইল সে পড়তে লিখতে কাজে লাগাতে পারবে, কোন ডিরেক্টরিতে সে ঢুকতে পারে — ইত্যাদি। গুই মোডেও ঢোকা বা বেরোনো যায় একটা সিস্টেম থেকে। কিন্তু পিছনে কাজটা হয় কমান্ড দিয়েই — শুধু দেখানোর জন্যে থাকে ইঁদুরের কুটকুট, হাতির যেমন দুসেট দাঁত, দেখানোর আর খাওয়ার। তাই, কমান্ড প্রম্পটে লগ-ইন-এর ধারণাটাই আসল। তারপর যাব শেলে, ব্যাশ শেল কী করে সব প্রোগ্রাম আর কারনেলের মধ্যে মধ্যস্থতা করে।

।। দিন পাঁচ।।

১।। ইউনিক্স — মেশিননিরপেক্ষ অপারেটিং সিস্টেম

আর একবার একটু ফেরত যাওয়া যাক ইতিহাসে। সেই দানবিক মেশিনগুলোকে মনে করুন। হলঘরের বাড়ির স্টেডিয়ামের মত সাইজ। শুধু এই আখাম্বা সাইজটাই না, সবচেয়ে বড় সমস্যাটা এই যে, প্রতিটি মেশিনেই কাজ করছে আলাদা আলাদা সিস্টেম। প্রত্যেকটা মেশিনের নিজস্ব এবং স্বতন্ত্র। একটা মেশিনে চলা প্রোগ্রাম অন্য মেশিনে চলবে না। প্রতিটি প্রোগ্রামকে মেশিনমাফিক বদলে, কাস্টমাইজ করে নিতে হত অন্য মেশিনে চালাতে গেলে। প্রতিটি প্রোগ্রাম তথা সফটওয়ারের বেলাতেই এই একই ঝঞ্ঝাট। সাধারণ ইউজারের, সিস্টেম অ্যাডমিনিস্ট্রেটরের, প্রোগ্রামচির, সবার ঝামেলা। আর, একটা কম্পিউটার রাখা মানে তখন হাতিপোষা। কেনার খরচ, চালানোর খরচ, রক্ষণাবেক্ষণের খরচ, প্রতিবার প্রতিটি সফটওয়ারকে করে বদলে ফেলার খরচ, সাধারণ ইউজারকে ওই প্রোগ্রাম

শিখিয়ে নেওয়ার খরচ। এই ভাবে চলেছিল অনেকগুলো দিন। ১৯৬৯ নাগাদ, বেল ল্যাবরেটরির কেন থম্পসনদের টিম হাত দিল এই ঝামেলা মেটানোর কাজে। মূল ইসুটা খেয়াল করুন — কম্পেটিবিলিটি বা সাযুয্য। একটা মেশিনের জিনিষপত্তর যাতে মেশিন বদলানোর সময় ফেলে দিতে না-হয়। প্রতিটি মেশিনে প্রতিবার প্রতিটি সফটওয়ারকে কেচে গণ্ডূষ করার এই রেকারিং কেলো থেকে মুক্তি। যে কোনো মেশিনের জিনিষপত্তর যাতে অন্য মেশিনে চালানো যায়।

থম্পসনরা তাদের কাজটা শুরু করলেন একদম গোড়া মানে অপারেটিং সিস্টেমের স্তর থেকে। তারা একটা নতুন সিস্টেম বানালেন, যার নাম ইউনিক্স। আগের অপারেটিং সিস্টেম থেকে এটা তিনটে মূল জায়গায় আলাদা।

(১) চালানো এবং বোঝাটা বেশ সরল এবং সুদৃশ্য, সিম্পল এবং এলিগান্ট। সুদৃশ্য বলতে এখানে স্পষ্ট বুদ্ধিদীপ্ত, সহজে বোধযোগ্য, যৌক্তিক পদক্ষেপে পরপর বিন্যস্ত। এই অর্থে এলিগান্ট শব্দটা প্রায়ই ব্যবহার হয় কম্পিউটার শাস্ত্রে। প্রোগ্রামের বেলাতেও। সিম্পল এবং এলিগান্ট হওয়ার মূল সুবিধেটা এই যে, কিছুটা মনে রাখতে পারলে, তারপর ভেবে ভেবেই কাজ করা যায়, গোটা নিয়মকানুন মুখস্থ করতে হয়না। কাজের একটা এলাকায় যে ধরনের পদক্ষেপে যে ধরনের যুক্তিমালা ব্যবহার হল, সেটা অন্য জায়গাতেও হয়, তাই একবার বুঝে গেলেই সেটা নানা জায়গায় কাজে লাগানো যায়।

(২) আগেকার অ্যাসেম্বলি কোড থেকে সরে এসে সরাসরি সি নামের কম্পিউটার ভাষায় লেখা হল এই অপারেটিং সিস্টেম। এতে মূল মজাটা এই যে, আগের ওই অ্যাসেম্বলি কোড সাধারণভাবে অপাঠযোগ্য। জানিনা সত্যি কিনা, আমি এই গল্প রিচার্ড স্টলম্যান এবং অ্যালান কক্স দুজনের সম্পর্কেই শুনেছি, এরা নাকি অ্যাসেম্বলি কোড পড়ে প্রোগ্রাম বুঝতে পারে। হতেই পারে এটা একটা শাহরিক রূপকথা, আর্বান লেজেন্ড। যদি নাও হয়, তাহলেও সেটা কোনো কাজের কথা না। অসম্ভব পরিশ্রম এবং সময়ের অপব্যয়। একটা গোটা ফিল্মকে স্থির স্লাইডে দেখার মত, সেকেন্ডে চোদ্দটা করে ফ্রেমের স্লাইড ধরে ধরে চাইলে একটা ফিল্মকে তো দেখাই যায়। এতদিনকার চলে আসা সেই অ্যাসেম্বলি কোডের বদলে এবার প্রোগ্রামগুলোকে সি দিয়ে স্টেপ-বাই-স্টেপ লেখা পড়া এবং বোঝা যাবে। কোথাও কোনো গড়বড় হলে, সংশোধন করা যাবে। মানববোধ্য হয়ে উঠল গোটাটা।

(৩) বোধযোগ্য মানবিক হাইলেভেল কোডে লেখা মানেই আর একটা বড় নাটকীয় বদল কিন্তু ঘটে গেল। এবার থেকে ওই অপারেটিং সিস্টেম রিসাইকল করা যাবে, একটা মেশিন থেকে অন্য মেশিনে নিয়ে যাওয়া যাবে, সেই মেশিনে নতুন করে কম্পাইল করে নেওয়া যাবে অপারেটিং সিস্টেমের কোডটা। সেই কম্পাইল করতে গিয়ে যদি দেখা যায়, কোনো সমস্যা ঘটছে, প্রয়োজনমাফিক তাকে বদলানো যাবে, সহজেই। কারণ, সেটা তো সি নামক মানববোধ্য ভাষায় লেখা। একটা মেশিনের উপাদান বা ডিভাইস অন্য মেশিনের থেকে আলাদা। তার মানে আলাদা ডিভাইস ড্রাইভার, আলাদা আলাদা রকমের সফটওয়ার আদেশ। উপাদান মোতাবেক আলাদা আলাদা ড্রাইভার কোডের মধ্যে ঢুকিয়ে বা বাদ দিয়ে নতুন মেশিনে কম্পাইল করে নিলেই অপারেটিং সিস্টেমটা নতুন মেশিনে চালানো যাবে। কারণ সেটা সি-তে লেখা। মানে, এই প্রথম আমরা পেলাম পুনর্ব্যবহারযোগ্য অপারেটিং সিস্টেম। মেশিন থেকে মেশিনে যে কোনো প্রোগ্রামের সাযুয্য বা কম্পেটিবিলিটির ভিত্তিটা তৈরি হল। এই পুনর্ব্যবহারযোগ্য অপারেটিং সিস্টেম দিয়ে সাযুয্যের প্রথম শর্তটা পালিত হল। অপারেটিং সিস্টেমটা কম্পেটিবল হয়ে উঠল।

এখানে সবচেয়ে গুরুত্বপূর্ণ এই তিন নম্বর পয়েন্টটা, মানে, রিসাইক্লিং। এর আগে অব্দি বাজারে যত কম্পিউটার বিক্রি হত, তার প্রত্যেকটারই আলপিন-টু-এলিফ্যান্ট গোটাটাই একেবারে নিজের। এই কম্পিউটারগুলোর গোটা অপারেটিং সিস্টেমের গোটা কোডটাই তৈরি হত গোটাটাই একদম ওই মেশিনের জন্যে আলাদা করে বানিয়ে তোলা। একটা মেশিন বিক্রি করা মানেই তার গোটা অপারেটিং সিস্টেমটা লিখে নেওয়া, তার কাজের সঙ্গে মিলিয়ে, যে যে হার্ডওয়ার উপাদান ব্যবহার করা হবে মেশিনে তাদের ড্রাইভারগুলোকে মিলিয়ে, তাদের উপরে যে যে কাজ করা হবে সেই কাজের প্রয়োজনের সঙ্গে মিলিয়ে। সেই গোটাটাই বদলে গেল ইউনিক্সে এসে। ইউনিক্স অপারেটিং সিস্টেমে গোটাটাই হয়ে উঠল মেশিন-নিরপেক্ষ। মানে, যা সব মেশিনেই এক। শুধু একটা ছোট জায়গাকে বাদ দিয়ে। কোন কোন ভৌত উপাদান মেশিনটায় থাকছে, তার ভিত্তিতে কিছু আলাদা মডিউল বা খণ্ডাংশ গুঁজে দেওয়া হবে অপারেটিং সিস্টেমে বা কারনেলে। ধরুন যে মেশিনে হার্ডডিস্ক নেই, তার হার্ডডিস্ক ড্রাইভারের মডিউল লাগবে

না, আবার খুব আলাদা ধরনের কোনো মেমরি থাকলে, তার বিশেষ রকমের মডিউলটা দিয়ে দিতে হবে কারনেলে। দু-একটা কথা বলে নেওয়া যাক অপারেটিং সিস্টেম আর কারনেল নিয়ে।

এই পাঠমালা চলতে চলতেই গোটা বিষয়টা আমাদের কাছে অনেক স্পষ্ট হয়ে উঠবে। একটা বাস্তব কেজো জিনিষকে সংজ্ঞা দিয়ে বোঝা খুব কঠিন। তবু একটা সংজ্ঞা দেওয়ার চেষ্টা করা যাক। কারনেল কাকে বলে? কারনেল হল অপারেটিং সিস্টেমের হৃদপিণ্ড, যার উপর মেশিনের ভৌত যন্ত্রাংশগুলো সহ যাবতীয় উপাদান নির্ভর করে থাকে। এটা তো প্যালার সেই 'লাথি মানে জাড্যাপহ' — সেইরকম হল। ব্যাপারটা কী? কারনেল কী করে? কারনেল দেখভাল করে একদম নিম্নতম স্তরে ভৌত যন্ত্রাংশগুলোর একটার সঙ্গে অন্যটার সম্পর্ক, নিয়ন্ত্রণ করে মেশিনের রসদ-বন্টণ, যার মধ্যে পড়ে মেমরি তথা হার্ডডিস্ক সহ যাবতীয় স্মৃতিভূমি, এবং তথ্য ঢোকা-বেরোনো বা ইনপুট-আউটপুট। মানে, কোন ব্যবহারকারী ডিস্কভূমির কোন কোন অংশে নিজের পদচিহ্ন রাখতে পারবে, সেটাও শেষ অব্দি নিয়ন্ত্রণ করে কারনেল। কী ভাবে সেটা বুঝতে আমাদের আট নম্বর দিনের ফাইলসিস্টেম অব্দি সবুর করতে হবে। অর্থাৎ সিস্টেমের নিরাপত্তার দেখভালের দায়িত্বও এক ভাবে দেখলে কারনেলের হাতে চলে আসছে। আর মেমরি এবং ইনপুট-আউটপুটের নিয়ন্ত্রণ এবং বন্টন নিয়ে কিছু কথা আমরা ইতিমধ্যেই বলেছি, পরে আরো অনেকটা বলব। কিন্তু একটা অস্বস্তি লাগল না? এইগুলো তো অপারেটিং সিস্টেম বা ওএস-এর কাজ, বারবার বলেছি আগে। তাহলে? কারনেলই কি ওএস?

একটা সিস্টেম, যার ভিতরে কাজ করা যায়, অপারেট করা যায়, যে নিজে অপারেট করে আপনাকে অপারেট করতে দেয়, সেটাই ওএস। তাহলে, মনে মনে ভাবার চেষ্টা করুন। ওএস-এর মধ্যে কী কী থাকবে? হার্ডওয়ার বা ভৌত উপাদানগুলোর সঙ্গে কথা বলতে পারা এবং তাদের ভাগবাঁটোয়ারা করতে পারা। এটা প্রাথমিক। এবার এটার উপর দাঁড়িয়ে প্রয়োগ বা অ্যাপ্লিকেশনগুলো চালানোর একটা জমি বানিয়ে দেওয়া। এই জমি বানানোর জন্যে তার দরকার পড়ে সিস্টেমের কনফিগারেশন তথা কিছু প্রাথমিক প্রোগ্রাম চালানোর ব্যবস্থা করা — যে প্রক্রিয়া বা প্রসেসগুলো না চললে অন্য প্রোগ্রামগুলো চালাতে পারবেনা ইউজার। ইউজারের চালানো ইউজার অ্যাপ্লিকেশনগুলো চলার জন্যেই এই সিস্টেমের অভ্যন্তরে এই প্রক্রিয়াগুলো চলা দরকার।

ধরুন, খুব সহজ, লোকেট বা 'locate' বলে একটা ইউজার কমান্ড দিয়ে কোনো ফাইলকে খোঁজা যায়। কিন্তু এই খোঁজাটা সম্ভব হওয়ার জন্যে প্রথমে গোটা সিস্টেমের যাবতীয় ফাইলের তালিকার একটা ডেটাবেস বানিয়ে নেওয়ার এবং মাঝেমাঝেই সেই ডেটাবেস পুনর্নবীকরণের দরকার পড়ে, কারণ ফাইল তো নিয়তই বদলাচ্ছে সিস্টেমে, কাজ করা মানেই ফাইল বানানো এবং বদলানো। এর কমান্ডটা হল আপডেটডিবি বা 'updatedb'। এবং সিস্টেম যাতে নিজে নিজেই এই ফাইলতালিকা পুনর্নবীকরণের কাজটা সময় থেকে সময়ে করে যেতে পারে তার জন্যে এই কাজটাকে তুলে দেওয়া যেতে সময় মাফিক কাজ করার যখ বা ডিমনের হাতে। এরকম একটা ডিমন বা যখ হল অ্যাট বা 'at'। সত্যিই, অ্যাট সেভেন-থার্টি কিছু করে দিতে বলে রাখলে তাকে তার কখনো ভুল হবেনা, সাড়ে সাতটা বাজা মাত্রই সে তার কাজ শুরু করে দেবে। এবার, মজার কথা হল, এই লোকেট কমান্ডটা যে কোনো ইউজার চালাতে পারে। কিন্তু ওই আপডেটডিবি বা অ্যাট নিয়ে কিছু করার জো নেই তার, শুধু সিস্টেমপ্রভু মানে রুট সেটা পারবে, কারণ, ওইখানে কিছু করতে গেলে কারনেলের স্তরে নিয়ন্ত্রণ লাগবে। এই অ্যাট বা আপডেটডিবি হল সিস্টেমের আভ্যন্তরীন প্রক্রিয়াগুলোর উদাহরণ। নয় এবং দশ নম্বর দিনে গিয়ে আমরা এরকম আরো অনেক সিস্টেম প্রসেস বা সিস্টেম প্রোগ্রামের সঙ্গে পরিচিত হব। তার মানে দেখুন, অপারেটিং সিস্টেমের মধ্যেই অন্তত দুটো স্তরকে খুব সহজে চিনে নেওয়া যাচ্ছে। একটা স্তর ভৌত উপাদানের দুনিয়া। আর একটা স্তর সিস্টেম প্রোগ্রামের। এই প্রথম স্তরটা হল কারনেলের যা কখনো কখনো ঢুকে পড়ে দ্বিতীয়টার মধ্যেও। আর এদের এবং এদের সঙ্গে আরো অনেককিছু মিলিয়ে তৈরি হয় ওএস। যার ভিত্তির উপর দাঁড়িয়ে আমরা কম্পিউটারে কাজ করি।

একটু অতিসরলীকৃত একটা চেনা চেহারা দেওয়ার চেষ্টা করা যাক। ধরুন, আপনি একটা সেলফোন কানেকশন নিলেন, যা দিয়ে ইন্টারনেট সংযোগ করা যায়। আপনি চালাতে পারলেন না আপনার গ্নু-লিনাক্স সিস্টেমে। লাগে মেল করলেন। ধরুন, জানলেন, যে ওই ধরনের ভৌত উপাদান এমনিতে কারনেলে দেওয়া নেই, কিন্তু ওয়াপ বা ওয়ারলেস-অ্যাকসেস-প্রোটোকল দিয়ে ইন্টারনেট করার জন্যে প্রয়োজনীয় মডিউল করে কারনেলে কম্পাইল করে নেওয়া যায়। আপনার সিস্টেমে এই বাড়তি মডিউলটা দিয়ে আপনি কারনেল কম্পাইল করে নিলেন। কারনেলে এই

বদল মাত্র আপনার সিস্টেমের কনফিগারেশনে এবং সিস্টেম প্রোগ্রামে কিছু প্রয়োজনীয় বদল ঘটে গেল। মানে মোট ওএসটাই বদলে গেল। এবার তার উপর দাঁড়িয়ে আপনি কানেক্ট করলেন নেটে এবং ব্রাউজার চালিয়ে নেট চরাতে শুরু করলেন। সেগুলো ইউজার প্রোগ্রাম।

২।। ইউনিক্স থেকে গ্নু-লিনাক্স

আগেই বললাম, ইউনিক্স সবচেয়ে বড় যে জায়গাটা নিয়ে এল — সেটা হল একই অপারেটিং সিস্টেমকে বারবার ব্যবহার করতে পারা, রিসাইক্লিং। হার্ডওয়ারের উপর নির্ভর করে থাকছে অপারেটিং সিস্টেমের একটা ছোট্ট জায়গা — যার নাম কারনেল। সিস্টেম আলাদা মানে আলাদা কারনেল, কারণ, এই কারনেলটা হল অপারেটিং সিস্টেমের ভিত। কিন্তু তার উপর যে ইমারতটা দাঁড়িয়ে আছে তাকে কিন্তু হার্ডওয়ার মাথায় রাখতে হয়না, মানে শেল, কমান্ড আর অ্যাপ্লিকেশনকে। যন্ত্রপাতির ডিটেইলস মাথায় রাখে কারনেল। ইউনিক্স-এর ক্ষেত্রে এই ইমারতটা লেখা সি-তে — একটা হাইলেভেল মনুষ্যবোধযোগ্য ভাষায়। তাই সেই অংশটাকে এবার যে কোনো মেশিনে নিয়ে যাওয়া যায়, শুধু ওই মেশিনের নিজস্ব কারনেলের ভিতটুকু লাগে। অর্থাৎ, এই প্রথম, বেল ল্যাবরেটরির থম্পসন এবং অন্যান্যদের কাজের ভিত্তিতে আমরা একটা অপারেটিং সিস্টেম পেলাম যা মেশিন-নিরপেক্ষ।

স্বাভাবিকভাবেই, সঙ্গে সঙ্গে উৎসাহিত হয়ে উঠল সফটওয়ার কোম্পানিগুলো। অ্যাদ্দিন তাদের প্রত্যেক মেশিনের জন্য আলাদা আলাদা সেট বানাতে হত, এবার তারা অন্তত আগের দশগুণ করে বেচতে পারবে এক একটা সফটওয়ার। অনেক নতুন কিছু এবার ঘটতে শুরু করল। আলাদা আলাদা বিক্রেতার আলাদা আলাদা মেশিন একই নেটওয়ার্কে এখন পরস্পর প্রতিপ্রাপকতায় আসতে পারছে, এর তথ্য ও চটকে দিচ্ছে, এর ছাপার কাজ আকার পাচ্ছে ওর প্রিন্টারে, ইত্যাদি। আগে যা সম্ভব ছিলনা, কারণ, তাদের অপারেটিং সিস্টেম ছিল আদ্যোপান্ত আলাদা, আর একটা কম্পিউটার কী করে অন্য কম্পিউটারের সঙ্গে কথা বলবে সেই নিয়মকানুনগুলো, যার টেকনিকাল নাম প্রোটোকল, সেই প্রোটোকলটাও তো মোট সিস্টেমেরই একটা অংশ। একজন ব্যবহারকারী এখন আলাদা আলাদা মেশিনে বসে কাজ করছে একই উপায়ে, একই ভাবে, একই কমান্ড দিয়ে — বা কমান্ডের বদলে মেশিনের একই জায়গায় একই লাথি মেরে — ব্রুস উইলিসের আরমাগেডনে স্পেসশিপের কম্পিউটারটা মনে করুন। ইন্ডিপেন্ডেন্স ডে-তেও, যতদূর মনে পড়ছে, জেফ গোল্ডব্লামের ভাইরাস কাজ শুরু করেনি মেশিন কিল খাওয়ার আগে অব্দি। এই পৃথিবীতো ব্যবসার, এখানে বিপ্লবী হতে গেলেও এনআরআইদের স্পনসরশিপ লাগে, এমতাবস্থায় সফটওয়ারওয়ালাদের এতাবৎ সমর্থন, ইউনিভার্সিটিবাজরা তো আগে থেকেই ছিল, জিতা রহো ইউনিক্স, ভীষণ জ্যান্ত রকমে বাড়তে বদলাতে নতুন নতুন ছানা পয়দা করতে লাগল ইউনিক্স, পরবর্তী দুটো দশক জুড়ে। দিনকে দিন আরো আরো নতুন কাজ করা যাচ্ছিল, নতুন ধরনের যন্ত্র, নতুন ছাঁদের প্রোগ্রাম, তাই হার্ড আর সফটওয়ারওয়ালারাও তাদের সিস্টেমে ইউনিক্স লাগিয়েই চলছিল।

পরপর মূল ইউনিক্স, এবং তার পিছনে পিছনে এর মূল ব্র্যান্ড বা ডিস্ট্রোগুলো কী ভাবে এসেছে তার একটা ছোট তালিকা। এখানে আমরা গুরুত্বপূর্ণগুলোই দেখিয়েছি কেবল। প্রথম দুটো ইউনিক্সের নামের বানানের তফাতটা খেয়াল করুন। পরে আসছি এই কথায়।

ইউনিক্সের সময়ধারা	
১৯৬৯	**UNICS**
১৯৭১	**UNIX** Time-Sharing System
১৯৭৪	MERT, PWB/UNIX
১৯৭৬	UNSW
১৯৭৭	LSX, Mini Unix, PWB, RT, TS, USG, Wollogong
১৯৭৮	1BSD, CB Unix, Interactive IS
১৯৭৯	2BSD, BRL Unix, UCLA Secure Unix, UNIX 32V
১৯৮০	3BSD, 4BSD, UCLA Locus, Xenix OS
১৯৮১	QUNIX, UNIX System III
১৯৮২	HP-UX, PC/IX, SPIX, SunOS, Ultrix-11, UNIX System IV, Venix

১৯৮৩	Coherent, mt Xinu, Sinix, UNIX System V
১৯৮৪	Dynix, **Minix**, QNX, SCO Xenix, Ultrix 32M, Unicos, UNIX System V Release 2, Xinu
১৯৮৫	IBM IX/370, Interactive 386/ix, Mach, MIPS OS, UNIX System V/286
১৯৮৬	A/UX, AIX/RT, Chorus, GNU, Plan 9, UNIX System V Release 3, UNIX System V/386
১৯৮৭	CTIX, HPBSD, SCO Xenix System V/386
১৯৮৮	BSD Net/1, Chorus/MiX, IBM AOS, IRIX, more/BSD, NeXTSTEP, Ultrix, UNIX System V Release 4
১৯৮৯	Acorn RISC iX, AIX PS/2, AIX/6000, Atari Unix, BOS
১৯৯০	AIX, AIX/370, AMiX, OSF/1, Solaris 1
১৯৯১	AIX/ESA, ASV, BSD Net/2, BSD/386, **Linux**, RISC iX, Trusted Xenix, UNIX Interactive
১৯৯২	386 BSD, AOS Reno, BSD/OS, Solaris 2
১৯৯৩	Dynix/ptx, FreeBSD, HP-UX BLS, MVS/ESA OpenEdition, NetBSD, Unicox-max, UnixWare
১৯৯৪	4.4BSD Lite 1, ArchBSD, Lites, Open Desktop, SCO UNIX
১৯৯৫	4.4BSD Lite 2, AOS Lite, DEC OSF/1 ACP, Digital Unix, OpenBSD, OpenServer 5, Sinix ReliantUnix, Trusted IRIX/B
১৯৯৬	Mk Linux, OPENSTEP, OS/390 OpenEdition, QNX/Neutrino, Unicos/mk
১৯৯৭	OS/390 Unix, ReliantUnix, Rhapsody
১৯৯৮	Monterey, Trusted Solaris, UnixWare 7, xMach
১৯৯৯	Darwin, Mac OS X, Mac OS X Server, Tru64 Unix
২০০০	AIX 5L, Debian GNU/Hurd, HP-UX 11i, Minix-VMD
২০০১	GNU-Darwin, Open UNIX 8, QNX RTOS, Security-Enhanced Linux, z/OS Unix System Services
২০০২	MicroBSD, MirBSD, SCO UnixWare 7
২০০৩	DragonFly BSD, ekkoBSD

তালিকার তথ্যসূত্র http://www.levenez.com/unix/, গিয়ে দেখুন, ভারি চমৎকার ছবি দিয়ে বোঝানো আছে বেল ল্যাবরেটরির মূলটা থেকে শুরু করে গোটা ইতিহাসটা।

ইউনিক্স প্রথমদিকে মিলতে মৈনাক মেইনফ্রেমে, বা তার ছোটভাই মিনি মেশিনে। ইউনিক্স চাও? কোনো বিশ্ববিদ্যালয়ে যাও, নিদেন কোনো সরকারী দপ্তরে, বা কোনো সুবৃহৎ ব্যবসায়। কিন্তু আজ আমরা যে পিসি মেশিনগুলোকে চিনি, তাদের আদিজনিতৃ মাইক্রো মেশিনেরা কিন্তু ইতিমধ্যে গোকুলে বাড়ছিল। আশির দশক তার প্রাত্যহিক পেইন্টের অনিবার্যতায় মানে প্রৌঢ়তায় পৌঁছতে পৌঁছতেই মাইক্রোরা ল্যান্ড করতে শুরু করল — এবারে আর শুধু বাজারে ব্যবসায় বা বিশ্ববিদ্যালয়ে না, ডাইরেক্ট টু হোম, হোম কম্পিউটারের ধারণা রৈ রৈ করে বাড়ছে। আর, ততক্ষণে, ইউনিক্স-ও পিছিয়ে নেই, বাজারে এসে গেল ইউনিক্স-এর হোম ভার্শন। কিন্তু কেন থম্পসনদের উন্মুক্ত আদি লিপির, ওপন সোর্স কোডের, দিন আর নেই, হোম মেশিনের উপযুক্ত প্রতিটি ইউনিক্সই ততদিনে কিনতে হচ্ছে।

লিনাস টরভাল্ডস তখন হেলসিঙ্কি বিশ্ববিদ্যালয়ে কম্পিউটার শাস্ত্রের ছাত্র। টরভাল্ডস-এর মনে হল, দাতব্য অ্যাকাডেমিক ইউনিক্স-এর একটা হোম সংস্করণ ফ্রিতে দেওয়া গেলে ভারি ভাল হত। আগেই বলেছি, মডেল তো তার কাছে ছিলই, ট্যানেনবম-এর মিনিক্স, কোড লেখা শুরু হল। শুধু লেখা না, রীতিমত একটা গবেষণা, যাতে অন্য অনেককে তার নিজের পাশে দরকার পড়ল, একা একা যে সমস্যাগুলোর সমাধান করা যাচ্ছিল না। এটা সেইরকম একটা আবেদন, নেটে, সাহায্যের আশায় — আমরা কম্প-অস-মিনিক্স নিউজ গ্রুপে টরভাল্ডস-এর মূল আবেদনটাই তুলে দিলাম। এর মধ্যে যে পোজিক্স স্ট্যান্ডার্ড এসেছে তা নিয়ে আমরা চার নম্বর দিনে কথা বলেছি। টরভাল্ডসের চেষ্টা ছিল একটা সর্বসাধারণের জন্য সম্পূর্ণ ফ্রি অপারেটিং সিস্টেম দেওয়া, যা মূল ইউনিক্সের সঙ্গে মিলবে। এই মেলার জায়গাটা রাখতেই তার পোজিক্স-এর কথা বলা। লিনাসের এই কারনেলটা বানাতে গিয়ে যে প্রাথমিক সফটওয়ারগুলো প্রয়োজন পড়ল তা তদ্দিনে এসে গেছে, রিচার্ড স্টলম্যানের শুরু করা গ্নু তথা ফ্রি সফটওয়ার আন্দোলনের দৌলতে।

```
From: torvalds@claava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: Gcc-1.40 and a posix-question
Message-ID: 1991Jul3.100050.9886@claava.Helsinki.FI
Date: 3 Jul 91 10:00:50 GMT
Hello netlanders,
Due to a project I'm working on (in minix), I'm interested in the posix
standard definition. Could somebody please point me to a (preferably)
machine-readable format of the latest posix rules? Ftp-sites would be
nice.
```

নতুন নতুন যন্ত্রাংশ বা গ্যাজেট নিয়ে আমরা এখন প্লাগ-অ্যান্ড-প্লে-র কথা বলি, যাকে অনেকে প্লাগ-অ্যান্ড-প্রে-ও বলেন, মেশিনে লাগাও তারপর বিশ্বকর্মা হোক, কি অন্য কোনো ফেভারিট ঈশ্বর, তার কাছে প্রার্থনা শুরু করো, ঠাকুর গ্যাজেটটা যেন কাজ করে, আরো যদি প্রতিবেশটা হয় উইন্ডোজ, অন্তত একবার রিস্টার্ট তো বাধ্যতামূলক। কিন্তু প্লাগ-অ্যান্ড-প্রে তখনো না-থাকুক রোজ রোজ গজাচ্ছিল নতুন নতুন যন্ত্রাংশ, সে নতুন ধরনের নতুন স্পিডের হার্ডড্রাইভ হোক, কিম্বা নতুন ধরনের ভিডিওকার্ড বা মনিটর যাই হোক। তার জন্যে নিত্য নতুন আলাদা আলাদা রকমের ড্রাইভার তো দিতে হবে, নইলে অপারেটিং সিস্টেম ওগুলোকে কাজে লাগাবে কী করে? যেটা ঘটছিল তা হল নতুন একটা যন্ত্রাংশ বাজারে এলে কেউ না কেউ সেটা গ্নু-লিনাক্স সিস্টেমে ব্যবহার করার চেষ্টা করছে এবং সেখান থেকে তৈরি হচ্ছে একটা নতুন ড্রাইভার। তদ্দিনে 'লিনাক্স' নামটাও চালু হয়ে গেছে, যদিও 'কোড লিনাক্স' নামে ফিল্ম মোতাবেক লিনাস প্রস্তাব করেছিলেন 'ফ্রিক্স' — ফ্রিক লোকেরা যা ব্যবহার করছে, নার্ড বা ফ্রিক নামদুটো তদ্দিনে লিনাক্সোৎসাহীদের সঙ্গে জুড়ে গেছে, আমরা যেমন পাগল বা খ্যাপা বলি। তথাগত এই পাঠমালার প্রথম খসড়া পড়ার পরে খেয়াল করিয়ে দিয়েছিল, এর সঙ্গে ফ্রিডম-এর 'ফ্রি'-টারও একটা মাত্রা আছে। জনপ্রিয় নাম লিনাক্স-এর জায়গায় কেন আমরা গ্নু-লিনাক্স ডাকছি, সেটা আরো স্পষ্ট হবে পাঠমালাটা পরে।

তাই নতুন নতুন আসা হার্ডওয়ারের জন্যে লেখা নতুন নতুন ডিভাইস ড্রাইভার গ্নু-লিনাক্স ভাণ্ডারে বেড়েই চলছিল। এই কোডলিখিয়েরা তাদের নিজের নিজের পিসিতেই থামেনি, ব্রহ্মার এই আন্ডায় প্রায় যা যা হার্ডওয়ার পাওয়া সম্ভব বা অসম্ভব তাদের সবারই কোড প্রায় লিখে ফেলা হয়েছে। এই হবিবাজ 'নার্ড' বা 'ফ্রিক'-দের কল্যাণে লিনাক্সনীয় হার্ডওয়ারের সংখ্যা বেড়েই চলেছে। এই খ্যাপাদের দৌলতেই শুধু যেকোনো পিসি মেশিনে অনাবিল চালানো যায় লিনাক্স তাই নয়, প্রচুর অদ্ভুত বিচিত্র খ্যাপাটে যন্ত্রাংশও চমৎকার কাজ করে লিনাক্সে — অন্য সিস্টেমে যাদের ফেলে দিতে হত।

টরভাল্ডস-এর ওই আবেদনের ঠিক দুবছর পরে লিনাক্স-ব্যবহারকারীর সংখ্যা দাঁড়াল বারো হাজার। শুরু হয়েছিল সাইবারখ্যাপাদের নিয়ে — এখন সুস্থিত সঙ্গত সংসারী মাথায়-তেল-মেখে-চান-করা ব্যবহারকারীর সংখ্যা ক্রমে বাড়ছিল। গোটা সময় জুড়ে একবারো পোজিক্স স্ট্যান্ডার্ড-এর নিয়মকানুন ভাঙা হয়নি। ইউনিক্স-এর প্রতিটি কাজ বা ক্রিয়া বা ফাংশন একে একে যোগ করা হচ্ছিল গ্নু-লিনাক্সে। ক্রমে গ্নু-লিনাক্স একটা পূর্ণাঙ্গ ইউনিক্স প্রতিরূপ বা ক্লোন হয়ে দাঁড়াল। যা ওয়ার্কস্টেশন থেকে শুরু করে উচ্চশক্তির সার্ভার অব্দি সবকিছুতে ব্যবহার করা যায়। আজ হার্ড এবং সফটওয়ার বাজারে প্রতিটি গুরুত্বপূর্ণ কোম্পানির একটা করে পুরোদস্তুর গ্নু-লিনাক্স টিম আছে — যারা নতুন নতুন কোড নির্মাণ করেই চলেছে।

ডেস্কটপ ছোট মেশিনের জগতেও লিনাক্স আজ পূর্ণবিকশিত। প্রথমদিকের লিনাক্স কোডলিখিয়েদের দৌলতে সবচেয়ে জরুরি ছিল নেটওয়ার্কিং ইত্যাদি কাজ। এখন অফিস স্যুট বলতে আমরা যা বুঝি সেখানেও লিনাক্স দিয়ে কাজ করা যায় অন্য যে কোনো অফিস স্যুটের সঙ্গে সমান রকমে। বহু ক্ষেত্রেই আরো অনেক ভালো করে। আমাদের লিনাক্সীদের এটা ভালো লাগুক বা না-লাগুক এই ডেস্কটপ জগতে মাইক্রোসফট এখনো সবচেয়ে বড় শক্তি, তার সঙ্গে প্রতিযোগিতায় একের পর এক নতুন নতুন ভালো ভালো সফটওয়ার গ্নু-লিনাক্সে রোজই যোগ হচ্ছে। ওয়ার্কস্টেশনে কাজের উপযোগী করে। একটা সহজ গুই-নির্ভর ইন্টারফেস, উইন্ডোজ যেমন, এবং ওয়ার্ড একসেল পাওয়ারপয়েন্ট ইত্যাদি মাইক্রোসফটের অফিস প্রয়োগের গ্নু-লিনাক্স বিকল্প।

সার্ভারের জগতে গ্নু-লিনাক্স সবচেয়ে দৃঢ় এবং জোরালো সিস্টেমদের একটা। যেখানে ডেটাবেস এবং বাণিজ্যিক ব্যবহারের প্রতিটির জন্য নির্দিষ্ট গ্নু-লিনাক্স সমাধান আছে। ইন্টারনেটে প্রতিমুহূর্তে আমরা যার রেফারেন্স পাই সেই আমাজন বুক শপ চলে গ্নু-লিনাক্সে। আমেরিকার পোস্ট অফিস ব্যবস্থা, জার্মান সৈন্যবাহিনী চলে গ্নু-লিনাক্সে। কলকাতার সিইএসসির কাজের গ্নু-লিনাক্স ব্যবস্থায় কাজ করেছে এমন ছেলে আছে আমাদের কলকাতা লাগে। ইন্টারনেটে সার্ভিস প্রোভাইডাররা রোজ আরো আরো বেশি করে বেছে নিচ্ছে গ্নু-লিনাক্সকে। গ্নু-লিনাক্সে নেট-এর কাজ করাও সবচেয়ে নির্ঝঞ্ঝাট। টাইটানিক সিনেমা বানানোর গোটা সিস্টেমটা চালানো হয়েছিল লিনাক্সে।

অন্যদিকে স্মার্টকার্ড, মোবাইল ফোন, পিডিএ (PDA — **P**ersonal-**D**igital-**A**ssistant — বলতে এক কথায় বোঝায় হালকা হাতের তালুতে ধরে যাওয়ার মত পামটপ, নির্দিষ্ট কিছু কাজ করে, সময়তারিখ রাখা, নোট নেওয়া, ছোট ডেটাবেস, ক্যালকুলেটর ইত্যাদি, খুব দামি গুলোতে মাল্টিমিডিয়া অব্দি হয়) বা অর্গানাইজার ইত্যাদি এমবেডেড সিস্টেমে, মানে যেখানে অপারেটিং সিস্টেমটা ইতিমধ্যেই সদাসর্বদাই, অলওয়েজ অলরেডি, হার্ডওয়ারের মধ্যে রয়েছে — এই জগতটায় লিনাক্স ক্রমে প্রায় সর্বব্যাপী হয়ে উঠছে। কী কী ধরনের হার্ডওয়ারে একটা অপারেটিং সিস্টেম চালানো যাবে সেই নিরিখে গ্নু-লিনাক্স-এর পাশে আর কোনো অপারেটিং সিস্টেমই আসতে পারেনা। এর ইতিহাসটা তো আমরা আগেই বলেছি। অর্থাৎ, গ্নু-লিনাক্স এখন একটা পূর্ণাঙ্গ ব্যবস্থা — আমাদের সবার প্রয়োজনই যা থেকে মিটছে। প্রতিটা নতুন দিন নতুন প্রয়োজন গজিয়ে উঠবে, জ্যান্ত লিনাক্সীদের নিয়ে জ্যান্ত গ্নু-লিনাক্সও বদলাতে থাকবে।

৩।। ইউনিক্স থেকে গ্নু-লিনাক্স — সালগত ক্রমপঞ্জী

এই আলোচনাটা আমরা করব একদম একটা ইতিহাস বইয়ের শেষে দেওয়া সালপঞ্জীর মত করে, সাল ধরে ধরে। তার মধ্যে দু-এক জায়গায় দু-একটা কথা বাড়তি বলে নেব ধরিয়ে দেওয়ার জন্যে। এই পঞ্জীটা বানাতে গিয়ে আমি মূলত চারটে ওয়েবসাইট থেকে সাহায্য পেয়েছি। এক, বাংলাদেশ লাগ-এর (http://bdlug.hypermart.net/) জন্যে রাজিব হাসানের লেখা 'হিস্ট্রি অফ লিনাক্স'। দুই, ইউনিক্স ইতিহাসের সাইট http://www.levenez.com/unix/। তিন, লিনাক্সের ইতিহাস নিয়ে পিটার সেলাসের বক্তৃতা http://technetcast.ddj.com/ থেকে পাওয়া এক ঘন্টার এমপিথ্রি ফাইল — বক্তৃতাটা ভারি মজার, জিএলটির রিসোর্স সিডিতে দেওয়া আছে, সতেরো মেগাবাইট সাইজ। চার, বেল ল্যাবরেটরির নিজের ইউনিক্স ইতিহাসের সাইট, http://www.bell-labs.com/history/unix/। পঞ্জীটা শুরু হচ্ছে ইউনিক্সেরও জন্মের আগে, ১৯৫৭ থেকে।

১৯৫৭।। বেল ল্যাবরেটরিতে নিজেদের গবেষণার কাজে তখন ব্যাচ সিস্টেম ব্যবহৃত হচ্ছিল, চার নম্বর দিনের জেনারেশনের ইতিহাসের সঙ্গে মিলিয়ে নিন। এই ব্যাচ সিস্টেমে কাজের প্রক্রিয়াটা কখনোই খুব আরামপ্রদ হচ্ছিল না। বহু কিছু কাজ নিজেদের হাতে করে, প্রত্যেকবার, করে দিতে হচ্ছিল। তৈরি হল বেসিস (BESYS) বলে একটা অপারেটিং সিস্টেম। এগুলো সবই ওএস-এর প্রাগৈতিহাসিক সংস্করণ। এখনকার ওএস-দের সঙ্গে একে মেলানো অনেকটা ছিপকলির সঙ্গে ডাইনোসর মেলানোর মত। কিছুটা মিল বলা যায় চার নম্বর দিনে আমাদের দেখানো ফোর্ট্রান মনিটরিং সিস্টেমের সঙ্গে।

১৯৬৫।। বেল ল্যাবরেটরি তখন ক্রমে তৃতীয় জেনারেশন কম্পিউটারে চলে যাচ্ছে, বড় একটা ওএস তাদের আশু প্রয়োজন। জেনেরাল ইলেকট্রিক এবং ম্যাসাচুসেটস ইনস্টিটিউট অফ টেকনোলজির সঙ্গে মিলে একটা যৌথ প্রকল্পে হাত দিল বেল ল্যাবরেটরি। আমরা পেলাম সেই সকল টার্মিনালের এক কেন্দ্র, গোটা শহরের এক সিস্টেম — সেই সর্বশক্তিমান সর্বত্রগামী মাল্টিক্স (MULTICS)।

১৯৬৯।। উনসত্তরে ঘটল চার চারটে ঘটনা, সেই মুহূর্তে তাদের একটার সঙ্গে আর একটাকে মেলানো নোস্ত্রদার বাবার দ্বারাও সম্ভব ছিলনা। কিন্তু পরে, ইতিহাসের পাতায় তারা মিলে গেল একই ঘটমানতায়। পিটার সেলাস-এর মত করে বলতে গেলে, যা শুধু কম্পিউটার না, পৃথিবীর বিজ্ঞান ও প্রকৌশলের মানচিত্রটাকেই বদলে দিল। (১) মানুষ চাঁদে পা দিল। (২) অগাস্ট সেপ্টেম্বরে কেন থম্পসন আর ডেনিস রিচি তৈরি করলেন ইউনিক্স-এর প্রথম কাঠামো। তার প্রথম এবং আশু কারণ, এটিঅ্যান্ডটি (AT&T) মাল্টিক্স ব্যবহার বন্ধ করে দিল, এবং থম্পসন আর রিচির দরকার পড়ল একটা নতুন অপারেটিং সিস্টেমের দরকার পড়ল,

যে সিস্টেমে তারা তাদের প্রিয় গেমস, ‘স্পেস ট্রাভেল’ খেলে উঠতে পারবে। এই খেলাটা তারা খেলত, ডিইসি বা ডেক-এর পিডিপি ৭ মেশিনে। পিডিপি সিরিজের কথা মনে পড়ছে? (৩) চৌঠা ডিসেম্বর একই সাথে বসল চারটে ইম্প (IMP — **I**nterface-**M**essage-**P**rocessors) বা আদিমতম প্রোটোটাইপ রুটার, অর্থাৎ এককথায় ইন্টারনেটের অঙ্কুরোদগম। এই চারটে জায়গা হল, এলএ, সান্টা-বারবারা, মেনলো পার্ক এবং উটা বিশ্ববিদ্যালয়। (৪) ২৮শে ডিসেম্বর টরভাল্ডস পরিবারে জন্মালো নতুন বাচ্চা, নাম রাখা হল লিনাস।

১৯৭১।। ইউনিক্স প্রথম এডিশন প্রকাশিত হল। এই প্রথম তার বানান হল UNIX। খোজা করা মাল্টিক্স — এই প্যারডি থেকে একটা পূর্ণবিকশিত ব্র্যান্ডনেম তৈরি হল, ওএস-এর। এই ইউনিক্সের ম্যানুয়ালের ভূমিকায় কেন থম্পসন আর ডেনিস রিচি লিখলেন একটা তাৎপর্যপূর্ণ বাক্যাংশ, ‘ … there are now only ten installations of UNIX os, but more to come …’, ‘...এই মুহূর্তে ইউনিক্স অপারেটিং সিস্টেম ব্যবহার হচ্ছে মাত্র দশ জায়গায়, কিন্তু আরো আসছে ...’। এই ‘আরো আসছে’ মানে ঠিক কতটা আসছে তা যদিও, বোধহয়, ওরা নিজেরাও আন্দাজ করতে পারেননি। এই ইউনিক্সে ছিল মোট ষাটটা কমান্ড। এর পর, আপনারা যখন এই কমান্ডগুলোর আজকের চেহারার সঙ্গে পরিচিত হতে থাকবেন, ছয় সাত আট নম্বর দিনে, এই পুরোনো চেহারাগুলোর সঙ্গে মিলিয়ে যাতে আমোদ পেতে পারেন, তাই কয়েকটা কমান্ডের উদাহরণ দিলাম এখানে। যেমন, ‘`b`’, মানে বি (B) ভাষায় লেখা প্রোগ্রাম কম্পাইল করো। বি ভাষার কথা তো আমরা উল্লেখ করেছি আগেই, সি নামের কম্পিউটার ভাষার পূর্বসূরী। সি (C) ভাষায় লেখা প্রোগ্রাম কম্পাইল করার জন্যে জিসিসি-কে দেওয়া কমান্ডটা মনে করতে পারছেন? আর একটা কমান্ড ছিল ‘`boot`’, মানে মেশিন রিবুট করো, অফ করে অন করো। বা কনক্যাটেনেট করার, ফাইল স্ক্রিনে দেখানোর এবং পরপর জুড়ে দেওয়ার কমান্ড ‘`cat`’, বা, যে ডাইরেক্টরিতে কাজ করছি সেটা বদলানোর কমান্ড ‘`chdir`’। বা ‘`cp`’ — ফাইল কপি করো, ‘`chmod`’ — ফাইলের ব্যবহারবিধি বদলাও, ‘`chown`’ — ফাইলের মালিকানা বদলাও, ‘`mv`’ — ফাইলের জায়গা বদলাও বা নাম বদলাও, ‘`wc`’ — ওয়ার্ড-কাউন্ট বা শব্দসংখ্যা গোনো, বা ‘`who`’ — কে বা কে-কে এই মুহূর্তে সিস্টেম ব্যবহার করছে সেটা দেখাও। ইত্যাদি। এই তালিকাটা লিখতে গিয়ে আমার নিজের যে কী বিচিত্র লাগছে, এর একদম প্রথমটা বাদ দিয়ে অন্য সবকটাই রোজ ব্যবহার করি, একদম রোজ, অথচ এগুলো আজ থেকে ঠিক পঁয়ত্রিশ বছর আগে থম্পসন আর রিচির বানানো। একাত্তরের এই সিস্টেমে পরবর্তীকালের অন্য ইউনিক্সগুলোর, যার মধ্যে ওপন-সোর্স ইউনিক্স বা গ্নু-লিনাক্সও একটা, সবচেয়ে গুরুত্বপূর্ণ জায়গার একটা একেবারেই নেই — তা হল পাইপ, যার বিশদ আলোচনা আসবে দশ নম্বর দিনে, আমাদের পাঠমালার এগারো নম্বর অংশে।

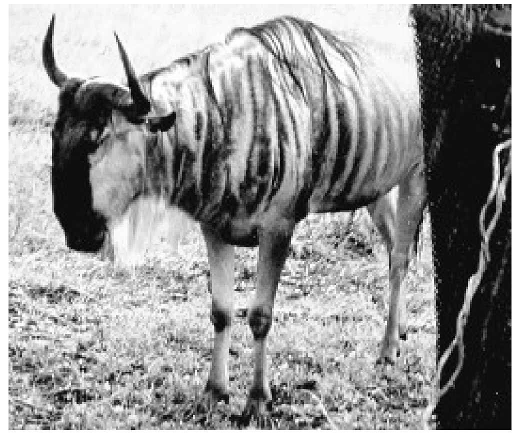
১৯৭২।। ইউনিক্স-এর দ্বিতীয় ভার্শন বেরোলো। আর সবচেয়ে বড় যেটা, ডেনিস রিচি তার বি-গত বিরক্তি থেকে সি-তে পৌঁছলেন। বি ভাষায় কাজের অসুবিধেগুলো একটা নতুন কেজো এবং ভালো কম্পিউটার ভাষার আবির্ভাবকে জরুরি করে তুলেছিল অনেকের কাছেই।

১৯৭৪।। বিল জয় এসে ইউনিক্স-এর কাজে যোগ দিলেন, এড (ed) বলে পুরোনো এডিটর-টা ব্যবহার করতে করতে তিতিবিরক্ত জয় বানালেন এক্স (ex) বলে একটা এডিটর। অপারেটিং সিস্টেম বা প্রোগ্রামিং-এর কাজে কমান্ড এডিটর কতটা গুরুত্বপূর্ণ সেটা তো এক নম্বর দিনেই আমরা আলোচনা করেছি। এবং ছবি নিয়ে খেলা করা বা গুই মোড আসতে তখনো বহু বহু দেরি। তখন এডিটরের ভূমিকা ছিল আরো অনেক বড়। এই এড কিন্তু খুবই শক্তিশালী একটা প্রোগ্রাম। এবং পরবর্তী প্রচুর এডিটরই পিছনে এডকে ব্যবহার করে। এডের একটা ভারি চমৎকার টিউটোরিয়াল আছে কারনিঘ্যান পাইকের বইয়ের অ্যাপেনডিক্স-এ। এই এড-স্রষ্টা বিল জয়ই পরে বানান ভিআই (vi) বলে একটা এডিটর যা গোটা ইউনিক্স জগতের কমান্ড এডিটিং-এ প্রায় একচ্ছত্র ছিল, রিচার্ড স্টলম্যানের ইম্যাক্স আসার আগে অব্দি।

১৯৭৫।। ইউনিক্স ভার্শন ছয় বেরোলো। এবং আরএমএস বা রিচার্ড স্টলম্যান লিখলেন তার ইম্যাক্স-এর প্রথম ভার্শন। রিচার্ড স্টলম্যান ছিলেন এমআইটির বিখ্যাত আর্টিফিশিয়াল ইন্টেলিজেন্স ল্যাবরেটরির সাইবারবিজ্ঞানী। ওই আর্টিফিশিয়াল ইন্টেলিজেন্স ল্যাবের অন্য প্রায় প্রত্যেকেই যখন চড়া দামে বিক্রি হয়ে

যাচ্ছে বড় বড় বাণিজ্যিক সংস্থায়, ঠিক তখনই স্টলম্যান শুরু করেন তার ফ্রি সফটওয়ার নিয়ে লড়াই। সময়টা খেয়াল করুন, পঁচাত্তর সাল। ইন্টারনেট তখনো আসেনি। ইন্টারনেটের পূর্বপুরুষ ছিল আর্পানেট, সেই আর্পানেটে সাইটের সংখ্যা তখন একশোরও নিচে, সাইট মানে আজকে আমরা যা বুঝি তার থেকে বহু যোজন দূরে তারা, ইন্টারনেট ঘটে উঠতে তখনো অনেক দেরি, যে ইন্টারনেটকে বাদ দিয়ে, আরো ষোল বছর পরে গ্নু-লিনাক্স ব্যাপারটা ঘটেই উঠতে পারত না। একটু বাদেই আমরা আসছি সেই প্রসঙ্গে। এই আর্পানেট (ARPA — **A**dvanced-**R**esearch-**P**rojects-**A**gency-NET) শুরু হয়েছিল উনিশশো ষাট-এর দশকে আমেরিকার ডিপার্টমেন্ট অফ ডিফেন্সের অ্যাডভান্সড রিসার্চ প্রোজেক্টের কাজে, যাতে বিভিন্ন বিশ্ববিদ্যালয় এবং গবেষণা সংস্থাগুলোর মধ্যে সরাসরি তথ্য বা ইনফরমেশন যাতায়াত ঘটতে পারে, পরে সত্তরের দশকে এর নাম হয় ডার্পা। এই আর্পা বা ডার্পা প্রাথমিকভাবে ব্যবহার হত মিলিটারির কাজে, যার থেকে পরে ইন্টারনেট গজিয়েছে। এই বিবর্তনটার একটা বেশ ইন্টারেস্টিং ইতিহাস আছে। নেটওয়ার্কিং বা বিশেষ করে টিসিপি/আইপি প্রোটোকলের ইতিহাসের এই অংশটা। কিন্তু সেটা আমাদের এই পাঠমালার এক্তিয়ারের বাইরে।

১৯৭৬।। ছিয়াত্তর সালে বলার মত, বা, আসলে না-বলার মত, ঘটনা একটাই। বিল গেটস-এর একটা চিঠি। 'ওপেন লেটার টু দি হবিস্টস'। যাকে সেলাস তার ওই ঘন্টাভর বক্তৃতায় ডেকেছেন 'দি ইনসিপিয়েন্ট ফুটপ্রিন্ট অফ দি র‍্যাপ্টর অফ রেডমন্ড বলে'। রেডমন্ড হল গেটস-এর ঠিকানা, আর র‍্যাপ্টরের মূল অর্থ শিকারী মাংসভুক পাখি, কিন্তু জুরাসিক পার্কে এতবার র‍্যাপ্টর ব্যবহার হয়েছে ভেলোসির‍্যাপ্টর অর্থে, সেই ঘিনঘিনে ছোট সাইজের ডাইনোসরগুলোর নাম হিশেবে, ওই উদ্বৃত্ত অর্থটা কিছুতেই এড়ানো যায়না এই দু-হাজার তিনের পৃথিবীতে। এই চিঠিতে বিল গেটস হবিস্টদের বা হবি-বাজদের উদ্দেশ্য করে লিখেছিলেন, '...most of you steal your software...' — তোমরা বেশিরভাগই সফটওয়ার চুরি করো। এখানে 'হবি' শব্দটার এই ব্যবহারটা বেজায় আমেরিকান, আমাদের অনেকটাই অপরিচিত। এটা বলতে বোঝায় ওই কম্পিউটার হুইজ কিডদের। বিল গেটসের চিঠির প্রায় পনেরো বছর পরে, ১৯৯১-এ, কম্প-অস-মিনিক্স মানে মিনিক্সের নিউজগ্রুপে লিনাক্স যখন প্রথম তার বানানো কারনেলটার খবর দেয়, একুশ বছরের সেই সদ্যযুবক, তাতে লেখে, I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones, আমি একটা ফ্রি অপারেটিং সিস্টেম বানাচ্ছি। নিছক একটা হবি, শখ — গ্নু, মানে রিচার্ড স্টলম্যানের গ্নু-র মত বড় বা পেশাদার কিছু নয়। এই চিঠির 'হবি' শব্দটার সঙ্গে মেলান গেটস-এর চিঠিটাকে। এই 'হবিস্ট' শব্দটার ধনাত্মক উদ্বৃত্ত অর্থটা পরে গোটাটাই স্থানান্তরিত হয়ে গেছিল 'হ্যাকার' (hacker) শব্দটায়। এখানে একটা কথা খেয়াল করিয়ে দেওয়া ভালো, 'হ্যাকার' বা 'হ্যাক করা' শব্দবন্ধ, মূলত অর্ধসাক্ষর সাংবাদিকদের কল্যাণে, দাঁড়িয়ে গেছে একটা খারাপ বিশেষণ — যারা অন্যের মেশিনে অন্যের নিজস্ব জগতে ঢুকে পড়ে এবং নিজের লাভের বা মজার জন্যে নানা বেআইনি কাজ করে। কিন্তু, কম্পিউটার লেখালেখির জগতে পড়ার সময় খেয়াল রাখবেন, 'হ্যাকার' শব্দটা রীতিমত সম্মানিত শব্দ, যারা কম্পিউটারের কার্যধারা একদম ভিতর থেকে বোঝে এবং বদলাতে পারে, মানে সিস্টেম বিষয়ে যথেষ্ট পারদর্শী প্রোগ্রামার। ওই খারাপ অর্থটার জন্যে সঠিক শব্দটা হল 'ক্র্যাকার' (cracker)। 'হ্যাকার' শব্দটার অর্থ এই একশো আশি ডিগ্রি ঘুরে যাওয়ার ভিতরে একটা গোপন ক্রিয়া আছে। আজকের পৃথিবীর সবচেয়ে সমর্থ এবং কুলীন ব্যবসা কম্পিউটার, আর সব কিছুকে যা শাসন করে। এই সাইবার শিল্পের পুঁজির আজ অব্দি সবচেয়ে বড় আতঙ্কটা তৈরি করেছিল হ্যাকাররা, এদের সাজানো বাগানে একপাল গ্নু ছেড়ে দিয়ে। ফ্রি লিবর ওপন — যে চেহারাতেই আসুক, সাইবার-বিজ্ঞানে উণ্মুক্ততা এনেছে এই হ্যাকাররাই। হ্যাকারদের প্রতি পুঁজির অবশ্যম্ভাবী আতঙ্ক এবং বিতৃষ্ণাটা এবার সংবাদমাধ্যম মারফত ছড়িয়ে গেছিল জনমানসে। ঠিক যে ভাবে শাসকের চিন্তাকে বারবার শাসিতের চিন্তা বানানো হয়েছে ইতিহাস জুড়ে।



গ্নু — ফ্রি সফটওয়ার ফাউন্ডেশনের ছবিতে এবং জঙ্গলে

১৯৭৭।। সাতাত্তরে এল বিএসডি প্রথম ভার্শন। এর সঙ্গে ছিল ওই এক্স এডিটর এবং পাস্কাল বলে একটা কম্পিউটার ভাষার বার্কলে সংস্করণ। এর জন্যে মোট দাম ধার্য ছিল কুড়ি ডলার। এই কুড়ি সংখ্যাটাকে মনে রাখুন, এখুনি আমাদের তুলনা করতে কাজে লাগবে। আর দুতিন বছরের মধ্যেই, শুধু যে এই দামটাই বীভৎস ভাবে বদলে যাবে তা নয়, এর সঙ্গে চালু হবে লাইসেন্সিং নিয়ে উন্মত্ত কড়াকড়ি। এই সময় অব্দি আবহাওয়াটা অনেকটাই বিদ্যাচর্চার, দাম যা নেওয়া হচ্ছে সেটাও ন্যূনতম যতটুকু প্রয়োজনীয়। এর পরেই আবহাওয়াটা চলে যাবে একচেটিয়া পুঁজির মুনাফাবাজিতে। আর সফটওয়ারের উপরেও আসবে বাধানিষেধ। যাতে একটা যুগের প্রোগ্রামারদের কাজের উত্তরাধিকার পরের যুগে আর পৌঁছবে না, পুঁজি তাকে আটবে দেবে। তার মুনাফার স্বার্থে। এতকাল অব্দি মানুষের বিদ্যাচর্চার বিকাশটা ছিল সামগ্রিক, প্রতিটা প্রজন্ম যা করত সেটা পেয়ে যেত পরের প্রজন্ম। সেই বিবর্তন প্রবাহটাকে আটকে দেওয়া হবে। আর সেই আটকে দেওয়াটার প্রভাব আর শুধু কম্পিউটারেই সীমাবদ্ধ থাকবে না, কারণ, আর মাত্র একটা দশকের মধ্যে এমন সময় চলে আসছে যেখানে বিদ্যাচর্চার প্রতিটি এলাকারই জীয়নকাঠি আসলে থাকবে কম্পিউটারের মধ্যে।

১৯৭৯।। চার নম্বর দিনে অনেকটা এবং আজকের আলোচনার শুরুতেও অপারেটিং সিস্টেম বা ওএস-এর যে স্থানান্তরযোগ্যতা বা পোর্টেবিলিটি নিয়ে কথা বলেছি আমরা — সেটা প্রথম বাস্তব হয়ে উঠল উনআশিতে এসে। বেল ল্যাবরেটরি ইউনিক্স ভার্শন সাত। এই গ্রহের প্রথম ওএস যা একাধিক মেশিনে চলল, ডেক-এর পিডিপি সিরিজ এবং ইন্টারডেটার একাধিক মডেলে। আর এই উনআশিতেই, ইউজনিক্স বা ইউনিক্স ইউজার গ্রুপের মিটিং-এ একটা ঘটনা ঘটল যা কয়েক বছরের মধ্যে বদলে দেবে গোটা সফটওয়ার তথা কম্পিউটারের ইতিহাসকে। টরন্টোয় ইউজনিক্স-এর মিটিঙে এটিঅ্যান্ডটি ঘোষণা করল, এখন থেকে তাদের ইউনিক্স-এর দাম পড়বে বিদ্যাচর্চার প্রতিষ্ঠানগুলোয় সিপিইউ পিছু ৭৫০০ ডলার করে, এবং পূর্ণাঙ্গ ব্যবসায়িক লাইসেন্সের জন্যে ষাট হাজার ডলার। মানে আগের প্যারার ওই কুড়ি ডলারের মাত্র তিরিশ হাজার গুণ। এই মিটিঙেই উপস্থিত ছিলেন অধ্যাপক অ্যান্ড্রু ট্যানেনবম, তার প্ল্যান ছিল তার আসছে বছরের ওএস-এর ক্লাসে ছেলেদের ইউনিক্স দিয়ে কাজ করানোর। তার মানে তার, তিরিশ জন ছাত্রের তিরিশটা মেশিনে ইউনিক্স লাগাতে গেলে লাগবে সাড়ে সাত হাজার করে মোট দু লক্ষ পঁচিশ হাজার ডলার মাত্র। ট্যানেনবম তার মিনিক্স সংক্রান্ত কাজ শুরু করলেন, এটিঅ্যান্ডটি-র এই উন্মত্ত লাইসেন্সিং-এর বিপরীতে — ছাত্রদের পড়াতে গেলে না করে কোনো উপায় ছিলনা।

১৯৮০।। আশিতে বেরোলো বিএসডির ভার্শন চার। এর পর থেকে নববই অব্দি চলল বিএসডি-র ভার্শন ৪.১, ৪.১এ, ৪.১বি, ৪.১সি, ৪.২, ৪.২, ৪.৩, ৪.৪ — এই বিদঘুটে নামের পরপর সংস্করণের শোভাযাত্রা। এর কারণও আর কিছু না, সেই এটিঅ্যান্ডটি। এটিঅ্যান্ডটি বার্কলে সফটওয়ারকে অনুমতি দিয়েছিল যে তারা অ্যাকাডেমিক গবেষণার জন্যে ভার্শন চার দিতে পারে, কিন্তু সেটাকে বদলে কোনো নতুন সংস্করণ বাজারে

ছাড়তে পারবে না। তাই বার্কলেকে প্রমাণ করতেই হচ্ছিল যে ওগুলো কোনো নতুন ভার্শন না, জাস্ট একটু আপডেট, ছাঁটকাট। সচরাচর মূল সংখ্যাটার বদল দিয়ে সংস্করণ বদলানো বোঝানো হয়, আর তার সংযোজন সংখ্যা দিয়ে খুচরো বদলগুলোকে।

১৯৮৪।। চুরাশির জানুয়ারিতে আরএমএস তৈরি করলেন তার গ্নু (GNU) প্রোজেক্ট। এবং এই চুরাশিতেই এল গ্নু-ভার্শন বা নতুন সংস্করণ ইম্যাক্স যা আগেরটার চেয়ে আমূল পরিবর্তিত। এবং ইম্যাক্স নিয়ে কিছু বলার লোভ সামলানো শক্ত, কিন্তু তার চেয়েও শক্ত কিছু বলা। সবচেয়ে সহজে যা বলা যায় তা হল, আসুন, গ্নু-লিনাক্স শিখুন। ইম্যাক্স সম্পর্কে একটা চ্যাংড়ামি বরং উল্লেখ করা যাক — নিন্দুকেরা বলে থাকে ইম্যাক্স নামের কমান্ড এডিটরটা আসলে গোপনে গোপনে একটা অপারেটিং সিস্টেম, যাতে খুব সহজসাধ্য একটা কমান্ড এডিটরের মৃদু অভাব আছে।

১৯৮৫।। গ্নুর ম্যানিফেস্টো বা ইশতেহার প্রকাশিত হল ডক্টর ডবস জার্নালে, মে পঁচাশি ইশুতে। এর আগে তিরাশি থেকেই এই ইশতেহারটা লিখছিলেন আরএমএস। যেখানে একটা ফ্রি অপারেটিং সিস্টেমের তৈরির কথা ঘোষণা করা হয়, এবং সেই অপারেটিং সিস্টেমের কাজের জন্যে প্রয়োজনীয় কম্পাইলার ইত্যাদি যন্ত্রপাতি তৈরির প্রয়োজনীয়তার কথাও। পঁচাশির এই লাইসেন্স প্রপীড়িত ইউনিক্স জগতে একটা নাটকীয় বদল আনলো গ্নু-র সি কম্পাইলার কালেকশন বা জিসিসি। যে কোনো ইউনিক্স ব্যবহারকারীর একটা প্রাথমিকতম প্রয়োজন ছিল একটা ভালো সি কম্পাইলার। এবং ঠিক এর আগে আগেই সান মাইক্রোসিস্টেম তার সফটওয়ার বিক্রির রকমটা বদলে ফেলেছিল। আগেই বললাম, ক্রমে নিয়ন্ত্রণটা বিদ্যাচর্চার জগত থেকে চলে যাচ্ছিল পুঁজির হাতে। সান মাইক্রোসিস্টেম তার সমস্ত সফটওয়ার এতদিন একসঙ্গে বেচত, একটা প্যাকেজ হিশেবে। এবার, আরো লাভের আশায়, আলাদা আলাদা ভাবে বিক্রি করতে শুরু করল। এবং, আলাদা ভাবে সি কম্পাইলারের প্যাকেজের জন্যে ধার্য করল একটা কদর্য রকমের চড়া দাম। ফ্রি সফটওয়ার ফাউন্ডেশন, যে নামটা তখনো আসেনি, কিন্তু স্টলম্যানের নেতৃত্বে আন্দোলনটা শুরু হয়ে গেছে, সেই এফএসএফের পালে একটা বড় হাওয়া এল এই জিসিসি দিয়ে। এই পঁচাশিতেই এল জিএনইউ পাবলিক লাইসেন্স বা জিপিএলের ভার্শন এক, আমরা আগের দিন যার আলোচনা করেছি। এখানে সেলাসের একটা খোঁচালো মন্তব্য না-দিয়ে পারছি না। সেলাস নাকি একাধিকবার আরএমএসকে মানে স্টলম্যানকে এই মন্তব্যটা শুনিয়েছে, এবং প্রত্যেকবারই স্বভাবসিরিয়াস আরএমএস এতে ব্যাপক খার খেয়ে গেছে। মন্তব্যটা এই যে, ফ্রি সফটওয়ার ফাউন্ডেশনের পুরষ্কারটা পাওয়া উচিত বিল গেটস-এর, কারণ তার এবং তার জাতীয় র‍্যাপ্টরদের উজবুক লোভ এবং যুক্তিহীন লাইসেন্সিং ছাড়া ফ্রি সফটওয়ার আন্দোলন কোনোদিন তৈরিই হতে পারত না।

১৯৮৬।। এল মিনিক্স।

১৯৮৭।। প্রেন্টিস হল ট্যানেনবম-এর ‘অপারেটিং সিস্টেম’ বইটা ছাপল, যার সঙ্গে এমনিতেই পাওয়া যাচ্ছিল ওই বারো হাজার লাইন কোড, যা দিয়ে তৈরি করে নেওয়া যায় একটা অপারেটিং সিস্টেম। এখন নেট থেকেও পাওয়া যায় মিনিক্স-এর এই কোড। কিন্তু তখনো নেট নেট হয়ে ওঠেনি, আগেই বলেছি।

১৯৯০।। আইবিএম তৈরি করল আইক্স (AIX — **A**dvanced-**I**nteractive-e**X**ecutive), ইউনিক্স-এর আইবিএম সংস্করণ।

১৯৯১।। কম্প-অস-মিনিক্সে লিনাস টরভাল্ডজ-এর সেই ছোট্ট বিনয়ী একটা চিঠি, দুনিয়ার কম্পিউটিং ইতিহাসকেই যা বদলে দিল। আর, খেয়াল করুন, নেট ইতিমধ্যেই নেট হতে শুরু করেছে। অন্তত এর মধ্যেই সেই জায়গাটা তৈরি হয়ে গেছে নেট-এর দৌলতে, যাতে টরভাল্ডজ পোস্ট করা মাত্র তা পলক ফেলতে না-ফেলতেই পৌঁছে যায় অজস্র মানুষের কাছে। পঁচাত্তরে ছিল একশোর নিচে, অগাস্ট ১৯৮১-তে নেট-এ সাইটের সংখ্যা দাঁড়ায় ২১৩, ১৯৯৪-এ যা ৭৫ লাখে পৌঁছে গেছিল। ৯১-এ লিনাস টরভাল্ডজ যখন তার মেল পাঠাচ্ছেন, তখন সাইটের সংখ্যা কম করে পাঁচ লাখ, যাদের কাছে সেই মুহূর্তে চলে যাচ্ছে সেটা। এই অবস্থাটা নিয়ে ভারি মজার কিছু উপাখ্যান আছে ‘কোড লিনাক্স’-এ। এই ৯১-এর ১৯শে ডিসেম্বর চালু হয়

প্রথম লিনাক্স নিউজগ্রুপ, কম্প-অস-লিনাক্স। জানুয়ারি বিরানববই-এ, যে সময়টা আমার নিজের জীবনে খুব গুরুত্বপূর্ণ, মানে যার কখনো পড়ার কথা মাথায় রেখেও এই পাঠমালাটা লিখছি আমি, সে যখন জন্মাচ্ছে, তখন লিনাক্স ব্যবহারকারীর সংখ্যা ঠিক একশোর উপর। ৯১-এর শেষে এসে লিনাক্স সংস্করণ ০.১০ ঘোষিত হল। তখনো এই কুচো অপারেটিং সিস্টেম দিয়ে কাজ করা যেত শুধু এটি হার্ড ডিস্কে, লগ-ইন বলে কিছু ছিলনা, সরাসরি ব্যাশ শেলে ঢুকে যেত। এর মানে কী সেটা আজই দেখব আমরা। ভার্শন ০.১১-এ এসে অবস্থা অনেকটা বদলেছিল, তাতে একাধিক ভাষার কিবোর্ড লাগানো গেল, ফ্লপি ডিস্ক ব্যবহার করা গেল, ভিজিএ ড্রাইভার এল।

১৯৯২।। বিরানবরই-এর উনত্রিশে জানুয়ারি লিনাক্স বিষয়ে আপত্তি এল খোদ অ্যান্ড্রু ট্যানেনবম-এর কাছ থেকে, সেই কম্প.অস.মিনিক্স-এ, বিষয়, লিনাক্স চলেনা, লিনাক্স ইজ অবসলিট। “LINUX … is a giant step back … like taking an existing, working C program and rewriting it in BASIC …”, একটা চালু সি প্রোগ্রামকে ফের ওই বেসিকে লেখার মত — লিনাক্স হল কম্পিউটার মননকে পিছিয়ে নিয়ে যাওয়া। এর মধ্যের উপ্ত গোটা অপমানটাকে বোঝাই সম্ভব না, একজন সিরিয়াস প্রোগ্রামসচেতন মানুষের কাছে বেসিক নামের কম্পিউটার ভাষাটা কতটা অশ্লীলতা সেটা খেয়াল না-করে। বেসিকের কথা তো আগেই বলেছি আপনাদের। এখানেই শেষ হয়নি ট্যানেনবম-এর আপত্তি, আবার এল তিরিশে জানুয়ারি। সকন্টেক্সট এই গোটা তর্কাতর্কিটাই আপনারা পাবেন ওরিলির ভয়েসেস ফ্রম দি ওপন সোর্স রিভলিউশনের অ্যাপেনডিক্স এ-তে, http://www.oreilly.com/catalog/opensources/book/appa.html। অন্য অনেকের মতামতের চেয়ে, স্বাভাবিকভাবেই মিনিক্স প্রণেতা ট্যানেনবমের মতামতের গুরুত্ব টরভাল্ডজ বা অন্যান্য লিনাক্সীদের কাছে অনেক বেশি ছিল। তাই আঘাতের প্রকোপটা ভালোই ছড়িয়েছিল। যাইহোক, ইতিহাসই প্রমাণ করেছে, ট্যানেনবম নয়, ঠিক ছিলেন টরভাল্ডজ। বিরানব্বইয়েই চালু হল লিনাক্সের প্রথম বাণিজ্যিক সংস্করণ — সুজে লিনাক্স। আর এই একই বছরে এক্স-উইনডোজ মানে গুই বা গ্রাফিকাল ইউজার ইন্টারফেস ঢুকল গ্নু-লিনাক্স অপারেটিং সিস্টেমে। এর আগে অব্দি ছিল শুধুই কমান্ড মোড।

১৯৯৩।। লিনাক্সীর সংখ্যা এদিকে বেড়েই চলল। তিরানব্বয়ে এলো আরো দুটো বাণিজ্যিক সংস্করণ রেডহ্যাট আর ডেবিয়ান।

১৯৯৪।। চুরানব্বইয়ে এল আরো একটা কমার্শিয়াল ডিস্ট্রিবিউশন — ক্যালডেরা। এই বছরেই চালু হল প্রথম লিনাক্স জার্নাল।

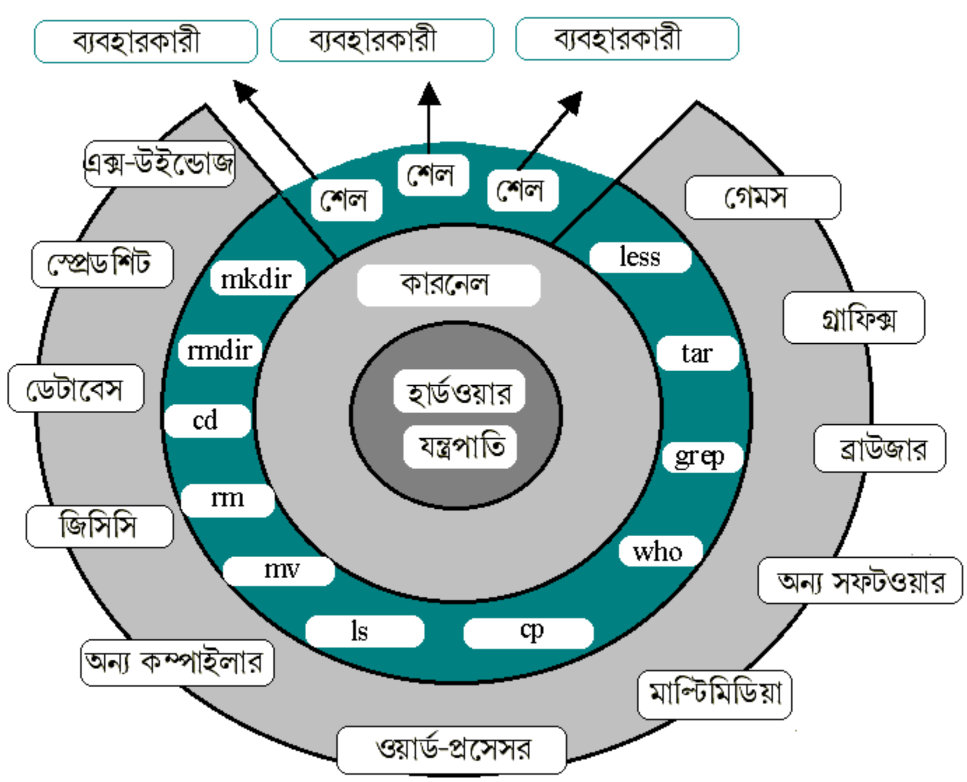
১৯৯৫।। এলো লিনাক্সের কারনেল ১.২, তার সাইজ তখন আড়াইশো কেবির উপর, যেখানে, টরভাল্ডজ-এর প্রথম ঘোষণার সময়, আকারটা ছিল মাত্র ৬৩ কেবি। কারণ, নতুন নতুন ধরণের হার্ডওয়ার, নতুন নতুন গঠনের ড্রাইভার নিয়ে আসতে হচ্ছে কারনেলের ভিতর। গ্নু-লিনাক্সের ব্যবহার তো বেড়েই চলেছে। নতুন নতুন মেশিনে, নতুন নতুন গঠনের মেশিনে।

১৯৯৬।। ফ্রি-তে বন্টনযোগ্য সফটওয়ারের কনফারেন্স হল, কনফারেন্স শুরু হল লিনাস টরভাল্ডজ-এর বক্তব্য দিয়ে, শেষ হল আরএমএসের বক্তৃতায়। এই বছরেই এটিঅ্যান্ডটি মামলা করল ক্যালিফোর্নিয়া বিশ্ববিদ্যালয়ের বিরুদ্ধে, কারণ তারা ইউনিক্সের কপিরাইট মানছে না। এই বছরের এপ্রিল মাসে লস অ্যালামস ন্যাশনাল ল্যাবরেটরি গ্নু-লিনাক্স ব্যবহার করে তৈরি করল ৬৮টা সংযুক্ত পিসির একটা সুপার কম্পিউটার, যাতে পারমানবিক শক ওয়েভের সিমুলেশন বা নকল তৈরি করা যায়। এবং চালু অন্য যে কোনো প্রযুক্তির চেয়ে এতে খরচ পড়ল অনেক অনেক কম। গ্নু-লিনাক্স ক্রমে তার নিজের জায়গা বানিয়ে নিতে শুরু করল। সিরিয়াস প্রযুক্তি, সিরিয়াস বিদ্যাচর্চায় যেমন, তেমনি একা একলা ব্যক্তিগত পিসিতেও।

টরভাল্ডজ-এর তৈরি প্রথম লিনাক্স ফ্রি কারনেলের পর প্রায় দেড় দশক চলে গেছে। এখন সেটা জ্যান্ত গ্নু-লিনাক্স দুনিয়ার প্রাত্যহিকতা। সেটা আর এইরকম লিপিবদ্ধ ক্রমপঞ্জী থেকে নয়, জ্যান্ত রকমে জ্যান্ত কাজ করে জানার। গ্নু-লিনাক্স শিখুন আর জানতে থাকুন, তৈরি করতে থাকুন সেই প্রাত্যহিকতাটা। এবার আমরা যাব সরাসরি একটা গ্নু-লিনাক্স অপারেটিং সিস্টেমের কাজ করার পদ্ধতির বিষয়ে একদম শুরুর আলোচনায়।

৪।। কারনেল আর শেল — অপারেটিং সিস্টেমের শাঁস আর খোসা

আগেও আমরা বলেছি, গোটা ইউনিক্স এবং গ্নু-লিনাক্স কাঠামোয় গোটা প্রক্রিয়াটাকে শাসন করে দুটো গুরুত্বপূর্ণ সত্তা, তারা হল কারনেল আর শেল। শেল কথা বলে সরাসরি ব্যবহারকারীর সঙ্গে। ব্যবহারকারীর প্রয়োজন মত তার পছন্দের প্রয়োগমূলক প্রোগ্রাম বা অ্যাপ্লিকেশনগুলোকে জাগিয়ে তোলে শেল। আগেই বলেছি, একটা প্রোগ্রামের একবার চলার নাম একটা প্রক্রিয়া বা প্রসেস। এদের বলা যায় ইউজার প্রসেস। আবার এই প্রোগ্রামগুলো চালাতে গেলে সিস্টেমের ভিতরে যেসব আভ্যন্তরীণ প্রক্রিয়া চালু হওয়া দরকার, সিস্টেম প্রসেস, তাদেরও চালু করে মূলত শেল। এবার এই ইউজার প্রসেস এবং সিস্টেম প্রসেসদের চলতে গেলে যে যে রসদ প্রয়োজন, সে হার্ডডিস্ক হোক, প্রিন্টার হোক, মেমরি হোক, যাই হোক — সেই প্রয়োজনের তালিকাটা তৈরি করা এবং কারনেলের কাছে পৌঁছে দেওয়াটাও শেলের কাজ। প্রয়োজনীয় এই ভৌত রসদগুলোর উপর কিন্তু শেলের কোনো সরাসরি নিয়ন্ত্রণ নেই। তাকে আর্জি জানাতে হয় কারনেলের কাছে। আর কারনেল কথা বলে সরাসরি হার্ডওয়ার বা ভৌত যন্ত্রপাতির সঙ্গে। শেলের চালু করা প্রক্রিয়াগুলোর জন্যে রসদ ব্যবহারের আর্জি মেটায় কারনেল, সেই সেই রসদের ডিভাইস ড্রাইভারের কাছে আদেশ পৌঁছে দিয়ে — দেখেছি আমরা, দুই নম্বর দিনে। কারনেলের কাছে জানানো আর্জির যা যা ফলাফল, এবং ইউজার প্রসেসদের চলার যে ফলাফল তাদের ফের ইউজারের কাছে পৌঁছে দেয় শেল। মানে মূল কাজটুকু করে ভৌত উপাদানেরা, তাদেরকে নিয়ন্ত্রণ করে কারনেল। কারনেল এবং প্রসেসগুলোর ভিতরে, তথা কারনেল এবং ইউজারের ভিতরে মধ্যস্থতা করে শেল। এখনই গোটাটা স্পষ্ট হবেনা, একটু একটু করে হতে থাকবে, এবং এটাই আবার আমাদের এই পাঠমালার শেষ মানে দশ নম্বর দিনের বিষয়, সেখানে গিয়ে আমরা অনেকটাই বুঝতে পারব। এবার শেল আর কারনেলের এই সামগ্রিক দেওয়া-নেওয়ার ছবিটা দেখুন। একদম কেন্দ্রীয় ছোট্ট বৃত্তটা হার্ডওয়ারের। তাকে ঘিরে আছে কারনেল, শাঁস। গোটা অপারেটিং সিস্টেমের শাঁসটুকু হল কারনেল। তার উপরের স্তরে রয়েছে বিভিন্ন কমান্ড আর শেল। কারনেল যদি শাঁস হয়, শেল হল অপারেটিং সিস্টেমের খোসা। যে খোসার গায়ে ব্যবহারকারীরা মানে আমরা আঁচড় কাটি, বদলাই, কাজ করি।



এখানে আমরা কয়েকটা খুব প্রাথমিক কমান্ডকে উল্লেখ করেছি। ‘mkdir’, ‘rmdir’, ‘cd’, ‘rm’, ‘mv’, ‘cp’, ‘ls’, ‘who’, ‘grep’, ‘tar’, ‘less’ — এরা খুব সাধারন প্রাত্যহিক ছোটখাটো কাজ করে। মেক-ডির বা

'mkdir' নতুন ডিরেক্টরি বানায়। রিমুভ-ডির বা 'rmdir' ডিরেক্টরি ওড়ায় — অবশ্য যদি সেটা ফাঁকা হয়। চেঞ্জ-ডির বা 'cd' এইমুহূর্তে আছি, কাজ করছি যে ডিরেক্টরিতে সেটা বদলায়, মানে একটা ডিরেক্টরি থেকে অন্য ডিরেক্টরিতে যায়। রিমুভ বা 'rm' ফাইল ওড়ায়, বা, অন্যকিছু বাড়তি শর্ত সঙ্গে দিলে ডিরেক্টরিও ওড়ায়। মুভ বা 'mv' ফাইল বা ডাইরেক্টরি এক জায়গা থেকে অন্য জায়গায় নিয়ে যায় বা তাদের নাম বদলায়। লিস্ট বা 'ls' কোনো একটা ডিরেক্টরিতে কী কী ফাইল বা সাবডিরেক্টরি আছে তার তালিকা বা লিস্টটা দেখায়। কপি বা 'cp', ফাইল কপি করে। হু বা 'who' দেখায় কে কে সেইমুহূর্তে সিস্টেমে ঢুকে রয়েছে, কাজ করছে — মানে, টেকনিকালি বললে, লগ-ইন করে রয়েছে। কোনো একটা বিশেষ শব্দ অক্ষর বা তাদের কোনো একটা সমষ্টি কোনো একটা বা একাধিক ফাইলে আছে কিনা সেটা খুঁজে দেয় গ্রেপ বা 'grep'। টার বা 'tar' হল এক বা একাধিক ফাইল বা ডিরেক্টরিকে একটা সিন্দুক ফাইলে পরিণত করার কমান্ড। আর লেস বা 'less' হল একটা পেজার — কোনো একটা ফাইলকে পাতা বাই পাতা পরপর দেখিয়ে যায় লেস। ব্যস্ত হবেন না, আলাদা করে কোনো কমান্ডকে মনে রাখার চেষ্টা করবেন না, প্লিজ। নিজের গ্নু-লিনাক্স বাক্সের সঙ্গে সময় কাটান, গ্নু-লিনাক্সে নিজেকে সংক্রমিত হতে দিন, দেখবেন, আপনা থেকেই মনে থাকবে। বরং মনে রাখার চেষ্টা করলেই টেনশন হবে, তাতে আরো ভুলবেন। আর এত কমান্ড রয়েছে গ্নু-লিনাক্সে যে কাজের মাধ্যমে তাদের সঙ্গে অভ্যস্ত হওয়া ছাড়া মনে রাখার কোনো উপায় নেই। কাজ করতে থাকুন নিজের গ্নু-লিনাক্স সিস্টেমে।

এবার দেখুন, এই কমান্ডগুলো বা ইউজার হিশেবে চালানো যে কোনো প্রোগ্রাম আমরা যখনি চালাচ্ছি, আমরা সরাসরি কথা বলছি শেলের সঙ্গে, ব্যবহারকারী বা ইউজার হিশেবে। তাদের আর কারনেলের মধ্যে যোগাযোগ করাচ্ছে শেল। কারনেল তাদের ভৌত উপাদানের প্রয়োজন মেটাচ্ছে। 'cp' ইত্যাদি খুব সহজ কমান্ডও যদি ধরেন, তাকেও তো একটা ফাইল ডিস্ক থেকে পড়তে হচ্ছে, এবং তার একটা প্রতিরূপ ডিস্কে লিখতে হচ্ছে। তার মানে ন্যূনতম হলেও ডিস্ক আর মেমরি এই দুটো ভৌত রসদ তাকে কাজে লাগাতেই হবে। রসদের এই প্রয়োজনটা মেটায় কারনেল। সবশেষে আমাদের দেওয়া কাজের ফলাফল আবার আমাদের কাছে পৌঁছে দেয় শেল। ধরুন, আমরা টাইপ করার সময় ভুল নাম দিয়েছি। এবার, শেল তো সেটা জানেনা, সে কপি করার প্রোগ্রামকে জাগিয়ে তুলল। কপি করার প্রসেস ডিস্ক পড়তে চাইল, কারনেল ডিস্ক ড্রাইভারকে জাগাল, ডিস্ক পড়ে দেখা গেল ওই নামে ওখানে কোনো ফাইলই নেই। তখন শেল আবার আমাদের কাছে ফেরত এল, এবং আমাদের জানাল, কপি করা যায়নি, কারণ, যে নামের ফাইল কপি করতে চাইছি, সেই নামে ডিস্কে কোনো ফাইলই নেই। আর যদি ওই নামে কোনো ফাইল থাকে, তাহলে কিছুই বলল না। গ্নু-লিনাক্সে কাজ করার সময় এটা খেয়াল রাখবেন। শেল কিন্তু অত্যন্ত গম্ভীর রিজার্ভ বাকসংযমী। কোনো কাজের কমান্ড দেওয়ার পর কিছুই না জানানো মানে কাজটা নির্বিঘ্নে হয়ে গেছে। এই গোটাটাকেই দেখানো আছে ছবিতে।

একটু বিশদ করে বোঝার চেষ্টা করা যাক এবার, ইউজার এবং শেলের মধ্যের সম্পর্কটাকে। তার জন্যে প্রথমে আমরা কমান্ড দেওয়ার প্রক্রিয়াটাকে বুঝি, আসুন। ধরুন আপনি প্রবন্ধ লিখছেন। প্রথমে আপনি নাম দিয়েছিলেন 'essay.text', তখনো ভেবেছিলেন, একটাই প্রবন্ধ হবে। এবার দেখলেন এই নিয়ে আপনার আরো প্রবন্ধ লেখার আছে। কী নাম দেবেন, নতুন করে ভাবলেন। এবার ভাবলেন নাম দেবেন এক দুই করে, 'essay.1.text', 'essay.2.text' ... ইত্যাদি। এবার, পুরোনো প্রথম প্রবন্ধটার নাম আপনাকে বদলাতে হবে। এর জন্যে কমান্ড বা আদেশ দিলেন, '`mv essay.text essay.1.text`'। কিন্তু আপনার লেখা তো থেমে নেই, এবার মনে হল এই প্রবন্ধের সিরিজে আর কয়েকটা লিখলে তো একটা বই হতে পারে, তাই, এই সমস্ত প্রবন্ধ ফাইলগুলো আলাদা একটা ডিরেক্টরিতে আনবেন ঠিক করলেন, যার নাম দেবেন 'book.one'।

আপনি কমান্ড দিলেন, '`mkdir book.one`'। অ্যাদ্দিনের সব প্রবন্ধগুলোকে এবার ওই ডিরেক্টরিতে পাঠাতে হবে, কমান্ড দিলেন '`mv essay*text book.one`'। এই *-এর চক্করটা একটা রেগুলার এক্সপ্রেশন, মানে যে বঁড়শিতে একসঙ্গে অনেক মাছ তোলা যায় এমন একটা পাইকিরি হুক, দশ নম্বর দিনে খুব ভালো করে আসব আমরা এটায়। *-এর আগের আর পিছের দুটো অংশের গঠনটা আপনি দিয়েছেন, মধ্যে *-এর জায়গাতে যাই থাকনা কেন, সেই নামের ফাইল পাঠিয়ে দেওয়া হবে ওই 'book.one' ডিরেক্টরিতে। এক জায়গায় আপনার সব প্রবন্ধের ফাইলগুলো পাওয়া যাবে।

এবার ধরুন, বইটা শেষ হয়েছে, আপনার এক বন্ধুকে পাঠাবেন বইটা, মেল করে, কিন্তু মাথায় আছে, টেলিফোন বিল, মাসের শেষের অশান্তি, অর্থনৈতিক ধ্বস ইত্যাদি। আপনি তাই গোটা ডিরেক্টরিটাকে একটা সিন্দুক বানিয়ে যারপরনাস্তি ছোট করতে চাইলেন। কমান্ড দিলেন '`tar -cjf book.one.tar.bz2 book.one`'। এতে বইয়ের গোটা 'book.one' ডিরেক্টরিটা সিন্দুক হয়ে ছোট্ট হয়ে অনেক বাইট সাইজ কমে নতুন একটা ফাইল তৈরি হল যার নাম 'book.one.tar.bz2'। এইটা ইমেল করে অনেক কম খরচেই পাঠাতে পারবেন বন্ধুকে। সে আবার পড়ার আগে উল্টো একটা কমান্ড দিয়ে সেটাকে ভেঙে নেবে। এই শেষ কমান্ডটায় আমরা আসলে দুটো কমান্ডকে একই সঙ্গে ব্যবহার করেছি, একটা হল 'tar', আগেই বলেছি, আর অন্যটা হল 'bzip2'। পরে শিখে যাবেন, গ্নু-লিনাক্স করতে করতে, না-শিখে মেলে বই পাঠাতে গেলে আপনার বাড়ি থেকে আপনাকে নির্দ্বিধায় 'কেয়ার অফ ফুটপাথ' করে দেবে। এই দুটো কমান্ডের কোনো একটাকে জানার যদি খুব কৌতূহল হয়, তাহলে আপনার গ্নু-লিনাক্স সিস্টেমে দাঁড়িয়ে কমান্ড দিন, '`man tar`' বা '`man bzip2`'। সিস্টেম আপনাকে এই দুটো কমান্ডের ম্যানুয়াল পাতাটা খুলে দেবে। এরকম যে কোনো কমান্ডেই আপনি পড়ে নিতে পারেন। শুধু পারেন না, পড়তে আপনাকে হবেই, যদি কাজ করতে চান।

এর প্রত্যেকটা কমান্ড আপনি কিবোর্ড থেকে টাইপ করে দিয়েছেন, কমান্ড প্রম্পটে। কমান্ড প্রম্পট (Command Prompt) মানে টাইপ করার জায়গা। যদি আপনি এক্স-উইনডোজ বা গুই-তে থাকেন, তাহলে একটা টার্মিনাল অ্যাপ্লিকেশন উইনডো। আর যদি এমনিতে থাকেন, তাহলে সিস্টেম বুট হওয়ার পর কালো যে জায়গাটা ফুটে ওঠে, কার্সর জ্বলতে নিভতে থাকে, আপনি কিছু টাইপ করলে সেটা যেখানে ফুটে ওঠে। মেশিনকে কোনো আদেশ বা কমান্ড দেওয়া মানে এই কমান্ড প্রম্পটে কমান্ডটা টাইপ করা, তারপর এন্টার (Enter) সুইচটা টিপে সেই আদেশটা সিস্টেমে ঢোকানো। যেই আপনি আদেশটা দিয়ে এন্টার মারেন, সঙ্গে সঙ্গে ওই জায়গাটা হাওয়া হয়ে যায়, কমান্ড প্রম্পটটা। যতক্ষণ সে গায়েব থাকছে আপনার আর কিছু করার নেই। কমান্ড প্রম্পট গায়েব মানেই সিস্টেম কাজ করছে। আপনার মেশিন কাজ করছে, জোনাকির মত আপনার মেশিনের টাওয়ারের গায়ে লাগানো হার্ড ডিস্ক কাজ করার কুচো বাল্বটা কিছু সময় ধরে তার কুতকুতে চোখ টিপটিপ করে চলেছে। তারপর আবার আপনার কমান্ড প্রম্পটটা ফেরত এল। মানে, মেশিনের কাজ করা শেষ হয়েছে, আবার আপনি নতুন আদেশ দিতে পারেন। তবে ছোটখাটো কাজ এত দ্রুত হয় যে এই ভ্যানিশ করে যাওয়া এবং ফেরত আসাটা খুব খেয়াল না-করলে বোঝাই যায়না। তার মানে দেখুন, আপনার কমান্ডটা পালন করতে গিয়ে আপনার মেশিনকে কিছু কাজ করতে হল তার ভৌত রসদদের নিয়ে, সে হার্ডডিস্ক, অন্য কোনো ডিস্ক, মেমরি, ইনপুট-আউটপুট, যাই হোক।

একটু আগের সিন্দুক বানানোর কমান্ডটা ভাবুন, 'tar -cjf book.one.tar.bz2 book.one'। আপনার আদেশে কী ছিল? 'book.one' ডিরেক্টরিতে আপনার কিছু প্রবন্ধ, 'essay.1.text', 'essay.2.text' ... ইত্যাদি নামের কিছু ফাইল। যার প্রত্যেকটা ফাইলে আপনার একটা করে প্রবন্ধ আছে। মানে আসলে আছে কিছু বাইট, সিস্টেম যাদের বর্ণমালা এবং যতিচিহ্ন করে আমাদের সামনে হাজির করেছে। এবার, 'book.one' নামের গোটা ডিরেক্টরিটাকে তার যাবতীয় ফাইল সমেত 'tar' নামের প্রোগ্রাম দিয়ে প্রথমে একটা সিন্দুক ফাইল বানানো হচ্ছে। ডিরেক্টরিময় সমস্ত ফাইল তখন সেই একটামাত্র ফাইলে পরিণত হল। এবার সেই সিন্দুকটাকে আবার ছোট করে একটা কোঁকড়ানো সিন্দুক বা কমপ্রেসড টার ফাইল বানানো হচ্ছে। 'book.one.tar.bz2' নামের সেই কোঁকড়ানো সিন্দুকটাকে ডিস্কে লিখে ফেলার পর তবে আপনার কমান্ডের মোট কাজ শেষ করতে পারবে আপনার সিস্টেম।

এই গোটাটা জুড়ে আপনার সিস্টেম কী কী করছে? প্রথমে, গোটা ডিরেক্টরিটা সবগুলো ফাইল সমেত পড়তে হচ্ছে। পড়া মানে কী? ওই সমস্ত ফাইলের সমস্ত বাইটের প্রতিরূপ তুলে মেমরিতে রাখা। এবার, 'tar' আর 'bzip2' প্রোগ্রামদুটো সিস্টেম কাজে লাগাতে গেলে আগে গোটা প্রোগ্রামদুটোর মধ্যে লেখা আদেশের বাইটগুলোকে মেমরিতে তুলে নিতে হবে, তারপর চালাতে হবে। তখন দুটো আপাতমৃত প্রোগ্রাম ফাইল জ্যান্ত হবে, তাদের আমরা ডাকব দুটো প্রসেস বলে — মনে পড়ছে? তারপর 'tar' আর 'bzip2' নামক প্রোগ্রামের আদেশতালিকা মোতাবেক ইতিমধ্যেই মেমরিতে তুলে নেওয়া 'book.one' ডিরেক্টরির যাবতীয় ফাইলের বাইট সমগ্রকে তুমুল চটকানো। চটকাতে চটকাতে যখন গোটাটা যখন ডিস্কে বেলে ফেলার মত জায়গায় আসবে তখন চটকানো শেষ। মেমরিবাসী সেই চটকে ফেলা বাইটসমগ্রকে এবার ডিস্কে পাঠানো, 'book.one.tar.bz2' ফাইলে লিখে ফেলার জন্যে, তবে কাজ

শেষ মেশিনের। আসলে কিন্তু জনান্তিকে, পর্দার পিছনে আরো বহু বহু কিছু ঘটছে, কিছুটা যার আন্দাজ আমরা আগেই দিয়েছি সিস্টেমের ডিমন বা যখদের কথায়, কিম্বা সিস্টেম প্রসেসের প্রসঙ্গে। এখানে আমরা সেটাকে ছোট করে নিলাম, জঙ্গল দেখতে গিয়ে যাতে গাছ হাওয়া না-হয়ে যায়। অর্থাৎ, আপনার মেশিনের ভৌত রসদগুলোর ভিতর অন্তত হার্ডডিস্ক আর মেমরি তো ব্যবহার করতেই হবে, আপনার আদেশ পালন করতে গেলে। কিন্তু।

কিন্তু আপনি কখনোই সরাসরি কারনেলের সঙ্গে, মানে সব রসদের ভান্ডারির সঙ্গে কথা বলেননি, বলেছেন শেলের সঙ্গে। মিডলম্যান হয়ে শেলের এই কাজটা খুব স্পষ্ট বোঝা যায় আপনার ওই অনেক মাছ একত্রে গাঁথার বড়শি মানে রেগুলার এক্সপ্রেশনটা দেখলে। আপনি তো বঁড়শিটা লাগিয়েই খালাস হয়েছিলেন, তার সঙ্গে মিলিয়ে মিলিয়ে কোন কোন মাছ ওই বঁড়শিতে গাঁথা হবে, আর কোনগুলো পরের বারের জন্যে রেখে দেওয়া হবে সেটা বলে দিল কে? আগেই তো বলেছি আমরা কারনেল হার্ডওয়ারের উপরে ছেয়ে দেওয়া একটা প্রলেপের মত, ওইসব তার কাজ নয়, তার কাজ ভৌত উপাদানদের নিয়ন্ত্রণ। এই কাজটাই করে শেল। এবং গ্নু-লিনাক্সে বেশির ভাগ ক্ষেত্রেই এই শেলটার নাম ব্যাশ (BASH — **B**ourne-**A**gain-**SH**ell) — কিন্তু এছাড়াও কর্ন শেল, জেড শেল, ইত্যাদি অনেকরকম হতে পারে।

আমরা বলেছিলাম, এই কমান্ডগুলো বা তাদের কোনো সমবায়, যেমন একটা খুব সরল সমবায় এই মাত্র আমরা দেখলাম, 'tar' আর 'bzip2', কাজ করে কারনেলের উপর দাঁড়িয়ে। আর শেল তৈরি করে দেয় আমাদের আর কমান্ডের মধ্যে কথা বলার জায়গাটা। আমরা এককথায় সুইপ করে বেরিয়ে গেলাম — এই কমান্ডগুলো বা তাদের কোনো সমবায় — এখানে পরে আমাদের বিপুল ভাবে ফিরে আসতে হবে, বারবার, এই সমবায় ধারণাটা হল ইউনিক্স লিনাক্স-এর সবচেয়ে জোরের জায়গার একটা, তার দার্শনিক ভিত্তিগুলোর একটা। আমরা যখন ইউনিক্সের ক্রমপঞ্জীতে প্রথম দিকের ইউনিক্সে পাইপ না-থাকার কথা বলেছিলাম, সেই পাইপ হল যাদের দিয়ে এই সমবায়টা করা হয় তাদের মধ্যে একটা জোরালো অস্ত্র।

আপনারা যারা উইন্ডোজ পরিবেশে কাজ করতে অভ্যস্ত, সেখানে দেখবেন, কোনো সফটওয়ার ভালো হচ্ছে উন্নত হচ্ছে মানেই আরো গাবদা হচ্ছে, বিএমআই যাচ্ছে পঁচিশ ছাড়িয়ে, কোমর যাচ্ছে পঁয়ত্রিশ ছাড়িয়ে, ভুঁড়ি বেড়ে যাচ্ছে বেল্ট ছাড়িয়ে — মানে আরো আরো পোকা বা বাগ গজাচ্ছে। এর ঠিক উল্টো দর্শন থেকে শুরু হয়েছিল ইউনিক্স-এর যাত্রা, গ্নু-লিনাক্স তাকে আরো বাড়িয়ে তুলেছিল। সেটা হল, যে কোনো একটা কমান্ড খুব ছোট একটা পরিসরে কাজ করবে, কিন্তু সেই কাজটুকু করবে অপ্রশ্নেয় দক্ষতায়। আর আরো আরো জটিল কাজ, বড় কাজ, ঘুরপথে কাজ, বেঁকা কাজ মানে জাস্ট ওই কুচো কুচো কিন্তু ধারালো কমান্ডগুলোকে নাও, তাদের একটার সঙ্গে অন্যটাকে জোড়ো, আর একটাকে, আর একটাকে, প্রয়োজন পড়লে জুড়েই চলো। এবার যেটা পাওয়া যাবে সেটা বড় কাজই করবে, কিন্তু একা একটা কোনো কমান্ড দিয়ে না, কমান্ডের সমবায় দিয়ে। এই মুহূর্তে আপনাদের কারোর কারোর মনে হতে পারে, এদুটোর মধ্যে এমন বিরাট পার্থক্যটা কোথায়, কিন্তু সেই উত্তরটার জন্যে আমাদের এগোতে হবে আরো অনেকটা।

ধরুন, একটা প্রাথমিক আন্দাজ দিয়ে নেওয়া যাক, এর গোটাটা আপনি এই মুহূর্তে নাও বুঝতে পারেন, চিন্তিত হবেন না, ধারণাটা বোঝার চেষ্টা করুন। পরে এর সবগুলো উপাদানই আমরা অনেক ভালো করে আনব। একটু আগে আমরা যে কমান্ডটা দিয়ে আপনার আস্ত বইটাকে একটা কোঁকড়ানো সিন্দুক বানিয়েছিলাম, সেই কমান্ডটাই ভাবুন,

```
tar -cjf book.one.tar.bz2 book.one
```

এর মধ্যে '`j`' অংশ টুকু আসলে বিজিপটু-র কাজ করছে। এই কমান্ডটাকে আমরা এইরকম ভেঙেও লিখতে পারতাম,

```
tar -cf - book.one | bzip2 > book.one.tar.bz2
```

এর মধ্যে দেখুন তিনটে আলাদা অংশ আছে, '`tar -cf - book.one`' এবং '`bzip2`' এবং '`book.one.tar.bz2`'। এখানে আমরা এই সমবায় টেকনোলজির দুটো জায়গা ব্যবহার করেছি, পাইপ এবং রিডাইরেকশন, পথ-দেখানো এবং চালান করা। প্রথমটার চিহ্ন '`|`' আর দ্বিতীয়টার চিহ্ন '`>`'। প্রথম চিহ্নটার বাঁ পাশের অংশটা book.one নামক ডাইরেক্টরিকে '`tar -cf -`' করে সিন্দুক বানাতে বলছে। এই অব্দি করে আমরা যদি ছেড়ে দিতাম তাহলে স্ক্রিনময় হুড়হুড় করে নেমে যেতে থাকত ওই রকম সব চিহ্ন, শূন্য নম্বর দিনে ভিশুয়াল চ্যাংড়ামিতে আমরা যা দেখিয়েছিলাম একটা পিডিএফ ফাইলকে টেক্সট এডিটর দিয়ে খুলে। কিন্তু স্ক্রিনে

ওই অপদেবভাষা দেখে আমাদের আর কী পুণ্য হবে? তাই বেচারাদের আমরা পথ দেখালাম, পাইপ করলাম, পাঠিয়ে দিলাম দ্বিতীয় অংশের কাছে, যেখানে সেই সিন্দুকটাকে কুঁকড়ে ফেলবে। কোনো আসুরিক শক্তি দিয়ে নয়, তার কায়দা আছে অনেক, এবং বেশ মজার কায়দা গুলো। ধরুন 'কালো ভালো আলো মালো' এদের আপনি লিখলেন 'কাভাআমা৪লো'। এটা একটা জঘন্য এবং তড়িঘড়ি, কুইক অ্যান্ড ডার্টি, উদাহরণ হল। কিন্তু আন্দাজটা পেলেন বোধহয়? যাইহোক, কোনো অজঘন্য এবং শ্লথ পদ্ধতিতে সিন্দুকটাকে কোঁকড়ানো হল। এই গোটাটাই ঘটল মেমরিতে। সরাসরি কাঁচা মেমরিতে বা ভারচুয়াল মেমরির অস্থায়ী ফাইলে। তারপর মেমরিবাসী বাইটগুলোকে চালান করা হল, রিডাইরেক্ট করা হল হার্ডডিস্কের নির্দিষ্ট কিছু সেক্টরে, যাদের জন্যে নাম সাব্যস্ত করে দেওয়া হয়েছে আগেই, 'book.one.tar.bz2'। এই গোটা প্রক্রিয়াটাকে আমরা শর্টকাটে আর একটা রকমে লিখেছিলাম

```
tar -cjf
```

যাইহোক, এখুনি এর গোটাটা বোঝা সম্ভব না, কিন্তু আন্দাজটা তৈরি করে নিন। যাতে কারনেল এবং শেলের গোটা কাজের পদ্ধতিটাকে বুঝতে পারেন।

আর ইউনিক্স জগতের সেই বিখ্যাততম পাইপ প্রদর্শনীটা তো আছেই, যেখানে আপনি সিস্টেমে আপনি সহ মোট কজন এই মুহূর্তে কাজ করছেন সেটা বোঝার জন্যে দিচ্ছেন

```
who | wc -l
```

এর মধ্যে 'who' অংশটা প্রথমে তৈরি করছে তার কাজের ফলাফল, মানে যারা সিস্টেমে কাজ করছে বা লগ-ইন করে রয়েছে তাদের তালিকা। 'who' তালিকাটা তৈরি করে লাইন পিছু একজন করে ইউজার বা ব্যবহারকারীকে রেখে, সেটাই 'who' প্রোগ্রামের নিয়ম। এবার 'who' প্রোগ্রাম তার কাজের ফলাফলটাকে পাঠিয়ে দিচ্ছে দ্বিতীয় অংশের কাছে, যেখানে 'wc -l' অংশটা তার লাইন গুনছে। যে কোনো টেক্সট ফাইলের উপর 'wc' চালালে, 'wc' ফাইলটার মোট লাইনের সংখ্যা, শব্দের সংখ্যা, এবং চিহ্নের সংখ্যা পরপর ফুটিয়ে তোলে। আর 'wc -l' দেখায় শুধু লাইনের সংখ্যা। এবার, 'who' প্রোগ্রামের ফলাফলটার লাইন গোনা মানে, আপনি যখন দ্বিতীয় অংশটার কাজের ফলাফল পাচ্ছেন, সেটা একটা সংখ্যা, ব্যবহারকারীদের মোট সংখ্যা। শুধু 'who' করলে পাওয়া তালিকা থেকে যা আপনি নিজেই গুনে নিতে পারতেন। এই গোটা কাজটাই ভাবুন। করে দিচ্ছে শেল, আপনার হাত থেকে গোটা কমান্ডটা সে পড়ে নিচ্ছে, তারপর তার মধ্যের বিভিন্ন অংশকে আলাদা আলাদা ভাবে ঘটিয়ে নিচ্ছে। ঘটাতে গিয়ে তাকে অনেক কিছু করতে হচ্ছে, কারনেলের কাছে আবেদন ইত্যাদি, কারণ, কাজ করা মানেই তো রসদ ব্যবহার করা, তারপর মোট কাজের ফলাফলটাকে ফের প্রোগ্রামগুলোর কাছ থেকে ফিরিয়ে আনা এবং স্ক্রিনে ফুটিয়ে তোলা। এই গোটা পদ্ধতিটা আজ আমরা প্রাথমিক রকমে বুঝব। আট নয় দশ নম্বর দিনে আরো অনেক কিছু জানার আছে।

কারনেল এবং তার চারদিকে কাজ করতে থাকা শেল আর কমান্ডগুলোকে মিলিয়ে তৈরি হয় অপারেটিং সিস্টেম। তার উপর দাঁড়িয়ে আছে প্রয়োগ সফটওয়ারগুলো, যেমন, গুই বা এক্স-উইনডোজ — সেটা নিজেও একটা অ্যাপ্লিকেশন, আবার তার উপর দাঁড়িয়ে কাজ করে অনেক অ্যাপ্লিকেশন। অন্য প্রয়োগ সফটওয়ারগুলোর ভিতর খুব বেশি ব্যবহার হয় স্প্রেড-শিট বা হিশেবনিকেশের সফটওয়ার, ডেটাবেস বা কোনো রাশি বা তথ্যের সমাহারকে নাড়াচাড়া এবং ব্যবহার করার সফটওয়ার, জিসিসি মানে আমরা জানি — সি বা সিপ্লাসপ্লাস জাতীয় হাইলেভেল ভাষায় লেখা প্রোগ্রামকে লোলেভেল মেশিনোপযোগী করে তোলার কম্পাইলার, ওয়ার্ড-প্রসেসর — যা দিয়ে আমরা লিখি, এই লেখাটাও লিখছি, গ্রাফিক্স — যা দিয়ে ছবি আঁকি, একটু আগে শেল আর কারনেলের ছবিটা আঁকলাম, ব্রাউজার মানে যা দিয়ে নেটে আমরা একটা ওয়েবসাইট থেকে আর একটা ওয়েবসাইট টুঁড়ে বেড়াই — এত সব যে লাগাতার পন্ডিতি মেরে যাচ্ছি তার খনি তো ওই নেটই, বা, মাল্টিমিডিয়া মানে গান-সিনেমা-অ্যানিমেশন ইত্যাদি।

আগের ছবিতে দেখুন শেল কারনেল আর হার্ডওয়ারে বিন্যস্ত কাজের বৃত্তাকার ছক, এবং এর সঙ্গে একবার মিলিয়ে নিন এক নম্বর দিনের সেই টেবিলটা, যেখানে আমরা অপারেটিং সিস্টেমের তিনটে স্তরে স্তরবিন্যাসটা দেখিয়েছি — কারনেল আছে হার্ডওয়ারকে প্রলেপের মত ঘিরে, অপারেটিং সিস্টেমের কেন্দ্রীয় জায়গায়। এই কারনেল হল, গ্নু-লিনাক্সের ক্ষেত্রে, সি-তে লেখা কিছু কোডের সমষ্টি। যারা সরাসরি হার্ডওয়ারের সঙ্গে কথোপকথন চালায়। যার পাগলা কুকুরে কামড়ানো রকমের দামী ভার্শন বেচে আসছিল এটিঅ্যান্ডটি, আর যার প্রথম ফ্রি সংস্করণটা নেটে

তুলেছিল লিনাস টরভাল্ডজ, মাত্র তেষট্টি কেবির। এই ফ্রি মানে শুধু দামের ফ্রি নয়। সেটা প্রথম অর্থ, যে কেউ যে কেউকে এই সফটওয়ার দিয়ে দিতে পারে, কোনো আইন ভাঙা হয়না তাতে। আর দ্বিতীয় অর্থটা অনেক বেশি ব্যাপক। যে কেউ, যে বুঝতে চায় কারনেলকে, সিস্টেমকে, সে এবার বুঝতে পারবে, কোড গুলো পড়ে পড়ে, কী ভাবে সেটা কাজ করছে, কী ভাবে আদেশমালার মধ্যে দিয়ে প্রোগ্রামগুলো চলছে — কী করে কাজ করে একটা কম্পিউটার সিস্টেম। বুঝতে পারবে, ধরতে পারবে, নতুন ধরনের সিস্টেমে নতুন ধরনের কাজের স্বপ্ন দেখতে পারবে। নতুন চিন্তাকে নতুন ধরনের প্রকৌশলকে নিয়ে আসতে পারবে, যদি তার সেই আনার যোগ্যতা থাকে। মানে, জানার মুক্তি, ফ্রিডম অফ নলেজ। গেটস-এর ভাড়াটে কম্পিউটার বিজ্ঞানীরা যে অব্দি গিয়ে পৌঁছতে পেরেছে সেটাই মানুষের চিন্তার সীমা বানিয়ে তোলার স্পর্ধা বিল গেটস করতে পেরেছে, কারণ, বাজার ওভাবেই কাজ করে। সেই সীমার বাইরে মানবসভ্যতার আবহমান জানার স্বাধীনতার অর্থে ফ্রি। আমরা তিন নম্বর দিনে যে কম্পিউটার যন্ত্র এবং চিন্তার যে ধারাবাহিকতাটাকে এনেছি, তাকে যে কোনো একটা জায়গায় ভাঙুন, দেখুন পরের টুকু আর আসছে না। এতদিন অব্দি সেটা এসে থেকেছে, এভাবেই, রেডমন্ডের র‍্যাপ্টর এবং তার স্বজাতীয়দের আগে অব্দি। এখন আমাদের চিন্তার সীমা হয়ে দাঁড়াচ্ছে গেটসের গেট দিয়ে যতটুকু আকাশ দেখা যায়। তার বাইরে যাওয়ার ফ্রিডমটা হল ফ্রি-এর দ্বিতীয় অর্থটা। ফ্রি বিয়ারের ফ্রিডমটাও উমদা জিনিষ, কিন্তু জানার ফ্রিডমটা বেহাত হয়ে যাওয়াটা আরো ডেঞ্জারাস। হঠাৎ একদিন দেখা যায়, হোজে মার্তির সেই কবিতার মত, পিট সিগারের গানের মত, কবে যেন আকাশ হাওয়া জঙ্গলটাও বেহাত হয়ে গেছে।

সিস্টেম যখন বুট করে, মানে মৃত একটা টিভি স্ক্রিনের মত মনিটর আর একটা বড় লম্বা টিনের কৌটোর ক্যাবিনেটে ঘেরা টাওয়ারের নিষ্প্রাণতা থেকে জ্যান্ত কাজমান একটা অস্তিত্ব হয়ে ওঠে, প্রাণ পায়, সেই সময়ে মেমরিতে কারনেলের এই সি-কোডগুলো তুলে নেওয়া হয় একের পর এক। পরে বুট-পদ্ধতির আলোচনায় আমরা আরো ভালো করে বুঝব। সিস্টেমের মোট রসদের ভাণ্ডারটাকে ম্যানেজ করে কারনেল, বিভিন্ন ব্যবহারকারীর চালানো বিভিন্ন প্রোগ্রাম কে কখন কার পরে প্রসেসরকে কাজে লাগাবে — সেটা ঠিক করে, এবং হার্ডওয়ারের যেসব বিটকেল খুঁটিনাটি জেনে কাজ করতে হলে আমাদের বরং কিবোর্ড না-টিপে ডুগডুগি বাজাতে হত সেগুলোকে ত্রুটিহীন রকমে মাথায় রেখে নিঃশব্দে আমাদের কাজ করার ব্যবস্থা করে চলে। বিভিন্ন ধরনের মাল্টিপ্লেক্সিং নিয়ে এক আর দুই নম্বর দিনে আমাদের আলোচনাটা মনে করুন। বিভিন্ন প্রায়োগিক সফটওয়ার বা অ্যাপ্লিকেশনগুলো তাদের প্রয়োজন মোতাবেক বিভিন্ন রসদের জন্যে আবেদন জানায় — সেটা এই কার্নেলের কাছেই — তারই নাম সিস্টেম কল, আমরা আগেই বলেছি। কার্নেল হল বিভিন্ন প্রোগ্রামের মেশিনে ঢোকার প্রবেশপথ।

৫।। ভৌত উপাদানকে ভুলবেন না

উইন্ডোজ থেকে গ্নু-লিনাক্সে কম্পিউটারটা বদলায় না, তার অঙ্গপ্রত্যঙ্গ বদলায় না, বদলায় তার অপারেটিং সিস্টেম, বদলায় ওই অঙ্গপ্রত্যঙ্গ গুলোকে মানে হার্ডওয়ার ডিভাইসগুলোকে অপারেটিং সিস্টেম কী ভাবে নাড়াচাড়া করে তার ব্যাকরণটা। তার সঙ্গে সঙ্গে বদলায় নামকরণটাও, দেখার পদ্ধতিটাও। যার একটু আধটু আমরা আগেই বলেছি। ভালো করে জানব ছয় আর সাতে গিয়ে। এবার, দেখে নিন, এই অ্যানাটমিটা আপনার মনে আছে কিনা। আমরা শূন্য নম্বর দিনে এর একদম প্রাথমিক পরিচয়টা করেছিলাম, মেশিনের কাজের ছক অনুযায়ী, যা মূলত ভন নয়মান আর্কিটেকচার প্রাথমিকতম ছকটা। তারপর এক আর দুই নম্বর দিনেও এসেছে, সিস্টেম কল এবং ইন্টেরাপ্ট আলোচনার সূত্রে। পিছিয়ে গিয়ে একবার দেখে আসুন। আমরা বুঝতে চাইছি গ্নু-লিনাক্স একটা কাঠামোর কাজের পদ্ধতি, গ্নু-লিনাক্স একটা ওএস ঠিক কী ভাবে কাজ করে সেটা বুঝতে চাই আমরা। সেটা প্রধানত একটা সফটওয়ার জগতের যাত্রা। কিন্তু বারংবার সেখানে দরকার পড়ে ভৌত বাস্তবতাটাকে মাথায় রাখার।

আমরা যে তালিকাটার সঙ্গে পরিচিত হয়েছি, তার মধ্যে প্রধানতমটা অবশ্যই মাইক্রোপ্রসেসর চিপ। এর পরেই কাজের গুরুত্বে দ্বিতীয় জায়গায় আসে মেমরি। এখানে মেমরি বলতে র‍্যাম চিপ এবং রম চিপ দুই-ই বোঝাচ্ছি আমরা। প্রসেসর চিপ আর মেমরি চিপ মিলে তৈরি হয় কম্পিউটারের হৃদপিণ্ড মানে, কম্পিউট করার বা তথ্য চটকানোর মূল কাজটা যে করে। নানা ধরনের অ্যাডাপ্টার কার্ড, বৈদ্যুতিক সংযোগ ইত্যাদি সহ এই চিপগুলো লাগানো থাকে মেশিনের মাতৃক্রোড়ে, মানে মাদারবোর্ডে। তথ্য এবং বিদ্যুতের সংযোগে এদের সঙ্গে যুক্ত থাকে

হার্ডডিস্ক এবং ফ্লপিডিস্ক এবং কম্পাক্টডিস্ক বা সিডি গোছের স্মৃতি তুলে রাখার ভাঁড়ার। প্রয়োজন মত সেখান থেকে লিখে রাখা তথ্য পড়ে ফেলাও যায়। চটকে ফেলা তথ্য মেশিন থেকে বার করার জন্যে আছে মনিটর বা কনসোল বা স্ক্রিন এবং প্রিন্টার এবং সাউন্ডবক্স গোছের আউটপুট ডিভাইস। ওই তথ্য-ভাঁড়ার দিয়েও এই কাজটা করা যায়। আর মেশিনে তথ্য ঢোকানোর জন্যে থাকে কিবোর্ড এবং মাউস জাতীয় ইনপুট ডিভাইস। বা কোনো তথ্য-ভাঁড়ার যা সেই মূহূর্তে ইনপুট ডিভাইস হিশেবে ব্যবহার হচ্ছে।

এর বাইরেও আরো বহু কিছু থাকে মেশিনে। যেমন এই ডিভাইসগুলোর অনেকের জন্যেই লাগে কন্ট্রোলার কার্ড — যার মাধ্যমে এই ডিভাইসকে ব্যবহার করে অপারেটিং সিস্টেম। এই কন্ট্রোলার কার্ডদের লাগানো থাকে মাদারবোর্ডে। কিবোর্ড জাতীয় ডিভাইসের আলাদা করে কন্ট্রোলার কার্ড লাগেনা, কারণ সেই কার্ডটা থাকে কিবোর্ডের ভিতরেই। এগুলোর বাইরে আরো বহু রকমের ডিভাইস হয়। যেমন রেইড (RAID — **R**andom-**A**rray-of-**I**nexpensive-**D**isks) — পরপর অনেকগুলো হার্ডডিস্কে একাধিক কপিতে লেখা থাকে ফাইল ফোল্ডার ফাইলসিস্টেম। এটা করা হয় তথ্যের বাড়তি নিরাপত্তার জন্যে, একটা সেট নষ্ট হয়ে গেলেও তথ্যটা যাতে হারিয়ে না যায়। এই রেইড অ্যাক্রোনিমটায় অন্য শব্দও দেখেছি। ইনএক্সপেনসিভ-এর জায়গায় ইনফর্মেশন, র‍্যান্ডম-এর জায়গায় রিডানড্যান্ট ইত্যাদি। হয় স্কাসি (SCSI — **S**mall-**C**omputer-**S**ystem-**I**nterface) — শুধু হার্ডডিস্ক ড্রাইভ নয়, অনেকসংখ্যক অনেকরকম ড্রাইভ একসাথে ব্যবহার করার খুব উন্নত একটা প্রযুক্তি। ইত্যাদি ইত্যাদি। এর মধ্যে কিছু ডিভাইসের খুঁটিনাটি নিয়ে আমরা পরেও আলোচনা করব। বিশেষত সাত নম্বর দিনে ডিভাইস ফাইলের আলোচনায়।

আগেই বলেছি, কম্পিউটারের ভিতরে এই প্রতিটা অঙ্গপ্রত্যঙ্গই একে অন্যের সাথে তথ্যপথ বা বাস দিয়ে যুক্ত। ভৌত আকারের দিক দিয়ে বলতে গেলে, ধরুন ওই সার্কিটবোর্ডটা জুড়েই বাসের বসবাস, যে বোর্ডে আমরা গুঁজে দিয়েছি কন্ট্রোলার কার্ডগুলো। ভিডিও কার্ড — যা আমার স্ক্রিনে লেখা বা ছবি ফুটে ওঠার প্রক্রিয়াটা চালায়। সাউন্ড কার্ড — যা আওয়াজের শব্দসংকেত বা অডিও সিগনাল এবং বৈদ্যুতিক সিগনাল এই দুটোকে একটা থেকে অন্যটায় বদলায়। ডিস্ক কন্ট্রোলার মানে যা ডিস্কগুলোকে নিয়ন্ত্রণ করে। বাসই সংযুক্ত রাখে একহাতে প্রসেসর চিপ আর মেমরি, এবং অন্যহাতে ডিস্ক-ড্রাইভ, ডিভাইস, কার্ড — এই সবকিছুকে। চিপ, স্ক্রিন, ডিস্ক — সবকিছুর সঙ্গে অন্য সবকিছুর তথ্যপথটাই হল বাস। এই বাস কী ভাবে যুক্ত রাখে একদিকে সিপিইউ আর মেমরি আর অন্যদিকে পেরিফেরাল বা ভৌত উপাদানগুলোকে তার কিছু আলোচনা আমরা ইতিমধ্যেই করেছি, এক আর দুই নম্বর দিনে। মেশিনের সমস্ত ভৌত উপাদানের ভিতর মাইক্রোপ্রসেসরই সেই কেন্দ্রীয় সত্তা যা অন্য সমস্ত কিছুকে চালায়। অথচ সে নিজে সরাসরি কোনো একটা অঙ্গপ্রত্যঙ্গকেও ছুঁতে পায়না। সব যোগাযোগই হয় বাস মারফত। বরং এক কোর মেমরির বা র‍্যামের সঙ্গেই প্রায়-সরাসরি, প্রায়-প্রত্যক্ষ একটা যোগ থাকে প্রসেসরের। কারণ প্রোগ্রাম চালাতে গেলে প্রোগ্রামগুলোকে মেমরিতে থাকতেই হবে। মনে পড়ছে ডিএমএ-র কথা? যখন হার্ডডিস্ক থেকে কোনো তথ্য পড়ার দরকার পড়ে, প্রসেসর তখন বাস মারফত একটা অনুরোধ পাঠায় ডিস্কপাঠের। বাস এই অনুরোধটাকে পৌঁছে দেয় হার্ডডিস্ক কন্ট্রোলারের কাছে। পড়ার কাজ শেষ হয়ে গেলে, ডিস্ক কন্ট্রোলার আবার বাস মারফত এই সিগনাল প্রসেসরকে পাঠায় যে অনুরুদ্ধ তথ্য পড়া হয়ে গেছে এবং তথ্যটা এখন মেমরির এত নম্বর খোপে তোলা আছে। এক নম্বর দিনের সেই রেজিস্টার এবং মেমরি লেখার ছকগুলো মনে করুন। এবার সেই তথ্য মেমরি থেকে তুলে নেয় প্রসেসর, আবার সেই বাস মারফত। এবং তথ্য চটকানোর কেন্দ্রীয় কাজে রত হয় সেন্ট্রাল প্রসেসিং ইউনিট বা সিপিইউ। কিবোর্ড বা কনসোলের সঙ্গেও প্রসেসরের যোগাযোগ ঘটে ওই বাস মারফত।

৬।। কম্পিউটারে প্রাণ-প্রতিষ্ঠা — বুটলোডার

যখন কোনো একটাও প্রোগ্রাম চলছে না — স্ক্রিনে কোনো লেখা বা ছবি কিছু নেই — হার্ড ডিস্কের কাজ করার আলোটাও অফ, তখন কম্পিউটারটা আর কিছুই না — অনেকটা কাঁচ প্লাস্টিক ধাতু তার স্ক্রু এইসবের একটা দামী মণ্ড। তাতে প্রাণ-প্রতিষ্ঠা হয়, কম্পিউটারটা কম্পিউটার হয়ে ওঠে যখন সেটা বুট করে। তার প্রথম কাজই তখন একটা বিশেষ প্রোগ্রামকে চালানো — যাকে নিয়ে আমাদের এই আলোচনাটা শুরু হয়েছিল — সেই প্রোগ্রামটার নাম অপারেটিং সিস্টেম। চালু হয়ে যাওয়ার পরে, অপারেটিং সিস্টেমের উপর এবার ভার পড়ে হার্ডওয়ারের সব বিতিকিচ্ছিরি ব্যাপারকে মাথায় রাখা সামলে রাখা নিয়ন্ত্রণ করা। যাতে অন্য সব প্রোগ্রাম হার্ডওয়ার ব্যবহার করতে

পারে। অপারেটিং সিস্টেমকে জাগিয়ে তোলার এই প্রক্রিয়াটারই অন্য নাম বুটিং। 'বুট' (boot) কথাটা এসেছে বুট মানে জুতো থেকে। বুট-স্ট্র্যাপিং-কে ছোট করে বুট। মানে, কাজটা কি শক্ত ভাবুন, নিজের জুতোর ফিতের ধরে নিজেকে টেনে তোলে, জ্যান্ত কাজের ভিতর নিয়ে আসে অপারেটিং সিস্টেম। আর শেষবার কম্পিউটার যখন অফ করে দিয়েছিলাম, তখন কোর বা র‍্যাম থেকে বাস থেকে সব তথ্য আর আদেশই বেরিয়ে ফুরিয়ে ভ্যানিশ করে গেছিল। শুধু রমে স্থায়ীভাবে লিখে রাখা সামান্য পরিমাণ অত্যাবশ্যক তথ্যটুকু ছাড়া মেশিনে আর কিছুই ছিলনা। খেয়াল রাখবেন, অফ-করা অবস্থায় একটা কম্পিউটারে কিন্তু কোনো অপারেটিং সিস্টেমই নেই। আপনার মেশিনে গ্নু-লিনাক্স ওএস ইনস্টল করলেন মানে, প্রয়োজনীয় ফাইল এবং কনফিগারেশন মেশিনটায় বানিয়ে তুললেন, এখন থেকে বিদ্যুৎ পাঠালেই সেখানে একটা গ্নু-লিনাক্স ওএস গজিয়ে উঠবে। সুইচ টিপে মেশিনে বিদ্যুৎ অন করার পর, সেই ওসটাই উঠতে থাকে, মানে মেশিন বা সিস্টেম বুট করতে থাকে। এই ওএস-নেই থেকে ওএস-আছে হওয়ার গোটা কার্যধারাটা বেশ জটিল। অনেকগুলো ধাপ আছে সেখানে, অনেক অনেক ধাপে অনেক অনেক আদেশ মেনে এগিয়ে তবে বুটপদ্ধতিটা সমাপ্ত হয়। এই কার্যধারার প্রাথমিকতম হদিশটা বুটমুহূর্তে মেশিনকে শেখায় বায়োস (BIOS — **B**asic-**I**nput-**O**utput-**S**ystem) — মানে অত্যন্ত প্রারম্ভিক কিছু নির্দেশ যা ভরা থাকে রম চিপের শরীরেই, যা মুছে যায়না, গেলে আর বুট করাই যেতনা। এই বায়োসে লিখে রাখা প্রাথমিকতম সংক্ষিপ্ত আদেশগুলো পালন করার মধ্যে দিয়েই বুটপ্রক্রিয়া শুরু হয়।

বায়োস সেই বুটমান ক্রমগজায়মান অপারেটিং সিস্টেমকে বলে দেয়, বাছা তুমি অমুক হার্ডডিস্কের অমুক জায়গা থেকে তোমার বুটপ্রক্রিয়ার পরবর্তী আদেশমালা পড়তে শুরু করো। সচরাচর ওই জায়গাটা হল মেশিনের প্রথম হার্ডডিস্কের প্রথম মানে সবচেয়ে কম-সংখ্যক সেক্টর। এই সেক্টরে পৌঁছনোর পর ওএস পাবে তার পরবর্তী ধাপের মন্ত্র। সিদ্ধি আসতে এখনো ঢের দেরি। এই মন্ত্রের নাম বুট-লোডার। বুট লোডার নানা রকম হয়। গ্নু-লিনাক্সে বুট লোডার মূলত গ্রাব (GRUB — **G**rand-**U**nified-**B**oot-**L**oader) আর লিলো (LILO — **Li**nux-**L**oader)। এছাড়া লোডলিন দিয়ে ডস বা উইন্ডোজ মারফতও বুট করানো যায়। আমরা এখানে আমাদের আশু আলোচনাতে লিলোকেই আনব, তার সোচ্চার কারণটা এই যে লিলোটা বোধহয় একটু কম জটিল। আর জনান্তিকে বলার কারণটা এই যে, আমি গ্রাব জানিনা। দু-একবার ঠেকায় পড়ে ব্যবহার করতে হয়েছে, সেটা একদম ধরো-পড়ো-করে-যাও নীতিতে, কিছুই না-বুঝে। গ্নু-লিনাক্সে প্রথম দিকে এটাই বোধহয় সবচেয়ে জনপ্রিয় এবং সফলতম আত্মশিক্ষানীতি।

এবার, প্রাণ-প্রতিষ্ঠার দ্বিতীয় ধাপ মানে লিলো সিনে এসে গেল মানেই, মেশিনের এবার কাজ হল ওই লিলোর আভ্যন্তরীন আদেশমালা পড়ে ফেলা এবং লিলোর গোটা তথ্যটাকে মেমরিতে তুলে ফেলা। এখন থেকে বুটপ্রক্রিয়ার গোটা দায়িত্বটাই লিলো নামের ওই বুট-লোডারের ঘাড়ে। সে এবার একটু একটু করে, ধাপে ধাপে, অপারেটিং সিস্টেমের অবশিষ্ট অংশটাকে ক্রিয়াশীল করতে শুরু করবে, জাগিয়ে তুলতে থাকবে। ক্রমে, তার কাজ সমাপিত হলে, বুটপ্রক্রিয়া শেষ হবে। বুটপ্রক্রিয়া ছাড়িয়ে সত্যিকারের জ্যান্ত একটা অপারেটিং সিস্টেম কাজ শুরু করবে কম্পিউটারে।

এই কাজটা বুটলোডার করবে স্টেপ-বাই-স্টেপ। প্রথমে সে একটা কারনেল খুঁজে বার করবে এবং সেটাকে তুলে নেবে মেমরিতে। পরে যখন আমরা লাইন ধরে ধরে বুটলোডার বুঝব, তখন দেখব, বুটলোডারের মূল কাজ এটাই, সঠিক উপযোগী কারনেলটা কোথায় আছে সেটা জানিয়ে দেওয়া। সেই কারনেল এবার চালু হবে, এবং নিজের কাজ শুরু করে দেবে। এইখানে এসে বুটলোডারের কাজ শেষ। বুটলোডারে দেওয়া থাকতে পারে একাধিক সম্ভাব্য কারনেলের হদিশ। ধরুন একটা মেশিনে একই সঙ্গে উইন্ডোজ আছে এবং গ্নু-লিনাক্স আছে দুটো, সুজে এবং স্ল্যাকওয়ার। মেশিন অন হওয়ার পর প্রথম ফুটে উঠবে লিলোর স্ক্রিন। লিলো জানতে চাইবে, আমরা কোন ওএস-এ বুট করতে চাই? উইন্ডোজ না সুজে না স্ল্যাকওয়ার। যদি আমরা উইন্ডোজ বলি, লিলো নিয়ন্ত্রনটা সরাসরি চলে যেতে দেবে উইন্ডোজ লোডারের হাতে, যা প্রথম হার্ডডিস্কের প্রথম পার্টিশনের গোড়াতেই লেখা থাকে — উইনডোজে এরকমই প্রথা। আর গ্নু-লিনাক্স যে দুটো অপারেটিং সিস্টেম আছে, সুজে বা স্ল্যাকওয়ার, তার দুজনেরই কার্নেল কোথায় আছে, এবং কোন পার্টিশনকে কোন সিস্টেমের মূল বলে ধরতে হবে — সেটাও লিপিবদ্ধ আছে লিলোর কাছে। ধরুন আমরা সুজেতে যেতে চাই, সেক্ষেত্রে লিলো সুজের কার্নেলটা যেখানে রাখা আছে সেটা পড়বে এবং চলে যাবে সুজে সিস্টেমের মূল বা রুট বলে যে পার্টিশন দেখানো আছে, সেইখানে। সেখানেই কাজ শুরু করবে।

লিলো কাজ করে লিলোর কনফিগারেশন বা খুঁটিনাটি-লেখা ফাইল দিয়ে। যার নাম লিলো-কনফ। এই কনফিগারেশন ফাইল, তাদের হদিশ, তাদের গঠন — এসব নিয়ে আমাদের পরে খুবই বিশদ করে জানতে হবে। এখন একটা আলগা আন্দাজ বানানোর চেষ্টা হিশেবে এই লিলো-কনফ (lilo.conf) ফাইলটার গঠনটা দেখুন। মনে রাখবেন, যেহেতু প্রায় কিছু না-জেনেই আপনাকে ফাইলটা পড়তে হচ্ছে, কিছুটা গুলিয়ে যাবেই। কিছু করার নেই। জাস্ট একটু আন্দাজ রেখে যেতে চাইছি। না-বোঝাটা নিয়ে মাথা ঘামাবেন না। পড়ে যান। পরিচিতিটুকুই আপাতত দরকার আমাদের।

'lilo.conf' ফাইলটায় চারটে অংশ আছে। প্রথমে একটা জেনেরাল বা সাধারণ অংশ। একদম উপরের প্রথম দশ লাইন। তারপর এক এক লাইন করে সাদা জায়গা ছেড়ে দিয়ে তিনটে অংশ — সুজে (SuSE), স্ল্যাকওয়ার (Slackware) আর উইনডোজ (Windows)। প্রথম দুটো অংশ সুজে আর স্ল্যাকওয়ার এদের দুজনেরই কারনেল সংক্রান্ত খুঁটিনাটি সিস্টেম কোথা থেকে পাবে তার হদিশ। গ্নু-লিনাক্স সিস্টেমে হার্ডডিস্ক নামকরণের প্রথা অনুযায়ী এই মেশিনের দুটো হার্ডডিস্কের নাম হল 'hda' আর 'hdb', হার্ডডিস্ক এ আর বি, ধরুন। আর গ্নু-লিনাক্সে, আমরা আগেই বলেছি, সব কিছুই এক একটা ফাইল। হার্ডডিস্ক মানে ডিভাইস ফাইল, তারা আছে /dev ডিরেক্টরিতে। এই সিস্টেমে দুটো হার্ডডিস্ক মানে দুখানা ফাইল। তাদের ঠিকানা '/dev/hda' আর '/dev/hdb'। প্রত্যেকটা হার্ডডিস্কে আবার একাধিক করে পার্টিশন আছে। যাদের নম্বর ১, ২, ৩, ৪ করে। এই খ্যাঁচগুলো পরে শিখব আমরা। আপাতত এটুকু বুঝুন, '/dev/hda1' মানে প্রথম হার্ডডিস্কের প্রথম পার্টিশন, '/dev/hda6' মানে প্রথম হার্ডডিস্কের ছয় নম্বর পার্টিশন, '/dev/hdb3' মানে দ্বিতীয় হার্ডডিস্কের তিন নম্বর পার্টিশন।

লিলো-র কনফিগারেশন ফাইল lilo.conf এর একটা উদাহরণ। নিজের এবং নিজের মনের স্বাস্থ্যের কথা ভেবে এর গোটাটা এখনি বোঝার চেষ্টা করবেন না, প্লিজ। এখানে তিনটে আলাদা অপারেটিং সিস্টেম দিয়ে এই মেশিনে বুট করার ব্যবস্থা করা আছে — সুজে, স্ল্যাকওয়ার এবং উইনডোজ। মোটামুটি গড়নটার একটা আন্দাজ পাওয়ার চেষ্টা করুন।

```
boot = /dev/hda
change-rules
reset
default = SuSE
lba32
menu-scheme = Wg:kw:Wg:Wg
message = /boot/message
prompt
read-only
timeout = 150

image = /boot/vmlinuz
    label = SuSE
    append = "hdc=ide-scsi splash=0"
    initrd = /boot/initrd
    root = /dev/hdb3
    vga = 789

image = /mnt/slackware/boot/vmlinuz
    label = SlackWare
    append = "hdc=ide-scsi"
    root = /dev/hda6
    vga = 789


other = /dev/hda1
    label = windows
```

তিনটে আলাদা ওএস-এর তিনটে ব্লক আছে পরে, তার আগে সেই অংশটা যেটা প্রত্যেকের বেলাতেই সত্যি। পরে এই লাইনগুলোকে ভালো করে বোঝার চেষ্টা করব আমরা।

'lilo.conf' ফাইলটার প্রথম লাইনটা দেখুন, 'boot = /dev/hda', এর মানে বুটসেক্টরটা রয়েছে প্রথম হার্ডডিস্কের গোড়াতেই। তারপরের আটটা লাইন আপাতত ছেড়ে দিন। দশ নম্বর লাইন, 'timeout = 150', এর মানে সেকেন্ডের এক দশমাংশের এককে ১৫০ একক, অর্থাৎ পনেরো সেকেন্ড সময় দেবে লিলো। এই পনেরো সেকেন্ড দাঁড়িয়ে

থাকার পরেও যদি আপনি কোনো একটা বিশেষ ওএস-কে নির্বাচন করে না-দেন, তাহলে ও নিজেই বুট করে যাবে ডিফল্ট ওএস-এ। এখানে ডিফল্টটা হল সুজে। দেখুন, চার নম্বর লাইনে আছে, 'default = SuSE'। আপনি পছন্দ করে তিনটের যে কোনোটাই দিতে পারেন। যেটাকেই আপনি পছন্দ করুন, সেই ওএস-এর কারনেলের ঠিকানা দেওয়া আছে তার নিজের অংশে। তিনটে ব্লক আছে দেখুন, সুজে স্ল্যাকওয়ার আর উইনডোজের। ছয়, পাঁচ আর দুই লাইন করে।

যেমন সুজে অংশের প্রথম লাইন দেখুন, 'image = /boot/vmlinuz'। এর মানে, ওই 'vmlinuz' নামের ইমেজ বা সুজের কারনেল ফাইলটা রাখা আছে '/boot' নামের ডিরেক্টরিতে। এই ডিরেক্টরিকে কোথায় খোঁজা হবে? সুজে সিস্টেমের শিকড়ে, রুট ডিরেক্টরিতে। তার ঠিকানা কী? সুজে ব্লকের পাঁচ নম্বর লাইনে দেখুন, 'root = /dev/hdb3'। জ্যামিতিতে যেরকম আমরা অরিজিন শিফট করে অন্য জায়গায় নিয়ে যাই এখানেও তেমনি একটা ঘটনা ঘটছে। বুট করার সময়, বুট লোডার পড়ার সময়, প্রথম জেনেরাল অংশে দেখুন, শিকড় দেখাচ্ছিল প্রথম হার্ডডিস্ককে। আগেই বললাম না, প্রথম হার্ডডিস্কের গোড়াতেই লেখা থাকে বুটলোডার, পরে এর ভৌত ব্যাপারটা বুঝব আমরা, হার্ডডিস্কের গঠনের আলোচনায়। আর সুজের কারনেলের প্রসঙ্গে দেখুন, শিকড়টাকে সরিয়ে নিয়ে যাওয়া হচ্ছে দ্বিতীয় হার্ডডিস্কের তিন নম্বর পার্টিশনে। '/dev/hdb3' বা দু নম্বর হার্ডডিস্কের তিন নম্বর পার্টিশনটাই শিকড় বলে ভাবা হবে, সেখানে গিয়ে সিস্টেম তার প্রয়োজনীয় ফাইলগুলো খুঁজবে। একই রকম একটা সরিয়ে নিয়ে যাওয়া ঘটছে স্ল্যাকওয়ারের বেলাতেও। শিকড়টা বদলে যাচ্ছে '/dev/hda6' বা প্রথম হার্ডডিস্কের ছয় নম্বর পার্টিশনে, কারণ সেটাই স্ল্যাকওয়ারের রুট। রুট পার্টিশন কী বস্তু সেটা আমরা ভালো করে জানব সাত নম্বর দিনে গিয়ে।

সুজে অংশে এর পরের লাইন দেখুন, 'initrd = /boot/initrd'। 'initrd' শব্দটা হল 'initialized-ramdisk' শব্দবন্ধের সংক্ষিপ্ত আকার। আট নম্বর দিনে গিয়ে আর একটু ভালো করে বুঝব আমরা। র‍্যামডিস্ক বলতে র‍্যামের একটা অংশকে আলাদা একটা ডিস্ক আকারে ধরে নিচ্ছে মেশিন, এবং সেই র‍্যামডিস্কটা ইনিশিয়ালাইজ করার কথা বলা হচ্ছে, মানে বুটলোডার যাতে এই র‍্যামডিস্কে কারনেলের চালু হওয়ার জন্যে প্রয়োজনীয় নানা ডিভাইস ড্রাইভার মডিউল তুলে নেয়, যা সিস্টেম চালু হওয়ার জন্যে দরকার পড়বে। কারনেল এই র‍্যামডিস্ক থেকে তার প্রয়োজনীয় ড্রাইভারগুলোকে পড়ে নেবে। এই লাইনটা কারনেলকে বলে দেয় কোথায় সে এই র‍্যামডিস্কটাকে পাবে তার ঠিকানা, মানে '/boot/initrd' নামের ফাইল। দেখুন, এই লাইনটা স্ল্যাকওয়ার অংশে নেই, কারণ স্ল্যাকওয়ার কারনেল অন্য ভাবে কাজ করে। সুজে অংশে এর পরের লাইন হল 'vga = 789'। এই লাইনটা বলে দেয়, কী রকম স্ক্রিনে কী সাইজের অক্ষর দিয়ে সে কমান্ড প্রম্পট ফুটিয়ে তুলবে তার হদিশ। এর নানা মান হতে পারে। যেমন, ৭৭১ মানে হল ৮০০×৬০০ পিক্সেল আকারের স্ক্রিনে আট বিটে ২৫৬ রঙের ডিসপ্লে। এখানে যে মানটা আছে, সেই ৭৮৯ মানে হল ওই একই সাইজের স্ক্রিনে ২৪ বিটে ১৬ মিলিয়ন রঙের ডিসপ্লে। মানটা ৭৯১ হলে, তার মানে হত ১০২৪×৭৬৮ পিক্সেল আকারের স্ক্রিনে ১৬ বিটে ৬৫ হাজার রঙের ডিসপ্লে। এইরকম। এর কোনটা করা যাবে, সেটা নির্ভর করে হার্ডওয়ারের উপর, র‍্যামের উপর, ভিডিও র‍্যামের উপর। এই জায়গাটায় বেশ ইন্টারেস্টিং অনেককিছু আছে, কিন্তু সেগুলো আমাদের এই পাঠমালায় আসবে না, ওগুলো ভালো করে বুঝতে হত এক্স-উইনডোজ বা গুই জানতে চাইলে। সেটা তো আমরা প্রথমেই আমাদের আলোচনা থেকে বাদ দিয়ে দিয়েছি।

'lilo.conf' ফাইলের সুজে বা স্ল্যাকওয়ার ব্লক দুটোয় দেখুন, খুব সামান্যই তফাত। শুধু ওই র‍্যামডিস্কের লাইনটা বাড়তি আছে সুজেতে। এছাড়া কাঠামোটা দুটোরই এক। উইনডোজ অংশে দেখুন এসব খুঁটিনাটি কিছুই নেই, কারণ উইনডোজ ওএস-এর হালহকিকত এবং বুট করার তরিকা দেওয়া আছে প্রথম হার্ডডিস্কের প্রথম সেক্টরেই। সেখান থেকে, সেই অনুযায়ী, পরপর সিস্টেম পড়ে নেবে, এবং লোড করবে। তার ব্যাকরণ গ্নু-লিনাক্স থেকে একদম আলাদা। এই যে লিলো কনফিগারেশন ফাইলের কিছু লাইনকে আমরা একদম ধরলামই না, এদের কিছু আলোচনা পরে আসবে।

তাহলে দেখুন, একাধিক সিস্টেমের একাধিক কারনেল একই সঙ্গে নানা হার্ডডিস্কের নানা পার্টিশনের নানা জায়গায় আগে থেকেই ভরা আছে। শুধু যে একাধিক সিস্টেমের একাধিক কারনেল তা নয় কিন্তু। একই সিস্টেম একাধিক কারনেল দিয়ে লোড করা যেতে পারে। ধরুন সুজে সিস্টেমে এমনিতে দেওয়া কারনেলে আমি আমার সিস্টেম বুট করি। এবার আমার প্রয়োজন হল, অন্য কিছু মডিউল কারনেলে ঢোকানোর। ধরুন আমি আরো দ্রুতগতি করার জন্যে

স্কাসি ব্যবস্থা করলাম মেশিনে। স্কাসি হার্ডডিস্ক লাগালাম আইডিই হার্ডডিস্কের বদলে। এবার আমার চালু কারনেলটায় যদি স্কাসি ড্রাইভার এমনিতে না-থাকে, তাহলে সেই ড্রাইভারটা সঙ্গে দিয়ে আমি একটা নতুন কারনেল দিয়ে আমার সিস্টেম লোড করতে পারি। বা, নতুন কারনেল সংস্করণ বেরোলো, সেটাকে কম্পাইল করে নেওয়া যেতে পারে নিজের সিস্টেমে। জিসিসি কম্পাইলার তো দেওয়াই থাকে গ্নু-লিনাক্স বাক্সে। যেমন এখন নতুন কারনেল হল ২.৬। আমাদের কোলকাতা লাগের সায়মিন্দু সহ অনেকেই বেশ মাসতিনেক হয়ে গেল ২.৬ কারনেল নামিয়ে নিজের মেশিনে কম্পাইল করে নিয়েছে। আমারটায় পুরোনো কারনেলই চলছে। এক, আমার অত আপটুডেট ড্রাইভার প্রয়োজনও পড়েনা, আর ওদের মত করে এইসব হার্ডওয়ার জটিলতা আমি আদৌ বুঝিনা। দু-একবার কারনেল কম্পাইল করেছি, কিন্তু সে বেশ লাগে-তুক-না-লাগে-তাক মেথডে। যাই হোক, যা বলছিলাম, একই সিস্টেমেরও একাধিক কারনেল থাকতে পারে। যে কোনো একবার শুধু একটা কারনেল দিয়েই বুট করা যাবে, যে কারনেলটা আপনি পছন্দ করে দেবেন লিলোকে।

বিদ্যুৎ অন করার পর, প্রাথমিক কিছু হার্ডওয়ার সংক্রান্ত লাইন স্ক্রিন জুড়ে চলে যাওয়ার পর, প্রথম যেটা ফুটে উঠবে সেটা লিলো স্ক্রিন, যেখানে লিলো তার সম্ভাব্য কারনেল কাম ওএস-গুলোর তালিকা ফুটিয়ে তুলবে। সেখান থেকে একটা কোনো কারনেলকে আপনি পছন্দ করে দেবেন। 'lilo.conf' ফাইল মোতাবেক লিলো ইনস্টল করে নিতে হয়। ইনস্টল করতে হয় '`lilo`' কমান্ড দিয়ে, '/etc' ডিরেক্টরিতে 'lilo.conf' ফাইলটা লিখে নেওয়ার পর। লিলো কী ভাবে ইনস্টল করা হবে, কী কী কারনেল রাখা হবে তার তালিকায়, সেইসব কনফিগারেশনটাই লেখা থাকে 'lilo.conf' ফাইলে — পরে এগুলো আরো বুঝব আমরা। লিলো ইনস্টল করার পর তৈরি হয় বুটকালীন কারনেল তালিকা ফুটে ওঠার এই স্ক্রিন। এই তালিকায় পরপর তিনটে ওএস-এর নাম ফুটে উঠবে, আমরা আমাদের একটু আগের উদাহরণ হিশেবে দেওয়া 'lilo.conf' ফাইলে সেই রকম কনফিগারেশনই দিয়েছি লিলোকে। এই তিনটে পরপর নাম হল, 'SuSE', 'Slackware', এবং 'windows'। দেখুন, 'lilo.conf' ফাইলের তিনটে ব্লকে তিনটে আলাদা নাম দেওয়া আছে, 'label = ' লাইনে। মানে তিনটে ওএস-এর তিনটে কারনেলের হদিশ। যদি সুজের দুটো কারনেল থাকত তাহলে আবার এখানে পছন্দের সংখ্যা চারটে। এবার, এই লিলো স্ক্রিনে দেওয়া তালিকায় উর্দ্ধমুখী তীরচিহ্ন (↑) আর নিম্নমুখী তীরচিহ্ন (↓) দিয়ে উঠে নেমে কোনো একটা নির্দিষ্ট অপারেটিং সিস্টেমকে, মানে সেই সিস্টেমের কারনেলকে পছন্দ করে আমরা যেই এন্টার মারি, সেই কারনেলটা লোড হতে শুরু করে। মানে তার ঠিকানা থেকে তাকে মেমরিতে তুলে নিতে শুরু করে মেশিন। অল্প সময়ের জন্যে কনসোলের কালো পর্দায় পরপর কতকগুলো সাদা বিন্দু ফুটে উঠতে থাকে — মানে, লিলো তখন কারনেল লোড করছে। এখনকার বিদ্যুৎগতি সব প্রসেসরের কল্যাণে এই ডিসপ্লেটা প্রায় চোখের পলক ফেলতে না-ফেলতেই মিলিয়ে যায়।

যাইহোক, কারনেল মেমরিতে উঠে গেল, জ্যান্ত হয়ে উঠল, তার নিজের কাজ শুরু করে দিল — এইখানেই লিলোর কাজ শেষ। এখানে মনে হতে পারে, শুধু শুধু লিলোকে আনা কেন? কারনেল পছন্দ করা, পছন্দ করা কারনেলকে তার ঠিকানা থেকে পড়তে শুরু করা, তারপর সবার শেষে কারনেলটাকে জ্যান্ত করে তোলা — এই অব্দি দায়িত্বটা তো বায়োসের কাছে থাকলেই পারত? তাহলে লিলো বলে আলাদা কোনো একটা প্রোগ্রামকে চালু করার দরকার পড়ত না। এতে সমস্যা এই যে, বায়োসে কতটা স্মৃতি রাখা যাবে তার পরিমাণটা খুব সীমিত। সেই অর্থে কম্পিউটারের বায়োস জিনিষটা বেশ আকাঠ, বায়োসের কোডগুলো প্রথম লেখা হয়েছিল যখন, তখন কম্পিউটার কাজ করত ৮ বিটে, মানে ৮ বিট ছিল তার স্মৃতি নাড়াচাড়ার একক। আর তথ্য লেখাপড়ার কাজগুলো হত এখনকার তুলনায় হাফটিকিট ওয়ানফোর্থ টিকিট কি আরো কুচো কুচো হার্ডডিস্কে। এখন এই ৩২ পেরিয়ে ৬৪ বিটের দিকে ধাবমান মেশিনের যুগেও বায়োস হুবহু এক না-থাকলেও আমূল বদলে গেছে এমনটাও বলা যায়না। তাই এই বায়োসের পক্ষে আজকের দৈত্যাকার হার্ডডিস্কের এত কম অংশই মাথায় রাখা সম্ভব যে তার দ্বারা সরাসরি কারনেল লোডিং সম্ভব নয়। বায়োস আপগ্রেড করে, যার অন্য নাম বায়োস ফ্ল্যাশিং, মানে বায়োসের পুরোনো তথ্য মুছে নতুন তথ্য লিখে বায়োসের কিছু জায়গা বদলানো যায় ঠিকই, কিন্তু, প্রথমত, বায়োস ফ্ল্যাশিং বেশ ঝঞ্ঝাটের কাজ, আর দ্বিতীয়ত, ফ্ল্যাশ করে তো আর আমরা বায়োসে রাখা তথ্যের পরিমাণ বাড়াতে পারছি না। আর, বুট লোডারের সুবিধাটা এই যে, তথ্যটা যেহেতু লেখে হার্ডডিস্কে, যতখুশি জায়গা, তাই, একসঙ্গে মেশিনে যতখুশি অপারেটিং সিস্টেম রেখে তার যে কোনোটাতে ঢুকে যেতে পারার সুযোগ।

৭।। কারনেল নিজের জমি বুঝে নিচ্ছে — ইনিট

কারনেল কাজ করতে শুরু করার পরই প্রথম যাতে হাত দেয় তা হল হার্ডওয়ারটাকে বুঝে নেওয়া মিলিয়ে নেওয়া। যাতে এরপরেই প্রোগ্রাম চালানোয় হাত দেওয়া যায়। এই হার্ডওয়ার খোঁজার সময় কারনেল যে গোটা মেশিনের সমস্ত কিছু খুঁজে দেখতে দেখতে যাচ্ছে তা কিন্তু নয়। কতকগুলো বিশেষ জায়গা আছে, যেমন ইনপুট-আউটপুট সংযোগবিন্দু বা আইও-পোর্টগুলো — বিশেষ কিছু বাস-রুটের ঠিকানা যেখানে কন্ট্রোলার কার্ডগুলো খাপ পেতে বসে থাকে — কখন তাদের আদেশ পাঠানো হবে কাজ করার — ইত্যাদি। মানে, কারনেল যে এলোমেলোভাবে হাতড়ে চলে তা নয়, কারনেলের মধ্যেই পর্যাপ্ত পরিমাণে এই তথ্য ভরা আছে যে কোথায় হাতড়ালে কী তার খুঁজে পাওয়ার কথা, আর ঠিক কী আকারে খোঁজ করতে চাইলে তবেই তার এই সন্ধানে সাড়া দেবে কার্ডগুলো। ফিজায় ঋত্বিকের মত বারবার নতুন করে খুঁজতে হবে, নাকি, রোড-এর মনোজ বাজপেয়ীর মত খোঁজা শেষ করে নিশ্চিন্ত হওয়া মাত্রই হঠাৎ করে উদয় হবে — এসব কারনেল আগে থেকেই জানে, তাকে চমকানো খুব শক্ত।

এই বিশেষ রকমে শিক্ষিত খোঁজাখুঁজিরই টেকনিকাল নাম অটোপ্রোবিং। স্বসন্ধান বলা যায়। গ্নু-লিনাক্স একটা ব্যবস্থায়, বুট করা শুরু করার পরেই, কনসোলের পর্দায় যে রাশি রাশি লাইন সাঁই সাঁই করে নিচ থেকে উপরে চলে যায় — সেসব কারনেলের এই অটোপ্রোবিং-এর ফলাফল। আইও পোর্টগুলো বাজিয়ে যাচাই করে যে বুঝে নিচ্ছে — কী কী রসদ এইমুহূর্তে সে পেতে পারে মেশিনের মধ্যে — পরে প্রোগ্রাম চালাতে গিয়ে কোনো ঘাপলা যাতে না-ঘটে। আর এক একটা করে উপাদান খুঁজে পাচ্ছে এবং নিজেকে সেইরকম করে প্রস্তুত করে নিচ্ছে কারনেল, মেশিনের সাথে নিজেকে একদম খাপে খাপে মিলিয়ে নিচ্ছে। এই কাজে গ্নু-লিনাক্স কারনেল বেজায় দড়। অন্য সমস্ত ধরনের ইউনিক্সের চেয়ে অনেকটা দক্ষ, আর ডস বা উইনডোজের কথা তো ছেড়েই দিন। অনেক পুরোনো লিনাক্সীই মনে করে যে, নানা ধরনের আবদ্ধ এবং উন্মুক্ত ইউনিক্সের মধ্যে থেকে গ্নু-লিনাক্স যে নিজের আলাদা জায়গা বানাতে শুরু করেছিল তার একটা বড় কারণ এত উচ্চস্তরের বুটকালীন স্বসন্ধান। এই উমদা অটোপ্রোবিং-এর কারণে গ্নু-লিনাক্স ইনস্টল করার ঝামেলা অনেকটা কমে যায়। নিজেই কারনেল খুঁজে নেয় ভৌত উপাদানগুলো এবং তাদের ড্রাইভার নিজেই লোড করে নেয়, কিছুই প্রায় তাকে আলাদা করে বলতে হয়না।

কিন্তু গোটা কারনেলটা মেমরিতে তুলে নেওয়া এবং চালানো মাত্রই কিন্তু বুটপ্রক্রিয়া শেষ নয়। অ্যাকচুয়ালি এটা প্রথম পদক্ষেপটা সমাপ্ত হল মাত্র। এই প্রথম স্টেপের শেষে কারনেল যে দায়িত্ব, মানে বুটপ্রক্রিয়ার নেতৃত্ব হাতে পেয়েছিল, সেই পতাকা এবার তুলে দেবে ইনিট-এর হাতে, যে ইনিট ফের শুরু করবে নতুন দিনের শুরুতে মেশিনের ঘরগেরস্থালি ঠিকঠাক ভালো করে গুছিয়ে নেওয়ার অনেকগুলো কাজ। ইনিট (init) আর একটা প্রোগ্রাম, ইনিট হল ইনিশিয়ালাইজেশন। একটা সিস্টেম প্রোগ্রাম, তাই রুট ছাড়া কেউ হাত দিতে পারেনা এই প্রোগ্রামে বা তার কনফিগারেশন ফাইলে। ঠিক লিলো প্রোগ্রামের যেমন কনফিগারেশন ফাইল ছিল ‘lilo.conf’, তেমনি ইনিট প্রোগ্রামেরও একটা কনফিগারেশন ফাইল থাকে, তার নাম ‘inittab’। সেটাও ‘lilo.conf’ ফাইলেরই মত থাকে সিস্টেমের ‘/etc’ ডিরেক্টরিতে। পরে আমাদের খুবই ভালো করে জানতে হবে এগুলো। আর ‘/etc’ ডিরেক্টরিটাকেও আলাদা ভাবে জানব আমরা নয় নম্বর দিনের ১.৪ নম্বর সেকশনে গিয়ে। আর আগেই তো বলেছি, এখুনি যদি কৌতূহল মেটাতে চান, যা খুবই ভালো অভ্যেশ, নিজের গ্নু-লিনাক্স সিস্টেমে দাঁড়িয়ে ম্যান (man) কমান্ড ব্যবহার করুন। আমি যেভাবে চেয়েছি তাতে এই পাঠমালাটা সত্যিই অর্থপূর্ণ হয়ে উঠবে যখন নিয়ন্ত্রণটা আমার হাত থেকে আপনি নিজের হাতে নিয়ে নেবেন। প্রথমদিকে এটা হবে খুব ছোট ছোট জার্নি। পরে সেটা বাড়তে থাকবে।

এই ইনিট বা ইনিশিয়ালাইজেশন প্রক্রিয়া হল সিস্টেম বুট করার পর, মানে বুটলোডারের হাত থেকে কারনেলের হাতে নিয়ন্ত্রণটা চলে যাওয়ার পর, কম্পিউটারে চলা প্রথম প্রক্রিয়া বা প্রসেস। ইনিট প্রোগ্রামের প্রসেস আইডি ১। আমরা এক আর দু-নম্বর দিনে প্রসেস নিয়ে আলোচনা করেছি, একটা প্রোগ্রাম যেই চলে সেটা একটা প্রসেস হয়ে ওঠে। প্রত্যেকটা প্রসেসেরই একটা করে আইডি বা আইডেন্টিটি বা নম্বর থাকে এবং থাকে বংশপরিচয়। মানে, উপরদিকে থাকে মাতাপিতা-প্রসেস, যে প্রসেসের থেকে এই প্রসেস উদ্ভূত হয়েছে, আর নিচের দিকে ছানাপোনা-প্রসেস, মানে এই প্রসেস থেকে যে প্রসেসরা উদ্ভূত হয়েছে। একমাত্র এই ইনিট ছাড়া। ইনিট হলেন স্বয়ম্ভু, কোনো মাতাপিতা নেই, যদিও, রাবণের বংশের মত ছানাপোনা আছে। একটা চলমান মেশিনে গোটা গ্নু-লিনাক্স সিস্টেম জুড়ে কাজ করতে থাকা সমস্ত প্রসেস-ই এই ইনিটের বংশধর, তাই এর আইডি ১।

পরে আমরা ইনিট প্রোগ্রাম এবং তার কনফিগারেশন ফাইল 'inittab' এগুলোকে অনেক বিশদে জানব। 'inittab' ফাইলের লাইন ধরে ধরে। আপাতত এটুকু মাথায় রাখুন, ইনিট কাজ করে রানলেভেল (runlevel) বলে একটা জিনিষ দিয়ে, যার মান এক থেকে ছয় অব্দি হতে পারে। 'init 1', 'init 2' থেকে 'init 6' অব্দি যে কোনো কমান্ড দিয়ে সেই সেই রানলেভেলকে ডেকে আনা যায়। '/etc' ডিরেক্টরির 'inittab' ফাইল, মানে ইনিটের কনফিগারেশন ফাইল থেকে কয়েকটা লাইন তুলে দিই। এই লাইনগুলো ফাইলে রাখা আছে ইনিট প্রক্রিয়াটা আমাদের বোঝার স্বার্থেই। যার প্রমাণ লাইনের গোড়ায় ওই '#' চিহ্নটা। এই চিহ্নটা লাইনের গোড়ায় থাকলে সেই লাইনগুলোকে বলে কমেন্ট লাইন। মেশিন এই লাইনগুলো পড়েনা, আমরা পড়ি, যদি পড়তে চাই। তবে মাথায় রাখবেন এটা সুজে সিস্টেমের ইনিটট্যাব ফাইল থেকে তোলা। গ্নু-লিনাক্সেরই একটা ডিস্ট্রিবিউশন থেকে আর একটা ডিস্ট্রিবিউশনের সিস্টেমে এই রানলেভেলগুলো অনেকসময় একটু একটু পৃথক হয়। লাইনগুলোয় অনেক জিনিষই আপাতত বুঝতে পারবেন না। অল্প জেনে বেশি বোঝার প্র্যাকটিস শুরু করে দিন। তাতে মাঝে মাঝে একটু কেলেঙ্কারি ঘটতে পারে, কিন্তু মোটের উপর বোঝাটা বাড়বে। রানলেভেল ০ মানে সিস্টেম বন্ধ করছেন, আর ৬ মানে নতুন করে বুট করছেন। রানলেভেল ১ মানে যেখানে একটাই মাত্র ব্যবহারকারী থাকবে সিস্টেমে, সেই রুট, সেই সাধারণ ইউজার। রানলেভেল ২ মানে একাধিক ইউজার থাকতে পারে, কিন্তু নেটওয়ার্ক নেই। রানলেভেল ৩ মানে মাল্টি-ইউজার এবং নেটওয়ার্ক দুই-ই পুরোপুরি আছে। রানলেভেল ৫ মানে এই রানলেভেল ৩ গোটাটাই, আর তার সঙ্গে এক্সডিএম (xdm — **x-d**isplay-**m**anager), মানে এক্স-উইনডোজ বা গুই ব্যবহার করার সুযোগ। আমি নিজে যেটা পছন্দ করি, সেটা হল রানলেভেল ৩ দিয়ে সিস্টেম বুট করা, তারপর ইচ্ছেমতন এক্স-উইনডোজে ঢোকা বা বেরোনো। আর রানলেভেল ৫ মানে যেখানে প্রথমেই আমি গুই দিয়ে ঢুকছি, এক্স-উইনডোজের বাইরে যেতেই পারছিনা। ছবি আর ইঁদুর — জিনা ইয়াহা মরনা ইয়াহা ইসকা সিওয়ায়ে যানা কাহা। আর, রানলেভেল ৪ সুজেতে ব্যবহারই হয়নি।

```
# runlevel 0  is  System halt   (Do not use this for initdefault!)
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network (e.g. NFS)
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot (Do not use this for initdefault!)
```

এক থেকে ছয় — কোন মানটা আমি ব্যবহার করতে চাই আমার সিস্টেমের ইনিটের রানলেভেল হিশেবে, সেটা দিয়ে দিতে হয় সিস্টেমকে, 'inittab' ফাইল বদলে। এটা সিস্টেম প্রোগ্রামের এলাকা বলে রুট ছাড়া আর কেউ হাত দিতে পারেনা এই ফাইলে। পরে আমরা দেখব, ইনিট প্রোগ্রামের কাজ হল গেটি (getty — **get-tty**) বলে আর একটা প্রোগ্রাম চালু করা। গেটি ব্যবহারকারীদের হয়ে 'tty' বা টেলিটার্মিনাল পেয়ে দেয়। চার নম্বর দিনের বহু-ব্যবহারকারীর সিস্টেম তৈরি হওয়ার আলোচনাটা মনে করুন। অনেকগুলো টেলিটার্মিনাল, তারা প্রত্যেকটাই যুক্ত মূল একটা সিপিইউ-র সঙ্গে। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সেই চিন্তার মূল মডেল এটাই। এবার, কোন টেলিটার্মিনাল গেটি পেয়ে দেবে ব্যবহারকারীর জন্যে, ব্যবহারকারী সেখানে কী কী সুযোগ এবং অধিকার পাবে, সেটাই ঠিক করা হয় রানলেভেল দিয়ে। 'inittab' ফাইলে যে রানলেভেল ঠিক করে দিচ্ছি আমরা, সেই মত সুযোগ ব্যবহারকারীকে দেবে গেটি। সাত নম্বর দিনে অধিকার এবং মালিকানার ব্যাপারটা স্পষ্ট হওয়ার আগে অব্দি এর গোটাটা বুঝতে পারবেন না। আলগা বোঝা দিয়েই কাজ চালাতে থাকুন, ছেড়ে দেবেন না। একসময় দেখবেন গোটা ব্যাপারটাই ধরে ফেলেছেন।

'/etc' ডিরেক্টরিতে 'inittab' ফাইলে ডিফল্ট ইনিট রান-লেভেল ঠিক করে দেওয়ার একটা লাইন থাকে। সেটাকে বদলে আমরা আমাদের সিস্টেমের ডিফল্ট রানলেভেল বদলাই। ইনিটট্যাব-এ লিপিবদ্ধ আলাদা আলাদা রান-লেভেল ২ বা ৩ বা ৫ অনুযায়ী কম্পিউটার ঢোকে মাল্টি-ইউজার মোডে বা নেটওয়ার্ক মোডে বা গুই মোডে। আগেই তো বললাম, এর এমন কোনো নিয়ম নেই যা সব ডিস্ট্রিবিউশনেই একইভাবে প্রযোজ্য। সুজেতে যা তা স্ল্যাকওয়ারে নয়। শুধু ০ আর ৬ রানলেভেল নির্দিষ্ট থাকে কম্পিউটার শাটডাউন বা অফ করে দেওয়ার সঙ্গে। ইনিটট্যাব ফাইলে আমরা আলাদা করে যে কোনো একটা রানলেভেলেই ঢোকার কথা বলে দিতে পারি।

ইনিট এবার চালায় কিছু স্ক্রিপ্ট। স্ক্রিপ্ট কাকে বলে তার একটা হালকা আন্দাজ দিয়েছি আমরা দিয়েছি, সেটা কিন্তু শোকেস বা তারও কম। গুদামে আরো প্রচুর আছে। দশ নম্বর দিন গোটাটাই তো এই স্ক্রিপ্ট নিয়েই। স্ক্রিপ্ট বলতে, মোটামুটি অর্থে, পরপর নিজে নিজে কম্পিউটারের করে যাওয়ার কাজের তালিকা। এই তালিকাটা বানাতে হয় নির্দিষ্ট কিছু নিয়ম মেনে। এর কিছু নিয়ম হল কমান্ড ইন্টারপ্রিটারের, আর কিছু নিয়ম স্ক্রিপ্টিং ভাষাটার। কমান্ড ইন্টারপ্রিটারের কথা এসেছিল এক নম্বর দিনের শেষে — যে আমাদের আদেশ মেশিনের বোঝার মত করে পৌঁছে দেয়। শেল হল কমান্ড ইন্টারপ্রিটার। এই শেলের নিজের কিছু নিয়মকানুন থাকে। আর স্ক্রিপ্ট দিয়ে একটা বিপুল পরিমাণ প্রোগ্রামিং কাজই করে নেওয়া যায়, করা হয়ও। সেই করার স্বার্থে ওই স্ক্রিপ্ট লেখার ভাষার কিছু ব্যাকরণ বানানো থাকে। সেগুলোও মানতে হয় স্ক্রিপ্ট লেখার সময়। স্ক্রিপ্ট লিখে প্রোগ্রামিং কাজ করে কমান্ড ইন্টারপ্রিটার দিয়ে, একে বলে ইন্টারপ্রিটেড ভাষায় কাজ করা। আর সি ইত্যাদি ভাষায় লেখা প্রোগ্রাম বানাতে হয় কম্পাইলার দিয়ে, তাকে বলে কম্পাইলড ভাষায় কাজ করা। এই দুটোর পার্থক্য নিয়ে বড় আলোচনা আছে দশ নম্বর দিনে গিয়ে। এই শেল স্ক্রিপ্টিং দিয়ে কত কত কাজ যে করে নেওয়া হয় বা করে নেওয়া যায় তার কোনো ইয়ত্তা নেই। পরে আমরা আসব এই আলোচনায়।

যা বলছিলাম, ইনিটের প্রাথমিক দায়িত্ব এইরকম কিছু স্ক্রিপ্ট চালানো। ইনিট এই স্ক্রিপ্টগুলোকে চালিয়ে দেখে নেয় ডিস্ক নেটওয়ার্ক কার্ড ডিসপ্লে কার্ড সাউন্ড কার্ড ইত্যাদি ভৌত রসদগুলো রসদগুলো মেশিনের মধ্যে সব ঠিকঠাক আছে কিনা। হার্ডওয়ার ইন্টারাপ্ট, সিস্টেম কল, ইত্যাদি যে ব্যাপারগুলো আমরা উল্লেখ করেছি দুই নম্বর দিনে, সিস্টেম বাসকে ঘিরে সেগুলো সঠিকভাবে কাজ করছে কিনা, এই স্ক্রিপ্টদের দিয়ে সেটাই পরখ করে নেয় ইনিট। বা কনফিগারেশনে যাদের যাদের পাওয়ার কথা তাদের ঠিকঠাক পাচ্ছে কিনা। ডিভাইস ড্রাইভারের যে যে মডিউলগুলো লোড করছে কারনেল তাদের সঙ্গে সিস্টেমের ভৌত ডিভাইসের কনফিগারেশন মিলছে কিনা, ইত্যাদি। এমনকি মডিউলগুলোর সঙ্গে জড়িত কোনো লাইব্রেরি শেষবার বুটের পরে যদি বদলে গিয়ে থাকে সেটাও আপডেট করে নেওয়ার কথা থাকে এই স্ক্রিপ্টে (এই যে কথাগুলো পড়ে যাচ্ছেন এর অনেকটাই আপনার হিব্রু বলে মনে হতে পারে, এমনকি এদের সম্পর্কে পরিচিতিগুলো পাঠমালার গোড়ার দিকে পড়ে আসার পরেও। কারণ এদের এখনো জানেন আপনি, চেনেন না, এখনো এরা আপনার মাথার বাইরে বসবাস করছে। বারবার পেতে পেতে একসময় দেখবেন মাথার ভিতরে ঢুকে গেছে, মিলে যাচ্ছে সব, কিন্তু সেটা ঘটবে এই পাঠমালায় না, আপনার মাথার মধ্যে। সেখানে পাঠমালাটা শেষ, আপনার নিজের যাত্রার শুরু)। শেষবার যদি মেশিন ঠিকভাবে বন্ধ বা শাট-ডাউন না-হয়ে থাকে, তাহলে ডিস্কে লেখা ফাইলব্যবস্থায় কিছু কিছু গোলযোগ চলে আসতে পারে। হঠাৎ কারেন্ট চলে গিয়ে এটা ঘটতে পারে, অন্য কারণেও ঘটে থাকতে পারে। ডিস্কের শরীরে লিপিবদ্ধ ফাইলসিস্টেমে কোনো গোলযোগ আছে কিনা, সেটাও খতিয়ে দেখে নেয় ইনিট তার এই ঘরগেরস্থালি গুছিয়ে নেওয়ার প্রক্রিয়ায়। কখনো কখনো প্রয়োজন অনুযায়ী কোনো শুশ্রূষার কাজ করতে হবে কিনা সেটা যাচাই করে নেওয়াও ইনিটেরই কাজ।

ইনিট এবার জাগিয়ে তোলে কিছু যক্ষ বা যখকে। কুবেরের ধনের মত লিনাক্স সিস্টেমকে অতন্দ্র প্রহরা দিয়ে যাওয়াই কাজ এই ডিমন বা যখগুলোর। কোনো যখ পাহারা দেয় প্রিন্ট স্পুলারকে (স্পুল কাকে বলে মনে করে নিন), এর নাম প্রিন্ট ডিমন (printd)। কোনো যখ পাহারা দেয় মেইলের নড়াচড়াগুলো, কোথা থেকে কোন মেইল কোথায় যাচ্ছে, তার নাম মেইল ডিমন (maild)। কোনো যখ পাহারা দেয় সময়সারনী মোতাবেক করার কাজ গুলোকে, ঠিক সময়ে কাজগুলো শুরু আর শেষ হচ্ছে কিনা, তার নাম ক্রন ডিমন (crond), ইত্যাদি, মনে পড়ছে? এই পাহারার কাজে কোনো যখেরই কখনো কোনো ভুল হয়না, স্বাধিকারপ্রমত্ত হয়না তারা কখনো, তাই তাদের কুবেরের হাতে শাস্তি পেতে হয়না, নির্বাসিত হতে হয়না, কান্তাবিরহগুরুভারে হাহুতাশ করতে হয়না। হায়, তাই কম্পিউটারের যখেদের কোনো মেঘদূত কাব্য লেখা হয়না।

এই ডিমন বা যখ-গুলো অনেকগুলো পরস্পর প্রতিযোগী কাজের বিভিন্ন ধরনের রসদ-চাহিদার উপরও নজর রাখে। কোনো দুটো প্রোগ্রাম একই সঙ্গে একই রসদ দাবি করে কোনো গোলযোগ বাধাচ্ছে কিনা সেটাও নজর রাখতে হয় এই যখদের। এইভাবে যখ বা সদাজাগ্রত ডিমনের আকারে এই ধরনের প্রোগ্রামকে রাখার কারণ এই যে, একসঙ্গে অনেকগুলো প্রোগ্রাম যখন চলছে, তাদের প্রত্যেকটার রসদচাহিদার উপর নজর রাখার জন্যে এক একটা আলাদা আলাদা প্রসেস চালু করা, এবং সেই প্রসেসগুলো আবার যাতে এ অন্যকে ঘেঁটে না-দেয় সেই ব্যবস্থা করার চেয়ে

বোধহয় সহজ হয় এমন একটা প্রোগ্রাম বানানো যা কখনোই থামবেনা। যতক্ষণ কম্পিউটার চলবে সেও চলবে, এবং কনস্টান্ট নজর রেখে যাবে সমস্ত প্রসেসের সদা বদলাতে থাকা সমস্ত চাহিদার উপর। কোনো একটা বিশেষ মেশিন চালু হওয়ার সময় কী কী ডিমন সেখানে চালু হবে এটা মেশিন থেকে মেশিনে আলাদা। ওই মেশিনে কী অপারেটিং সিস্টেম, কী কী প্রোগ্রাম সেখানে চলবে, কী কী কাজ করার আছে সেই মেশিনে, কী কী ধরনের হার্ডওয়ার সেখানে আছে — এই সবকিছুর উপর নির্ভর করে বদলে যায় যখদের তালিকা।

বুটপ্রক্রিয়ায় এবার আসে বিভিন্ন ব্যবহারকারীর জন্যে সিস্টেমকে প্রস্তুত করে তোলার পালা। এই ধাপে এসে গেটি চালু হয়। ইনিটের চালু করা গেটি নজর রাখতে শুরু করে টেলিটার্মিনাল বা কনসোলের উপর। গেটি কিন্তু একাধিক হতে পারে। গ্নু-লিনাক্সে, আগেই বলেছি, একটা প্রোগ্রাম চলা মানে তার মূল প্রোগ্রাম ফাইলটার একটা কপি একবার সক্রিয় হয়ে ওঠা। একটা প্রোগ্রামের একটা কপি একবার চলছে — এর মানে একটা প্রসেস। গেটি প্রসেস সেইরকম একাধিক চালু হতে পারে। ধরুন, মেশিনটা নেটওয়ার্কে লাগানো আছে। এই অবস্থায়, ডায়াল-ইন করে বা অন্য কোনো রকম নেটওয়ার্ক সংযোগ ব্যবহার ভৌগোলিক ভাবে দূরবর্তী কোনো ব্যবহারকারীও লগ-ইন করতে পারে সিস্টেমে। বাইরে থেকে কোনো ব্যবহারকারীর প্রবেশ ঘটছে কিনা কোনো কনসোলে সেটাও নজরে রাখার দায়িত্ব গেটির। তাই, গেটিকে নজর রাখতে হয় সিরিয়াল পোর্টেও। ব্যবহারকারীর প্রবেশ বা লগ-ইন করার গোটা খতিয়ানটা রাখে গেটি। যদি একাধিক টেলিটার্মিনাল মানে কনসোল থাকে তাহলে তাদের প্রত্যেকের জন্যেই এক কপি করে গেটি প্রসেস চালু হয়। গেটিই কনসোলে ফুটিয়ে তোলে লগিন প্রম্পট। এখানে আমরা ম্যানড্রেক ৯.১-এর লগ-ইন প্রম্পট দেখালাম। সমস্ত ডিমনগুলো এবং টেলিটার্মিনাল পিছু একটা করে গেটি প্রসেস চালু হওয়ার পর শেষ হয় ইনিট এর ১ নম্বর রানলেভেলের কাজ। রানলেভেল ২ বা ৩ বা ৫-এ যা যা প্রক্রিয়া চালু হয়, খেয়াল করে দেখুন একবার, তাদের মধ্যে কিন্তু রানলেভেল ১-এর সবগুলো প্রক্রিয়াই আছে। এবং আরো কিছু আছে তাদের প্রত্যেকটাতেই যা যা ১ নম্বর রানলেভেলে নেই। তাই রানলেভেল ১-এর প্রক্রিয়াগুলো চালু হবেই, বুট হওয়ার একটা স্তর অব্দি রানলেভেল ১-এর মত করে গোটাটা চালু থাকে। তারপর শুরু হয় '/etc' ডিরেক্টরিতে 'inittab' ফাইলে দেখানো ডিফল্ট রানলেভেলের জন্য নির্দিষ্ট তালিকা মেনে বাড়তি প্রক্রিয়া বা যখ বা পরিষেবাগুলো চালু হওয়ার কাজ। প্রতিটি রানলেভেলের সঙ্গে সঙ্গতি রেখে এই রকম একটা করে তালিকা তৈরি থাকে সিস্টেমে। আবার, নিজের ইচ্ছে হলে বিশেষ কোনো প্রসেসকে আপনি চালু বা বন্ধ করতে পারেন সেই তালিকা থেকে। ধরুন সেই ডিফল্ট রানলেভেল যদি ২ হয়, মানে 'inittab' ফাইলে লাইনটা যদি হয়, '`id:2:initdefault:`', তার মানে, এই ২ রানলেভেলের মত একাধিক ব্যবহারকারীর সিস্টেম চালু করতে হবে। অর্থাৎ, রানলেভেল ১-এর একক ব্যবহারকারী মানে মূল বা রুট ইউজার আছে এখানেও। তার সঙ্গে চালু হবে বাড়তি ব্যবহারকারীকে ঢুকতে এবং কাজ করতে দেওয়ার প্রক্রিয়া। সেগুলো চালু করে এবার রানলেভেল ২ শুরু হবে। এবার রুট ইউজার ছাড়াও অন্য একজন সাধারণ ইউজার এবার সিস্টেমে লগ-ইন করতে এবং কাজ শুরু করতে পারবে।

এখানেও কিন্তু গোটা বুটপ্রক্রিয়াটা না শেষ হতে পারে। ধরুন আপনার মেশিনে আপনি '/etc' ডিরেক্টরিতে 'inittab' ফাইলে ডিফল্ট রানলেভেল রেখেছেন ৩, মানে 'inittab' ফাইলে লাইনটা রয়েছে '`id:3:initdefault:`'। রানলেভেল ৩ মানে রানলেভেল ১ এর প্রক্রিয়াগুলো তো আছেই, তার সঙ্গে একাধিক ব্যবহারকারীর জন্যে প্রয়োজনীয় প্রক্রিয়াগুলো এবং নেটওয়ার্কের জন্যে দরকারি প্রক্রিয়াগুলোও চালু হবে। রানলেভেল ৩-এ বহু ইউজার আর নেটওয়ার্ক দুটোই আছে। তার মানে, রানলেভেল ১-এর প্রাথমিক প্রক্রিয়াগুলোর উপর এখন চালু হবে মাল্টিইউজার আর নেটওয়ার্ক সংক্রান্ত ডিমনগুলো অন্য প্রাসঙ্গিক প্রসেসরা। তবে শুরু হবে রানলেভেল ৩ — বুটপ্রক্রিয়া শেষ হবে। যদি কোনো কম্পিউটারে ব্যবস্থা এমনই করা থাকে যে সেখানে মাত্র একজন ব্যবহারকারী এবং নেটওয়ার্ক ইত্যাদি সেখানে নেই, (সেই গ্নু-লিনাক্স কিন্তু গ্নু-লিনাক্স নয়, একটা কবন্ধ মাত্র) সেখানে এই রানলেভেল ৩-এর ক্রিয়াগুলো আর ঘটবেনা। তবে এই কবন্ধ গ্নু-লিনাক্স-ও মাঝে মাঝে প্রয়োজন পড়ে। সিস্টেমে কোনো কেলো ঘটলে, সেটা সামলানোর জন্যে এরকম একটা মুন্ডুকাটা সিস্টেমে ঢুকে ত্রুটিগুলো সংশোধন করার দরকার পড়ে।

```
Mandrake Linux release 9.1 (Cooker) for i586
Kernel 2.4.19-24mdk on an i686 / tty1
localhost login:
```

৮।। তুমি কে গা? — লগ-ইন

থেবাই-এর স্পিংক্স-এর মত ঘাপলা ঘাপলা ধাঁধা জিগেশ করেনা বটে, কিন্তু অয়দিপাউসের সেই দানো যেমন অয়দিপাউসের কথা অয়দিপাউসের চেয়েও বেশি জানত, আমাদের এই সিস্টেমও তাই। আমাদের প্রত্যেকের কাজ, কাজের জায়গা, কাজের স্বাধীনতা, অধিকার, অনুমতি — এসব সে আমাদের নিজেদের চেয়েই অনেক বেশি খেয়াল রাখে। নথী বানিয়ে রাখে আমাদের প্রত্যেকের জন্যে। নথী মেলানোর জন্যে শুধু, প্রথমেই জানতে চায়, তুমি কে গা?

```
login:
```

প্রম্পটের এই চেহারাটা প্রমাণ করে, এই কনসোলটা এই মুহূর্তে ফাঁকা আছে। কোনো ইউজার এখন ঢুকতেই পারে সেখানে। কোলনের পর দপ দপ করতে থাকা কার্সর আমাদের জিগেশ করছে — নিজের নাম বসাতে হবে এবার। যে কোনো নাম নয়, যে নামে সিস্টেম আমাকে চেনে। সিস্টেমের মধ্যে যাবতীয় ব্যবহারকারীর প্রয়োজনীয় সমস্ত খুঁটিনাটির তালিকা বানানো থাকে। কোনো সিস্টেম ব্যবহার করার আগে সেই তালিকায় নিজের নাম তুলে দিতে হয়। ইউজারঅ্যাড (`useradd`) কমান্ড দিয়ে সিস্টেমের সেই তালিকায় নাম তোলার সময় যে নাম তাকে দেওয়া হয়েছিল, সেই নামটা এবার টাইপ করে দিতে হবে লগ-ইন প্রম্পটে। মেশিনে গ্নু-লিনাক্স কোনো ডিস্ট্রিবিউশন ইনস্টল করার মধ্যে অনেক সময় আমরা সরাসরি এই 'useradd' কমান্ডটা ব্যবহার করিনা। ইউজার যোগ করার উইনডোর ছবির মধ্যে নামটুকু টাইপ করে দিয়ে ইঁদুর ক্লিক করে নতুন ইউজারের নাম যোগ করি সিস্টেমে। কিন্তু গুই এর এই ইঁদুর-আর-ছবির প্রোগ্রামটা আসলে অন্তরালে এই ইউজারঅ্যাডকেই ব্যবহার করে।

যদি মেইনফ্রেম সিস্টেম হয়, যেখানে ব্যবহারকারীর সংখ্যা অনেক, সেখানে নিজেকে করতেই হয়না কাজটা। সিস্টেম অ্যাডমিনস্ট্রেটর নিজেই আমার প্রাথমিকতম পরিচয়টা তালিকায় ঢুকিয়ে দিয়েছে, সেটাও ওই ইউজারঅ্যাড ব্যবহার করে। আর আপনার আমার বা অন্য অনেকের নিজের টেবিলে রাখা নিজের ব্যক্তিগত পিসিতে, ব্যক্তিগত সিস্টেমে ব্যবস্থাটা নিজেই করতে হয়েছে, ইনস্টলেশনের সময় বা পরে। 'useradd' কমান্ডটাও একটা সিস্টেম প্রোগ্রাম, তাই রুট ছাড়া আর কেউ ব্যবহার করতে পারেনা। প্রথমে '`su`' কমান্ড দিয়ে রুট হয়ে, রুট বা অ্যাডমিনস্ট্রেটর-এর পাসওয়ার্ড দিয়ে, তবেই আর একজন ইউজার যোগ করা যায়। পরে আমরা আরো ভালো করে বুঝব, একটা গ্নু-লিনাক্স সিস্টেমের সুপারইজার বা রুট নামের ব্যবহারকারী হল চূড়ান্তভাবে একচ্ছত্র, কোনো কিছু করাতেই তার কোনো বাধা নেই, যে কোনো ফাইল যে কোনো সময়ে সে যে কোনো ভাবে বদলে নিতে পারে।

কিন্তু 'বদলে নিতে পারা' মানে তো 'বদলে ফেলতে পারা'-ও বটে। তাই, এমনকি নিজের ব্যক্তিগত সিস্টেমেও কখনো রুট হয়ে কাজ করতে নেই। ভুল প্রত্যেকের হয়। রুট হয়ে সিস্টেম ব্যবহার করার সময়, কোনো মুহূর্তের ভুলে, গোটা সিস্টেমকে অসংশোধনযোগ্য রকমে ঘেঁটে ফেলতে পারি আমরা, যে কোনো মুহূর্তে। এমনকি উড়িয়েও দিতে পারি গোটা সিস্টেমটা। এখানে বলে রাখি, যে কোনো ইউনিক্স-এই 'রুট' শব্দটা এই দুটো অর্থেই ব্যবহার হয়, এক, সুপারইউজার বা সর্বশক্তিমান চূড়ান্ত ব্যবহারকারী, আর দুই, রুট ডিরেক্টরি, যার কথা আমরা আগেই বলেছি লিলোর আলোচনায়, যে ডিরেক্টরির মধ্যে আছে আর সমস্ত ডিরেক্টরি। যাই হোক, যা বলছিলাম, তাই নিরাপদ উপায় হল এমনিতে নিজের জন্য সাধারণ একজন ব্যবহারকারী বা ইউজার বানিয়ে কাজ করা, এবং কোনো সিস্টেম প্রোগ্রাম ব্যবহার করার সময়টুকু 'su' করে সুপারইউজার বা রুট হয়ে নেওয়া। 'useradd' কমান্ড দিয়ে নিজের জন্যে একটা ইউজার যোগ করে নেওয়ার পরে, তার সিস্টেমে লগ-ইন করার সঙ্কেত বা পাসওয়ার্ডটাও বানিয়ে নিতে হয়। তার কমান্ডটা হল '`passwd`'। পরেও কখনো নিজের পাসওয়ার্ড বদলাতে চাইলে এই 'passwd' ব্যবহার করেই সেটা করা যায়। এই 'su', 'useradd', বা 'passwd' কমান্ডগুলো সম্পর্কে আর একটু জানতে চান? 'man' ব্যবহার করুন।

চার নম্বর দিনের থেকে মনে করুন, এক ধরনের বহু টার্মিনালের বহু ব্যবহারকারীর মেশিনে ব্যবহারযোগ্য ওএস হিশেবে বেড়ে উঠেছিল ইউনিক্স। ওরকম সিস্টেমে কম্পিউটার শেখার সুযোগ হয় খুব কম জনেরই। কিছু রিসার্চ ইনস্টিটিউট বা খুব কম দু-একটা অফিস। তারা অনেক ব্যাপক এবং গভীর ভাবে শেখে সিস্টেমকে। এরা এই পাঠমালার টার্গেট পাঠক নয়, এটা আমার মত গোলা লোকেদের জন্যে, যাদের কম্পিউটার শেখা এবং ব্যবহারটা এক ইউজারের এক কনসোলের ব্যক্তিগত স্ট্যান্ড-অ্যালোন পিসিতে, যারা সচরাচর অন্য মেশিনের সঙ্গে নেটওয়ার্কে সংযুক্তও থাকেনা। কখনো কখনো অনলাইন হয়, ইন্টারনেটে পৌঁছতে, কিন্তু নেটওয়ার্কটা সেখানে সর্বব্যাপী নয়।

এরকম সিস্টেমে সচরাচর একটা করে কনসোল থাকে। সেই একটা কনসোলকেও গ্নু-লিনাক্স একাধিক ভুতুড়ে পট মানে ভার্চুয়াল কনসোল করে তুলতে পারে, একের বেশি ভার্চুয়াল ব্যবহারকারী যেখানে লগ-ইন করতে পারে। যেন সেটা বহু কনসোলের বহু ইউজারের একটা মেশিন। কী করে এটা করা যায় সেই কথায় পরে আসছি। আগে ব্যাপারটা বুঝুন। এদের ভৌতিক ব্যবহারকারী বা ভার্চুয়াল ইউজার বলছি এই অর্থে যে একই কিবোর্ড থেকে কমান্ড দিয়ে একই স্ক্রিনে নিজেদের আদেশের ফলাফল দেখে কাজ চলছে সেখানে। বাস্তবে একটা লোকই সেটা করছে। কিন্তু সিস্টেম তাদের দেখছে একাধিক ব্যবহারকারী হিশেবেই। যেমন একটা বাস্তব বহু-টার্মিনাল বহু-ইউজার মেশিনে হত। ধরুন, আমার সিস্টেমে চারজন ব্যবহারকারী, 'atithi', 'dd', 'manu' এবং 'piu'। আর যে কোনো সিস্টেমেই একজন 'root' মানে সুপারইউজার বা রুট তো থাকবেই। এবার পাঁচটা ভৌতিক টার্মিনালে পাঁচজন ব্যবহারকারী হিশেবে লগ-ইন করে পাঁচটা কাজ ধরুন চালু করলাম। এবার এর, মধ্যে কোনো একটা ভৌতিক টার্মিনাল থেকে কমান্ড দিলাম, 'w'। 'w' কমান্ডটা একটা তালিকা তৈরি করে সিস্টেমে কে কে লগ-ইন করে রয়েছে এবং তারা কী কী করছে। দেখুন, প্রথম লাইনটা দেখাচ্ছে কটায় কমান্ডটা চালানো হয়েছে, কত সময় ধরে মেশিন চলছে, কত জন ইউজার ব্যবহার করছে, এবং শেষ কিছু সময়ে তারা মোট কতটা চাপ দিয়েছে সিস্টেমের উপর। দ্বিতীয় লাইনটা দেখুন, সেটা এক সারি হেডিং। প্রথম হেডিং 'USER' মানে ইউজারের নাম, দ্বিতীয়টা হল 'TTY' মানে কোন টার্মিনালে সে লগ-ইন করে রয়েছে, তৃতীয়টা হল 'LOGIN@' মানে কটার সময় সে লগ-ইন করেছে, চতুর্থটা 'IDLE' মানে কত সময় হল সে চুপচাপ বসে আছে, পঞ্চম আর ষষ্ঠটা হল সে সিপিইউকে কী রকম ব্যবহার করছে, আর শেষটা হল 'WHAT', মানে সে কী করছে। শুধু একটা জিনিষ খেয়াল করুন, এখানে 'root' চালিয়েছে 'w' কমান্ডটা, সে নিজে আছে চার নম্বর টার্মিনালে, আর সেই কমান্ডেরই ফলাফল এখানে লেখা, তাই একমাত্র তারই অলস সময় শূন্য, কারণ সেই মুহূর্তেই তো দেওয়া হয়েছে কমান্ডটা। এখুনি সবগুলো উপাদান বোঝার চেষ্টা করবেন না, প্লিজ। শুধু যে কথাটা বলছিলাম, দেখুন, একটা টার্মিনালের একা-একলা একটা ব্যক্তিগত পিসিকেও গ্নু-লিনাক্স সিস্টেম কিন্তু একটা পূর্ণাঙ্গ সিস্টেম হিশেবে নাড়াচাড়া করছে। যেখানে অনেক ব্যবহারকারী আছে, এমনকি নেটওয়ার্কও আছে, রিমোট লগ-ইন করা যায়। আপনার গ্নু-লিনাক্স সিস্টেমে 'man w' কমান্ড দিয়ে 'w' কমান্ডের ম্যানুয়াল পড়ে দেখুন, 'LOGIN@' মানে তৃতীয় স্তম্ভটা সেটাও দেখাত যদি কেউ রিমোট লগ-ইন করা থাকত এখানে। রিমোট লগ-ইন বলতে নেটওয়ার্কে সংযুক্ত অন্য কোনো মেশিন থেকে যদি কেউ এই সিস্টেমে লগ-ইন করা থাকত সেই খুঁটিনাটিটাও ধরে দিত 'w'।

```
18:01:52 up 8 min,  5 users,  load average: 0.00, 0.03, 0.02
USER     TTY       LOGIN@   IDLE   JCPU   PCPU  WHAT
dd       tty1      17:54    7:16   0.16s  0.08s links .
manu     tty2      17:54    1:36   0.79s  0.01s login
atithi   tty3      17:56    5:36   0.02s  0.02s -bash
root     tty4      17:56    0.00s  0.21s  0.00s w
piu      tty5      18:01   32.00s  0.02s  0.02s -bash
```

এবার ভৌতিক পট বা ভার্চুয়াল টার্মিনাল খুলে ভার্চুয়াল ইউজারের ভার্চুয়াল লগ-ইনের তরকিবটা বাতলে দেওয়া যাক। '<Ctrl>+<Alt>+<F1>' থেকে '<Ctrl>+<Alt>+<F6>'। মানে একইসঙ্গে কনট্রোল আর অল্ট আর একদম উপরের সারির একটা এফ-সুইচ, এক থেকে ছয়, পরপর আমাদের ভার্চুয়াল কনসোল ১ থেকে ৬ অব্দি নিয়ে যায় বা ফেরত আনে। কিবোর্ডে সবচেয়ে নিচুতলার বামপন্থীতম চাবিটাই হল কন্ট্রোল (<Ctrl>)। কমিউনিস্ট স্ট্রেটজ্যাকেটিং নিয়ে এটা কিবোর্ড-দেবতার কোনো গোপন মন্তব্য কিনা তা ফুকোদা বেঁচে থাকলে বলতে পারতেন। আর এই কনট্রোল চাবিটার থেকে ঠিক টু-স্টেপ-ব্যাক, দুই পা ডানদিকে পিছনে, দক্ষিণপন্থী বিচ্যুতিতে ভোগা সুইচটাই অল্ট (<Alt>)। এফ-ওয়ান (<F1>) থেকে এফ-সিক্স (<F6>) পাবেন একদম উপরের সারিটায় বাঁদিকে, এসকেপ সুইচটার পরেই। এই হল ছ-টা ভৌতিক পটের হিশেব। আর সাত নম্বর ভার্চুয়াল কনসোলটা তোলা থাকে গুই বা এক্স-উইনডোজের জন্যে।

এই ছয়পিস ভৌতিক পটের প্রত্যেকটাতেই আলাদা আলাদা করে লগ-ইন করতে পারে আলাদা আলাদা বা একই ব্যবহারকারী এবং কাজ করে চলতে পারে। যদি আপনি আমারই মত আইডেন্টিটি ক্রাইসিসে ভোগেন, নিজের নাম দেখতে খুব ভালোবাসেন, বারবার করে, তাহলে ছ-খানা কনসোলেই আলাদা আলাদা করে লগ ইন করুন। এবং

তারপর কমান্ড দিন 'who'। একটু আগের 'w' কমান্ডটার মতই 'who' কমান্ডটা ফুটিয়ে তোলে কে কে লগ-ইন করে রয়েছে সিস্টেমে, শুধু তাদের অতটা ডিটেইলস আর দেখায়না 'w' কমান্ডের মত। 'who' কমান্ড দেওয়ার পর দেখবেন ছবার আপনার নাম ফুটিয়ে তুলেছে সিস্টেম, পরপর ছ লাইনে। নিজেকে একই সঙ্গে ছজন বলে মনে করতে চাইলে, সেই কমান্ড-ও তো আগেই বলেছি, একটা কমান্ড সমাহার, 'who' আর 'wc' এই দুটো কমান্ডকে একসঙ্গে মিলিয়ে কাজ করে, 'who|wc -l'। এটা পরে আমরা পাইপিং-এর আলোচনায় দেখব, ইউনিক্স তথা গ্নু-লিনাক্স সিস্টেমের ইতিহাসে সবচেয়ে বিখ্যাত কমান্ডগুলোর একটা হল এই 'who|wc –l'। সিস্টেমে কতজন লগ-ইন করা আছে সেটা গুণে দেয়। এটা এর কারণ, আগেই যেমন বলেছি, প্রত্যেকটা ভৌতিক পট বা ভারচুয়াল কনসোলেই একখানা করে গেটি প্রসেস কাজ করছে। যাদের প্রত্যেকটাতেই এক এক জন আলাদা আলাদা আপনি লগ-ইন করেছেন, ছ বার, ছ-টা আলাদা জায়গায়। আপনার একটুও কৌতূহল হচ্ছেনা কমান্ডদুটো সম্পর্কে আর একটু জানার?

আবার ফেরত আসা যাক লগ-ইনের কথায়। প্রবেশ বা লগ-ইনের সময় সেই নামটাই দিতে হবে, যে নামে ইউজার বানানো এবং যোগ করা হয়েছিল। লগ-ইনের সময় যদি অন্য নাম দিই আমরা, ছোটদিদা মারা যাওয়ার পরে যে নামে আর কেউ ডাকেনি ততটা না-হলেও অন্তত যে নামে আপনি আপনাকে সিস্টেমে কোনোদিন ডাকেননি, সরি, এই স্পিংক্স তাকে চেনেনা, অতটা অর্ন্তযামী নয়। হতে পারে গ্নু-লিনাক্স, তাই বলে আশা করার তো একটা সীমা থাকবে। কিন্তু, মালটা হেভি ধূর্ত, অন্তর্যামী যে নয় সেটা বুঝতে দেবেনা আপনাকে। বা হয়ত একটু ইগো ক্রাইসিসে ভোগে, নিজের না-জানাটা স্বীকার করতে চায়না। যে নামই দিইনা কেন, এখন কিছু বলবেনা, পরের লাইনে সোনামুখ করে জানতে চাইবে

```
password:
```

আবার সেই দপদপ করতে থাকা নির্বাক প্রশ্ন। এর উত্তরে নিজের সঙ্কেত বা পাসওয়ার্ডটা দিতে হবে। যে পাসওয়ার্ড তাকে জমা দেওয়া হয়েছিল 'passwd' কমান্ড দিয়ে। মনে পড়ছে? এবার মজাটা এইখানে, ইউজারনেম আর পাসওয়ার্ড এই দুটো জিনিষের, যেকোনো একটা ভুল হলেই কম্পিউটার এবার জানাবে লগ-ইন ঠিক হয়নি। এবং নতুন করে আবার লগ-ইন করার জন্যে ব্যবহারকারীর নাম বা ইউজারনেম চাইবে। কিন্তু কখনোই আপনি আলাদা করে জানতে পারবেন না, দুটোর ভিতর ঠিক কোনটা ভুল হয়েছে।

```
login incorrect
login:
```

অর্থাৎ, আমরা একবারের তরেও জানতে পারলাম না, গন্ডগোলটা ঠিক কোথায় হয়েছিল ইউজারনেমে না পাসওয়ার্ডে। এটা ইউনিক্স সিস্টেমগুলোর একটা বাড়তি নিরাপত্তা ব্যবস্থা। বাইরে থেকে কেউ যদি ঢুকতে চায় সে যাতে একটুও বাড়তি তথ্য না পায়। আর লগ-ইন নেম এবং পাসওয়ার্ড দুটোই যদি ঠিক হয় এবার তিনি জানাবেন

```
Last login: Thu Oct 23 23:05:02 on tty1
```

দার্জিলিং মেলের ফার্স্ট ক্লাস কামরায় চৌকো জানলার সামনে ব্লাউজে পেন গুঁজে মাথা ঝুঁকিয়ে গালে টোল ফেলেনা সত্যিই, কিন্তু দেখেছেন, সিস্টেম আপনাকে কতটা 'মনে রেখে দেয়', আপনি নায়ক উত্তমকুমার না-হওয়া সত্ত্বেও। ঘ্যাঁচাঘ্যাঁচ জানিয়ে দিচ্ছে শেষ আপনি কখন সিস্টেমে ঢুকেছিলেন। এর সঙ্গে এক এক সিস্টেম এক এক সেট একস্ট্রা তথ্যও দেয়। কতটা তথ্য দেবে সেটাও নিয়ন্ত্রণ করা যায়। এই নিজের ইচ্ছে বা প্রয়োজন অনুযায়ী সিস্টেমের যে কোনো জায়গা বদলে নেওয়ার উপায়টা হল কনফিগারেশন ফাইল বদলে বদলে। গ্নু-লিনাক্স ব্যবস্থায় এটা প্রায় সর্বব্যাপী। পরে আমরা শিখব এটা, পাঠমালার একদম শেষ দিকে গিয়ে।

লগ-ইন করা মাত্র, কোনো কাজে হাত দেওয়ারও আগে, ইতিমধ্যেই, আপনার জন্যে একটা প্রোগ্রাম কিন্তু চালু হয়ে গেছে। সেটার নাম শেল। সচরাচর গড় গ্নু-লিনাক্স ব্যবহারকারীর শেলটার নাম ব্যাশ, কমান্ড ইন্টারপ্রিটার। আপনার দেওয়া কমান্ডকে যে ইন্টারপ্রিট করে পাঠায় সিস্টেমের কাছে। দিন নম্বর এক-এ আমরা অল্প আলোচনা করেছি এই নিয়ে। পরে আবার আসব। যে জায়গাটায় আপনি এখন আছেন, মানে কিবোর্ডে টাইপ করলেই সেই অক্ষরগুলো যেখানে ফুটে উঠবে, সেই জায়গাটার নামই কমান্ড প্রম্পট। একটা কথা খেয়াল রাখবেন, আপনি যে কমান্ডটা দিচ্ছেন, সেটা কিন্তু সরাসরি এখানে ফুটে উঠছে না। আপনি এন্টার মারা মাত্র আপনার কমান্ডটা পৌঁছে যাচ্ছে

সিস্টেমের কাছে। সেই কমান্ডের একটা প্রতিরূপ সে পাঠিয়ে দিচ্ছে স্ক্রিনে। আপনার দেখার জন্যে, শেষ কী কমান্ড গেছে। আর অন্য একটা প্রতিরূপ সে পাঠিয়ে দিচ্ছে ব্যাশ বা অন্য কোনো কমান্ড ইন্টারপ্রিটারের কাছে। কমান্ডের এই দুটো প্রতিরূপ একই সঙ্গে ব্যবহার করার টেকনিকাল নাম 'ডাবল ডুপ্লেক্স' (double-duplex)। এর আবার নানা বিশেষ তাৎপর্য আছে সিস্টেমে। জায়গাটা বেশ মজার। কিন্তু আমাদের এই পাঠমালায় আমরা অতটা দূর অব্দি যাবনা। নামটা দিয়ে রাখলাম যাতে কোথাও পেলে অপরিচিত না-লাগে। যাইহোক, যেখানে আপনার কমান্ডটা ফুটে উঠছে, সেই কমান্ড প্রম্পট-টা নানা চেহারা নিতে পারে।

`$_` বা `#_` বা `user@linux$_`

এদের কোনটা শেষ অব্দি ফুটে উঠবে সেটা নির্ভর করে কী ডিস্ট্রিবিউশন আপনি ব্যবহার করছেন তার উপর, মানে সেটা রেডহ্যাট না ম্যানড্রেক না ডেবিয়ান না সুজে না স্ল্যাকওয়ার ইত্যাদি, আর সেই মুহূর্তে আপনি সাধারণ ব্যবহারকারী না রুট, তার উপরেও। এই প্রম্পটের চেহারাটাও বদলে নেওয়া যায় সহজেই। এই খুঁটিনাটিগুলো এবং প্রম্পটের চেহারা থেকে সিস্টেমের যে কোনো কনফিগারেশনই কী ভাবে বদলাব, তার আলোচনা আসতে এখনো অনেকটা দেরি আছে। এইবার, এই সমস্ত ধাঁধার ঠিকঠাক উত্তর দেওয়ার পরে গ্নু-লিনাক্স সিস্টেমের স্পিংক্স আপনাকে ঢুকতে দিল। আপনি লগ-ইন করলেন, এবার আপনার কাজ করার জন্যে সে ফুটিয়ে তুলল কমান্ড প্রম্পট। এখন আপনি প্রম্পটে আছেন, আপনি কমান্ড দিয়ে যেই এন্টার মারবেন কম্পিউটার তার কাজ শুরু করবে। লগ-ইন শেষ হল।

নিচে আমার মেশিনে সুজে ৮.২-এ একবার লগ-ইনের গোটা লগ-ইন বানীসহ লগ-ইন প্রম্পট-টা তুলে দিলাম। ওই বানীটা আসলে ফরচুন কুকি, ওয়েফার মুড়ে দেওয়া কাগজে লেখা ভবিষ্যত বা অন্য কোনো বাণীর মত — গ্নু-লিনাক্স সিস্টেমের অনবদ্যতম চ্যাংড়ামোগুলোর একটা। এমন ব্যবস্থা করা আছে যে প্রতিবার আমি লগ-ইন করলেই একটি করে ফরচুন-বানী আমায় শোনায়, লগ-ইন মেসেজ সহ। কী করে ফরচুন বা অন্যান্য বাণী শোনানো শুরু বা বন্ধ করা যায়, সেগুলো আমরা পাব পাঠমালার শেষ মানে দশ নম্বর দিনে। এবার এই 'dd@linux:~>' লেখাটার পর জ্বলতে নিভতে থাকা কার্সরটাই আমার কমান্ড প্রম্পট, যেখানে আমি কমান্ড টাইপ করে এন্টার মারলে আমার মেশিন কাজ করবে।

```
Welcome to SuSE Linux 8.2 (i586) - Kernel 2.4.20-4GB-athlon (tty1).
linux login: dd
Password:
Last login: Thu Oct 23 08:55:42 on tty2
Have a lot of fun...
Anthony's Law of Force:
        Don't force it; get a larger hammer.
dd@linux:~>
```

৯।। লগ-ইনকে একটু খতিয়ে দেখা — পাসওয়ার্ড

সিস্টেমে আমি লগ-ইন করছি মানে গেটির কাছে নিজের পরিচয় জানাচ্ছি। গেটি তখন লগ-ইন (login) নামের প্রোগ্রামটা চালাচ্ছে। লগ-ইন প্রোগ্রামটা সিস্টেমের মধ্যে রাখা তালিকা থেকে আমার নাম আর পাসওয়ার্ড নিয়ে মিলিয়ে দেখছে, আমি আদৌ কোনো ইউজার কিনা, লগ-ইন করার এবং সিস্টেম ব্যবহার করার এক্তিয়ার আমার আছে কিনা। আলাদা করে নয়, দুটোকে একই সঙ্গে মেলাচ্ছে, এবং যদি এর একটাও না-মেলে, লগ-ইন করতে দিচ্ছেনা। আর যদি দুটোই মেলে, মানে, আমার এক্তিয়ার থাকে সিস্টেম ব্যবহার করার, আমাকে প্রবেশ অধিকার দিচ্ছে। এবং আমাকে প্রবেশ করানোর সঙ্গে সঙ্গে তাকে সেরে ফেলতে হচ্ছে একটু ঘরগোছানোর কাজ। মানে, আমার লগ-ইন সংক্রান্ত তথ্য সিস্টেমের 'মনে রেখে দেওয়া'। আর আমার জন্যে একপিস শেল চালু করে দেওয়া। এটুকু করেই সিন থেকে সরে যাচ্ছে লগ-ইন। ঠিক ওই বায়োস আর লিলোর মতই, গেটি আর লগ-ইন একই প্রোগ্রাম হতেই পারত। না-হয়ে তারা আলাদা প্রোগ্রাম হল ওএস বিবর্তনের কিছু ঐতিহাসিক কারণে।

কিন্তু আমার হাতে ওই একখানা খোসা মানে শেল ধরিয়ে দেওয়ার আগে সিস্টেমকে আরো কিছু কাজ করে নিতে হয়। যাতে সে বুঝে নিতে পারে, কোন কোন ফাইলে বা ডিরেক্টরিতে কাজ করার অনুমতি আমার আছে, এবং পরে, কাজ করে চলাকালীন আমি যখনি কোনো ফাইল বা ডিরেক্টরিতে কাজ করতে চাইব, তখনি মিলিয়ে নিতে পারে। এটা কোনো গ্নু-লিনাক্স তথা ইউনিক্স সিস্টেমে সবচেয়ে গুরুত্বপূর্ণ ব্যাপারগুলোর একটা। তবে পুরোনো ইউনিক্সের সঙ্গে গ্নু-লিনাক্সের এই নিরিখে কিছু পার্থক্যও আছে। গ্নু-লিনাক্স গোটা নিরাপত্তা ব্যবস্থাটাই দাঁড়িয়ে আছে এর উপর। আমি নিজেকে চেনাচ্ছি আমার নাম আর পাসওয়ার্ড দিয়ে। এই পাসওয়ার্ড আর নামকে সিস্টেম মিলিয়ে নিচ্ছে '/etc' ডিরেক্টরিতে 'passwd' নামের একটা ফাইলের সঙ্গে। এই ফাইলটা, '/etc/passwd', ঠিক ওই '/etc/lilo.conf' ফাইলের মতই, একটা টেক্সট ফাইল, যার মধ্যে পরপর কয়েক লাইনে প্রতিটি ব্যবহারকারীর বিষয়ে প্রয়োজনীয় তথ্যগুলো রাখা। এখানে এদের দুজনেরই নামের আগে /etc অংশটা জুড়ে দেওয়ার মানে এই যে একটা সিস্টেমে এই দুটো ফাইলকেই পাওয়া যায় রুট ডিরেক্টরি বা ' / '-এর মধ্যে 'etc' নামের ডিরেক্টরিতে। এর আগে অব্দি যাকে লিখছিলাম 'অমুক' ডিরেক্টরির 'তমুক' ফাইল, তাকে বোঝানো যেত 'অমুক/তমুক' ফাইল লিখে। এখানে আমরা লিখেছি '/অমুক/তমুক' ফাইল। এতে বাড়তি এইটুকু বলা আছে যে, 'অমুক' ডিরেক্টরিটা আছে রুট ডিরেক্টরিতে, গ্নু-লিনাক্সে যার নাম '/'। পরে অনেক ভালো করে বুঝতে হবে এসব। এখন একটু সেই '/etc/passwd' ফাইলটা থেকে কয়েক লাইন পড়া যাক।

এটা আমার মেশিনের '/etc/passwd' ফাইল, প্রথম লাইনে আর শেষ চার লাইনে 'atithi', 'dd', 'manu', আর 'piu' ছাড়া আর সবগুলো লাইনেই ব্যবহারকারী হিশেবে যাদের উল্লেখ তাদের প্রত্যেকেই সিস্টেমের বানানো — নিজের কাজ করার স্বার্থে সিস্টেম এই প্রোগ্রামগুলোকে ইউজার বানায়, কারণ, স্বাভাবিক ইউজারদের মত এরাও বিভিন্ন রসদকে ব্যবহার করে, কাজে লাগায়। একদম প্রথম লাইনেই রয়েছে সুপারইউজার বা রুট।

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/bin/bash
daemon:x:2:2:Daemon:/sbin:/bin/bash
lp:x:4:7:Printing daemon:/var/spool/lpd:/bin/bash
mail:x:8:12:Mailer daemon:/var/spool/clientmqueue:/bin/false
news:x:9:13:News system:/etc/news:/bin/bash
uucp:x:10:14:Unix-to-Unix CoPy system:/etc/uucp:/bin/bash
games:x:12:100:Games account:/var/games:/bin/bash
man:x:13:62:Manual pages viewer:/var/cache/man:/bin/bash
at:x:25:25:Batch jobs daemon:/var/spool/atjobs:/bin/bash
wwwrun:x:30:65534:WWW daemon apache:/var/lib/wwwrun:/bin/bash
ftp:x:40:49:FTP account:/srv/ftp:/bin/bash
gdm:x:50:15:Gnome Display Manager daemon:/var/lib/gdm:/bin/bash
postfix:x:51:51:Postfix Daemon:/var/spool/postfix:/bin/false
gnump3d:x:63:65534:GNUMP3 daemon:/var/lib/nobody:/bin/false
sshd:x:71:65:SSH daemon:/var/lib/sshd:/bin/false
ntp:x:74:65534:NTP daemon:/var/lib/ntp:/bin/false
vdr:x:100:33:Video Disk Recorder:/var/spool/video:/bin/false
nobody:x:65534:65533:nobody:/var/lib/nobody:/bin/bash
dd:x:500:100:dipankar das:/home/dd:/bin/bash
manu:x:501:100:Sriparna Das:/home/manu:/bin/bash
atithi:x:502:100: atithi:/home/ atithi:/bin/bash
piu:x:503:100:Smita Bhadra:/home/piu:/bin/bash
```

এই '/etc/passwd' ফাইলের প্রত্যেকটা লাইনেই, লক্ষ্য করুন, একটা কাঠামো আছে। মানে একই ধরনের বিষয় প্রতিটা লাইনেই এসেছে। এরসবগুলো খুঁটিনাটিতে আমরা এখন যাবনা, পরে আসবে। শুধু দেখুন প্রত্যেকটা লাইনেই শুরুতে ব্যবহারকারীর ইউজারনেম, তারপর একটা 'x' , যেটা আসলে তার পাসওয়ার্ডের সূচক। কোনো কোনো সিস্টেমে, যদি নিরাপত্তা নিয়ে ততটা মাথা না-ঘামানো হয়, পাসওয়ার্ডগুলো এখানেই থাকতে পারে, কিন্তু এই সিস্টেমে সেটা আছে অন্য একটা ফাইলে, যার নাম '/etc/shadow'। বলতে পারবেন, এই 'shadow' ফাইলটা আছে কোন ডিরেক্টরিতে? সেই ডিরেক্টরিটা আছে কোন ডিরেক্টরিতে? এই '/etc/shadow' ফাইলটা আবার সাধারণ ব্যবহারকারীর দেখার অনুমতি নেই। এখানে আপনাদের দেখাব বলে ফাইলটার কয়েকটা লাইন কপি করতে গেলাম। সিস্টেম করতে দিলনা। ব্যাটার সাহস ভাবুন, এদিকে যে ইলেকট্রিকে চলছে তার বিল দিই আমি। এখুনি সুইচ অফ

করে দিলেই পুরো দাঁত ছিরকুটে পড়বে। কপিটা আমায় ঘুরপথে করতে হল। প্রথমে সুপারইউজার হলাম, 'su' কমান্ড দিয়ে। এন্টার মারার সঙ্গে সঙ্গে সিস্টেম আমার কাছে রুট পাসওয়ার্ড জানতে চাইল। দিলাম সেটা। সুপারইউজার মোড চালু হল। তারপর এই ফাইলটা কপি করলাম অন্য একটা জায়গায় অন্য একটা ফাইলে। তারপর সেই ফাইলটার মালিকানা সঁপে দিলাম আমাকে, মানে, 'dd' নামের ইউজারকে। কারণ, সেটাও তো কপি করেছে রুট, রুটেরই মালিকানা, তাই আমি স্বাভাবিক অবস্থায় সেটা কিছু করতে পারব না। মালিকানা বদলানোর কমান্ডটার নাম 'chown'। ব্যস্ত হবেন না, এগুলোয় আমরা পরে আসব, এখানে একটু বললাম আগের থেকে একটা আন্দাজ দিয়ে রাখার জন্যে। আর নিজে নিজে কৌতূহল মেটানোর কোনো উপায় জানেন নাকি আপনি? এক্ষেত্রে এই পুরো পাঁয়তাড়াটা কষা গেল, ব্যক্তিগত পিসি হওয়ায় এর সুপারভাইজর আর ইউজার দুজনেই একই লোক, মানে আমি। তাই রুট পাসওয়ার্ড আমি জানি। আমি যখন আমার নিজের কাজের জন্যে সিস্টেম ব্যবহার করছি তখন আমি ইউজার, কারণ যা বললাম, রুট হয়ে কাজ করা সবসময়ই অত্যন্ত বিপজ্জনক। যে কোনো ডিরেক্টরিতে যে কোনো ফাইলে রুট-এর যে কোনো কাজের পারমিশন থাকে বলে সিস্টেম যখন তখন ঘেঁটে ফেলতে পারে রুট, একটু কমান্ডের ভুলে। ইনস্টলেশনের সময়ই দেখবেন, মানা করে দেয়, ওহে খোকা, নিজের জন্যেও একপিস ইউজার আইডেন্টিটি বানিয়ে নাও। ফাইলদের তথা ডিরেক্টরিদের এই অনুমতি/মালিকানা মানে পারমিশন/ওউনারশিপ — এগুলো আমাদের পরে খুবই ভালো করে বুঝতে হবে। এখন জাস্ট বুড়ি ছুঁয়ে গেলাম।

পাসওয়ার্ডগুলো কিন্তু এখানেও তাদের অবিকল চেহারায় নেই, আছে সঙ্কেতবদ্ধ বা এনক্রিপ্টেড চেহারায়। নইলে আপনি এখুনি আমার পাসওয়ার্ড জেনে ফেলতেন। এই এনক্রিপশন ব্যাপারটা বেশ ইন্টারেস্টিং। রোজার বরকে টেররিস্টরা তুলে নিয়ে গেছিল, সেই বরের পেশা ছিল এই এনক্রিপশন ভাঙা, ডিক্রিপশন। ক্লদ শ্যাননের এই এলাকায় গবেষণার কথা লিখেছিলাম তিন নম্বর দিনে, মনে আছে? এনক্রিপশনের ক্রিয়াপদ্ধতি বা বা অ্যালগরিদমটা এমন যে, কোনো একটা টেক্সটকে এনক্রিপ্ট করে সঙ্গেতবদ্ধ আকার দেওয়াটা খুবই সহজ, কিন্তু উল্টোটা, মানে সঙ্কেতবদ্ধ চেহারা থেকে তার প্রারম্ভিক আকারে ফিরিয়ে আনা প্রায় অসম্ভব। ভয়ঙ্কর শ্রম ও সময় ব্যয় করেও নিশ্চিত ফলাফল দেওয়াটা খুবই কঠিন। সিস্টেমও তাই এখানে একটা কুপথ মানে শর্টকাট ব্যবহার করে। প্রথমবার যখন ঢোকানো হচ্ছে সিস্টেমে, পাসওয়ার্ডটাকে এনক্রিপ্ট করে নেওয়া হয়। এবার সেই এনক্রিপ্টেড চেহারাটাকে লিখে রাখে '/etc/shadow' ফাইলে। এবার, যখনই আমাকে লগ-ইন-এর সময় পাসওয়ার্ড দিতে হচ্ছে, সিস্টেম সেই পাসওয়ার্ডটাকে আবার এনক্রিপ্ট করে দেখে নিচ্ছে যে, '/etc/shadow' ফাইলে রাখা চেহারাটার সঙ্গে সেটা মিলছে কিনা। এনক্রিপ্টেড চেহারাটাকে ভেঙে স্বাভাবিকটার সঙ্গে মেলানোর সে কোনো চেষ্টাই করছেনা। কারণ, যা বললাম, সেটা প্রায় অসম্ভব। এখানে আর একটা ব্যাপার আছে — মূল পাসওয়ার্ড আর তার এনক্রিপ্টেড চেহারার মধ্যে কোনো আম-দুধ-মেশামেশি বা ওয়ান-ইজ-টু-ওয়ান করেসপন্ডেন্স নেই। ক এবং খ দুজনেরই এনক্রিপ্টেড চেহারা একই গ হতে পারে, যদিও এর উল্টোটা ঠিক নয়, মানে একই ক থেকে দুটো আলাদা এনক্রিপশন গ আর ঘ পেতে পারিনা। তার মানে এনক্রিপ্টেডকে ডিক্রিপ্ট করে একাধিক চেহারা পাওয়া যেতে পারে, কিন্তু মূলটাকে যতবারই এনক্রিপ্ট করিনা কেন, ওই একই চেহারা পাব প্রত্যেকবার। তাই লগ-ইনের সময় দেওয়া পাসওয়ার্ডটাকে স্যাট করে একবার এনক্রিপ্ট করে দেখে নিলেই সিস্টেমের কাজ খালাস। 'login' প্রোগ্রাম প্রথমে আমাদের দেওয়া ইউজারনেম-টাকে মিলিয়ে নেয় '/etc/passwd' থেকে। তারপর, দেওয়া পাসওয়ার্ডটাকে এনক্রিপ্ট করে মিলিয়ে নেয় '/etc/shadow' থেকে। দুটোর একটাও যদি না-মেলে, তখন কনসোলে ফুটিয়ে তোলে ওই বাণী, 'login incorrect'। যদি দুটোই মিলে যায় তখন শেলকে জাগিয়ে তোলে এবং কনসোলে ফুটিয়ে তোলে কমান্ড প্রম্পট। এখানে দেওয়া '/etc/passwd' ফাইল থেকে দেখুন সব কজন ইউজারেরই জন্য লেখা শেলটা হবে '/bin/bash', মানে, ব্যাশ নিজেও তো একটা প্রোগ্রাম, সেই প্রোগ্রামটা আছে '/bin' ডিরেক্টরিতে, সেখান থেকেই চালাতে হবে তাকে। ঠিকানাটায়, ডিরেক্টরির নাম 'bin', তার আগে একটা '/' আছে। এর মানে কী?

আমার সিস্টেমের '/etc/shadow' ফাইল থেকে চারজন ইউজারের চারটে লাইন তুলে দিলাম।

```
dd:pV9quSXzPFD76:12296:0:99999:7:::
atithi:7NLMeKZlDh442:12298:0:99999:7:::
manu:XJNkQuu8YRCSU:12298:0:99999:7:::
piu:Lkut3i/V2Phzw:12353:0:365:7:-1::
```

একবার সফলভাবে লগ-ইন করে গেলাম মানে এবার যে যে ডিরেক্টরিতে আমার অধিকার আছে সেখানে আমি কাজ শুরু করতে পারব, আমার অধিকারাধীন ফাইলদের আমি বদলাতে পড়তে বা চালাতে পারব। এখানে এই মালিকানা এবং অনুমতির সঙ্গে গ্রুপ বা দলের একটা ধারণা আছে, সেটায় পরে আসছি। আর শুধু একটাই কথা বলার আছে লগ-ইনের কাজ নিয়ে। সেটা এই যে, শুধু শেলকে জাগিয়ে তোলা বা কমান্ড প্রম্পটকে ফুটিয়ে তোলাই না, আরো একটা কাজ করে লগ-ইন। ভারি পুণ্যের কাজ, পথহারা দিশাহারা মানুষকে তার বাড়িতে পৌঁছে দেওয়া, নিজের ঘরে বসিয়ে দেওয়া। প্রথম প্রথম একটা গ্নু-লিনাক্স সিস্টেমে ঢুকে যার দিশাহারা লাগেনা, এমন মানুষের স্বপ্ন দেখা খুবই স্বাস্থ্যকর অভ্যেশ — ওইসব স্বপ্ন থেকেই পৃথিবী বদলায়, বাস্তবতা বদলায়, বিপ্লব হয়, প্লেগের ওষুধ আবিষ্কার হয়। কিন্তু আমাদের আম-জনতাদের কাছে লগ-ইন একজন পরিত্রাতার মত — যাক বাবা, নিজের ঘরেই আছি। ইউজারকে তার নিজের নিজের হোমে, নিজের এলাকায় পৌঁছে দিয়ে যায় 'login'।

লগ-ইন হয়ে গেছে, কমান্ড প্রম্পট ফুটে উঠেছে, এখন যদি জানতে চাই, ফাইলসিস্টেমের ডিরেক্টরিজালের শাখা-প্রশাখার ঠিক কোথায় আছি, তার কমান্ড '`pwd`', প্রিন্ট-ওয়ার্কিং-ডিরেক্টরি। লিখে এন্টার মারা মাত্র, আমার বেলায়, কমান্ড প্রম্পটে ফুটে ওঠে, '/home/dd' — আমার ঘর। গ্নু-লিনাক্স সিস্টেমে '/home/dd' হল 'dd' নামের ইউজারের বাস্তুভিটে। 'dd'-র মালিকানার এলাকা। ইউজার 'manu'-র বেলায় সেটা হবে '/home/manu'। এভাবেই ঘটে প্রতিটা ইউজারের। সবারই নিজের ঘর ওই '/home' ডিরেক্টরিতে তার নিজের নামের একটা সাবডিরেক্টরি। অর্থাৎ 'login' ইউজারকে পৌঁছে দেয় তার নিজের ঠিকানায়। এবার সে '`ls`' কমান্ড লিখে এন্টার মারলে, মানে ফাইল ডিরেক্টরির লিস্ট দেখাতে বললে, নিজের হোমে নিজের শাসনাধীন এলাকায় নিজের মালিকানার ফাইল আর ডিরেক্টরিদের তালিকা দেখতে পায়। এবার '/etc/passwd' ফাইলের শেষ লাইনটা তুলে একটু বোঝার চেষ্টা করি।

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

এর প্রথমটা, 'piu' হল ইউজারের নাম, যে নাম দেওয়া আছে সিস্টেমে, যে নামে সে লগ-ইন করে। প্রথা অনুযায়ী এটা ছোট হাতের অক্ষরেই হয়। এর পর যতিচিহ্ন, ':'। এখানে যে কটা ফিল্ড বা আলাদা আলাদা ক্ষেত্র আছে, তাদের প্রত্যেকটাকেই পরেরটা থেকে আলাদা করা হয়েছে এই যতিচিহ্ন ':' দিয়ে। শুধু এই ফাইল নয়, আরো বহু ফাইলেই এই যতিচিহ্ন ব্যবহার হয়, পরে দেখবেন। পরের ফিল্ডটা 'x', একটা প্রতীক যা পাসওয়ার্ডের উপস্থিতিকে চিহ্নিত করছে। পাসওয়ার্ডটা বা তার সাঙ্কেতিক চেহারাটা কিন্তু এখানে নেই, আছে '/etc/shadow' ফাইলে। তৃতীয় ফিল্ডটা, '503', ইউজারের ব্যক্তি-পরিচিতি (**U**ser-**Id**entity — uid)। এই ইউআইডি প্রত্যেকের আলাদা আলাদা। তার পরেরটা, '100' হল তার দলপরিচিতি (**G**roup-**I**dentiy — gid)। এই ফাইলে শেষ চারজনেরই 'gid' এক। তার পরের ফিল্ডটা, 'Smita Bhadra', ইউজারের পুরো নাম। এর পরের ফিল্ড, '/home/piu' হল এই ইউজারের হোম-ডিরেক্টরি। শেষ ফিল্ডটা, '/bin/bash' হল তার শেলের নাম, গ্নু-লিনাক্সের ডিফল্ট শেল ব্যাশ। মানে, আলাদা করে না-বদলালে সিস্টেম এটাকেই একজন ইউজারের শেল বলে ধার্য করে নেয়। ইউআইডি বা জিআইডি, এবং অন্য ডিটেইলগুলোয় আমরা পরে আসব ব্যবহারকারীদের অনুমতি এবং অধিকারের আলোচনায়। জাস্ট একটু কৌতূহল হিশেবেই এখানে আমি এই পাসওয়ার্ড ফাইলের ম্যানুয়াল পাতা থেকে একটা কুচো অংশ একটু তুলে দিচ্ছি। গ্নু-লিনাক্সে যে কোনো কমান্ডেরই ম্যানুয়াল পড়তে হয় ম্যান (**MAN**ual — man) কমান্ড দিয়ে, আগেই বলেছি। আর কোনো কনফিগারেশন ফাইলের ম্যানুয়াল পড়তে হয় '`man 5`' দিয়ে, কারণ, ম্যানুয়াল পাতাগুলোর ৫ নম্বর সেকশনে থাকে বিভিন্ন কনফিগারেশন ফাইলের হদিশ। এখানে আমি এই ফাইলটা পেতে গিয়ে যে কমান্ডটা দিয়েছিলাম সেটা হল, '`man 5 passwd > passwd.text`'। অর্থাৎ, ম্যানুয়ালের পাঁচ নম্বর সেট থেকে পাসওয়ার্ড ফাইলের পাতাগুলোকে একটা ফাইলে রূপান্তরিত করো যার নাম 'passwd.text'। আজকের একদম গোড়ার দিকের চার নম্বর সেকশনের আলোচনার রিডাইরেকশন বা চালান করার আগের উদাহরণটার সঙ্গে মেলান। শুধু '`man 5 passwd`' লিখে এন্টার মারলে যে পাতাগুলো ও আমায় স্ক্রিনে দেখাত, সেই পাতাগুলোকে রিডাইরেক্ট করে দিল, চালান করে দিল এই 'passwd.text' ফাইলে। আমি সেই ফাইল থেকে আমি একটা অংশ এখানে তুলে দিলাম, বিভিন্ন ফিল্ডগুলো কী ভাবে থাকে তারই নির্দেশটা, এর আগেই '/etc/passwd' ফাইল নিয়ে যে আলোচনাটা করলাম আমরা, তার সঙ্গে মিলিয়ে পড়ুন। এর থেকে একটু ম্যান পড়ার অভ্যেসও হবে। ম্যান পড়ার ব্যাপারটা গ্নু-লিনাক্সে এমন বিরাট জায়গা নিয়ে থাকে যে আগে চ্যাংড়ামো করে লিনাক্সকে আরটিএফএম (**R**ead-

The-**F**ucking-**M**anuals) সিস্টেম বলা হত। গ্নু-লিনাক্সের ব্যাপক সুবিধে এটাই। সবকিছু সমস্তকিছু জানার উপায়ই দেওয়া আছে সিস্টেমের মধ্যেই।

```
PASSWD(5)
NAME
       passwd - The password file
DESCRIPTION
       _passwd_  contains  various  pieces  of information
for each user account.
Included is
             Login name
             Optional encrypted password
             Numerical user ID
             Numerical group ID
             User name or comment field
             User home directory
             User command interpreter
The password field may not be filled if shadow passwords
have been  enabled.
If shadow passwords are being used, the encrypted
password will be found in _/etc/shadow_.
```

১০।। সিস্টেম মানে জাগ্রত এবং ঘুমন্ত প্রসেসের চিড়িয়াখানা

প্রথম যেবার টপ (top) কমান্ডটা ব্যবহার করলাম, আমার ভারি একটা অপরাধবোধ হয়েছিল। টপ কমান্ডটা নিরবচ্ছিন্ন ধারাবাহিক ভাবে একটা জ্যান্ত তালিকা দিতে থাকে স্ক্রিনে। জ্যান্ত তালিকা মানে বাস্তব সময়ের সঙ্গে সঙ্গে বদলাতে থাকে। কী কী প্রসেস বা প্রক্রিয়া সেই মুহূর্তে চলছে সিস্টেমে, তারা কতটা সিপিইউ ব্যবহার করছে, কতটা মেমরি ব্যবহার করছে, কতটা সময় ধরে চলছে, সেই তালিকাটা দেখায় টপ। এবং তালিকাটা মুহূর্মুহূ বদলাতে থাকে স্ক্রিনে। একটা প্রসেস একে ছিল, চারে নেমে যাচ্ছে, চারেরটা দুইয়ে উঠে যাচ্ছে, এইরকম চলতেই থাকে। টপ চালানোর আগে বহুক্ষণ ধরে সেদিন চুপচাপ কম্পিউটারের সামনে বসে বসে ঝিমোচ্ছিলাম, যেমনটা আমি প্রায়ই করে থাকি, যদিও স্বীকার করিনা সচরাচর। এক নম্বর ভারচুয়াল টার্মিনালে (tty1) কী একটা ম্যান পেজ খোলা ছিল আর ব্যাকগ্রাউন্ডে দু নম্বর টার্মিনালে (tty2) এমপ্লেয়ারে নুসরতের একটা এমপিথ্রি বাজছিল, আসলে কিছুই করছিলাম না, হঠাৎ শখ হল, টপটা চালালাম তিন নম্বর টার্মিনালে (tty3) গিয়ে। কী ভয়ঙ্কর — আমার এই কিছুই না-করে ঝিমোনোটাও আহা অপারেটিং সিস্টেম বেচারার এই বীভৎস কর্মকাণ্ডের বিনিময়ে। চলতে থাকা প্রসেসের মোট তালিকাটা বিকট, অগণ্য বললেই ঠিক হয় মোট সংখ্যাটা। আপনি করে দেখুন। যদি না-চমকান, কথা দিচ্ছি, কদিন ধরে হিং-গোলমরিচ-আদা দিয়ে একটা মুরগির রেসিপি মাথায় ঘুরছে, কাউকে আগে একটা খাইয়ে দেখতে হবে, মরে টরে গেল কিনা, তবে নিজে খাওয়ার রিস্ক নেব, আপনাকে ফ্রিতে খাইয়ে দেব। এই অগণ্য প্রসেসের অপরিমেয় জটিলতাকে নিজেদের মাথায় ধারণ করার থেকে আমাদের বাঁচিয়ে দেয় শেল। গ্নু-লিনাক্সের এটা একটা ভারি গুরুত্বপূর্ণ বৈশিষ্ট্য যে কারনেল আর শেল এখানে সম্পূর্ণ স্বতন্ত্র দুটো প্রোগ্রাম। শেল আর কারনেলের এই সম্পূর্ণ আলাদা দুটো প্রোগ্রামের মধ্যে একটা জীবন্ত সংযোগও থাকে — সিস্টেম কল হল সেই যোগাযোগ। মনে পড়ছে সিস্টেম কলের আলোচনা, দুই নম্বর দিন থেকে? গ্নু-লিনাক্সে একাধিক শেল থাকতে পারে — আলাদা আলাদা প্রয়োজন এবং পছন্দ মোতাবেক — ব্যাশ শেল, সি শেল, কর্ন শেল, জি শেল ইত্যাদি। পরে আমরা আসব এইসব নিয়ে বিশদ আলোচনায়।

বুটের পুরো কর্মকাণ্ডটা এবার মনে মনে ঝালিয়ে নিন — কী ভাবে প্রথমে বায়োস, তারপর লিলো, তারপর ইনিট, তারপর কারনেল, তারপর গেটি, তারপর লগ-ইন পরপর তাদের কাজ শুরু এবং শেষ করে, শেলের হাতে পতাকা হ্যান্ডওভার করে, একে একে সিন থেকে কেটে যায় প্রত্যেকেই। সিনে মানে স্ক্রিনে রয়ে যায় শুধু শেল, মানে কনসোলের গম্ভীর কালো সিরিয়াস পর্দায় ফুটে থাকে কমান্ড প্রম্পট। এবার একটু বোঝার চেষ্টা করা যাক, স্ক্রিনের পিছনে তখন জনান্তিকে কী ঘটছে। পাওয়ার সুইচ অন করার পর বায়োস থেকে শুরু করে এই কমান্ড প্রম্পটে আপনার কোনো কমান্ড টাইপ করে এন্টার মারা, মানে প্রোগ্রাম চালানো অব্দি গোটা ব্যাপারটাকে একটা জ্যান্ত

চিড়িয়াখানা বলে মনে হচ্ছে না? বোধহয়, একটুও ভুল মনে হচ্ছে না। মাল্টিপ্রোগ্রামিং অপারেটিং সিস্টেমের বিবর্তন নিয়ে ফ্রেড ব্রুকস-এর '৯৬-এ লেখা একটা বইয়ের প্রচ্ছদে ছিল একটা প্রাগৈতিহাসিক আলকাতরার খাদানে পড়ে যাওয়া ডাইনোসরদের ছবি — পর্দার পিছনে ঘটতে থাকা ইউনিক্স-এর জুরাসিক জু। বুট থেকে শুরু করে প্রোগ্রাম চালানো অব্দি এই চিড়িয়াখানার প্রতিটি প্রসেস প্রতীক্ষা করে রয়েছে — কখন তাদের হাতে কোনো কাজ দেওয়া হবে। অপেক্ষা করে আছে ইভেন্টের — ঘটনার — কমান্ডে বা শেলে বা কারনেলের সঙ্গে তাদের সিস্টেম কলে — কিছু একটা ঘটবে, যাতে তাদের প্রয়োজন পড়বে, ডাক পড়বে। এই ইভেন্ট বা ঘটনা মানে হতে পারে আপনি কোনো চাবি টিপলেন কিবোর্ডে, বা ইঁদুর নড়ালেন বা ইঁদুরের পেটে কোনো সুইচ টিপলেন, বা মেশিন যদি নেটওয়ার্কিত থাকে তাহলে হয়তো তড়িৎপথ বেয়ে কোনো তথ্য বা নির্দেশ বয়ে এলো দূরবর্তী কোনো মেশিন থেকে। ইভেন্ট ঘটা মাত্র কাজ শুরু করে দেবে সেই প্রসেস, ওই ধরনের ইভেন্টে যার কাজ শুরু করার কথা, কিন্তু যতক্ষণ-না ইভেন্টটা ঘটছে, প্রসেসরা কিন্তু মরে যায়নি, তারা সব ঘাপটি মেরে খাপ পেতে বসে আছে।

এই প্রসেসগুলোর সবচেয়ে বড় প্রসেস কিন্তু খোদ কারনেল নিজেই। সবচেয়ে বড় এবং সবচেয়ে বিশেষ। কারণ, সে আর যাবতীয় ইউজার প্রসেস বা ব্যবহারকারী পদ্ধতিগুলোকে নিয়ন্ত্রণ করে — যখন তারা সক্রিয় হয়ে ওঠে, শেষ অব্দি কাজ করার সুযোগ পায়, তাদের অপেক্ষমান অবস্থা ছেড়ে। এবং কারনেলই সেই একমাত্র পদ্ধতি যাকে সরাসরি হার্ডওয়ারদের সঙ্গে কথা বলতে হয়। ইউজার প্রসেসগুলো কারনেলের কাছে তাদের অনুরোধ জানায় তাদের প্রয়োজন মোতাবেক। ধরুন একটা প্রোগ্রামের কিছু কিবোর্ড থেকে আসা ইনপুট দরকার, সরাসরি ব্যবহারকারীর কাছ থেকে কিছু একটা এখুনি পেতে হবে তাকে, নইলে সে কাজ করতে পারছে-না। এবার সে ওই কিবোর্ড ইনপুটের যাজ্ঞা করে দরখাস্ত পাঠাবে কারনেলের কাছে। বা ধরুন তাকে স্ক্রিনে কিছু লিখতে হবে, মানে স্ক্রিনে কিছু আউটপুট পাঠাতে হবে। তার জন্যেও কারনেলের অনুমতি লাগবে। অনুমতি লাগবে যদি তাকে হার্ডডিস্ক থেকে কিছু পড়তে হয়, বা হার্ডডিস্কে কিছু লিখতে হয়। যদি তার মেমরি কাজে লাগাতে হয় তাহলেও দরখাস্ত দিতে হবে কারনেলকে। আর কাজ করতে গেলে এই প্রয়োজনগুলো তার প্রতিমুহূর্তেই গজিয়ে উঠবে। ভেবে দেখুন, কোনো ভৌত রসদ ব্যবহার না-করে একটা প্রোগ্রাম চলবে কী করে? চলা মানেই তো অন্তত মেমরিতে তোলা মূল প্রোগ্রামটাকে জাগিয়ে রাখা, তার মানেই মেমরির ব্যবহার। তাই প্রতিমুহূর্তে তাকে দরখাস্ত পাঠাতে হচ্ছে কারনেলের কাছে। মুহূর্মুহূ এই অবিরত এবং নিরবচ্ছিন্ন দরখাস্তের স্রোতটারই অন্য নাম 'সিস্টেম কল'।

সাধারনত আইও বা ইনপুট-আউটপুট সংক্রান্ত প্রতিটা কাজই ঘটে কারনেলের মাধ্যমে। কারনেল সেই ব্যবস্থাটা করে যাতে একজন প্রসেসের আইও অন্য প্রসেসের আইও-র ল্যাজ মাড়িয়ে না দেয়। আপনার অ্যাকাউন্টিং সফটওয়ার যে বাইটগুলো পড়েছিল হার্ড ডিস্ক থেকে, ধরুন, আপনার সাবানের দাম, সেটা যাতে আপনার গ্রাফিক্স সফটওয়ারের খুলে রাখা ছবির ফাইলে দেগে না-দেয়। সামান্য দু-চারপিস ইউজার প্রসেসের কখনো সখনো একটু কারনেলের নিয়ন্ত্রণ কাটিয়ে পিছলে যাওয়ার অধিকার থাকে, কখনো সখনো তাদের সরাসরি আইও পোর্টে গিয়ে নিজের কাজ করার অনুমতি থাকে। এক্স-উইনডোজ লিনাক্সে কাজ করে একটা সার্ভার হিশেবে, এক্স উইনডোজ মানে আমরা তো আগেই বলেছি কমান্ডের জায়গায় ছবি দিয়ে তৈরি একটা প্রতিবেশ, যেখানে সব কিছুকে আমরা দেখি বুঝি বদলাই ছবি কাজে লাগিয়ে — এই এক্স উইনডোজ হল এই ধরনের পাঁকাল এবং পিছলে প্রসেসগুলোর মধ্যে প্রমুখ, যাকে প্রায়ই অধিকার দেওয়া হয় সরাসরি, কারনেল এড়িয়ে, আইও পোর্টে কিছু করার। যাইহোক, সে অনেক জটিল কথা।

অপারেটিং সিস্টেম আর ইউজার প্রোগ্রামগুলোর মধ্যে পারস্পরিকতা বা ইন্টারফেস হিশেবে এই সিস্টেম কলের ব্যবস্থা করে দেয় অপারেটিং সিস্টেমই। এক অপারেটিং সিস্টেম থেকে আর একটা অপারেটিং সিস্টেমে সবচেয়ে গুরুত্বপূর্ণ পার্থক্যের একটা হল এই সিস্টেম কলের গঠনের ফারাক। ধরুন, একটা বিশেষ অপারেটিং সিস্টেম ইউনিক্স — তার ফাইল সংক্রান্ত সিস্টেম কলগুলোকে ভাবছি আমরা — তাদের মধ্যে অনেকগুলো রকম আছে। ধরুন, একটা খুব সাধারন কাজকেই ভাবুন — একটা ফাইলকে একটা ডিস্কভূমি থেকে পড়ে আর একটা ডিস্কভূমিতে লেখা হচ্ছে। সচরাচর যাকে আমরা ফাইল কপি করা বলে ডাকি। এইটুকু একটা ছোট্ট কাজেও কিন্তু ব্যবহৃত হচ্ছে অনেকগুলো রকমের সিস্টেম কল। প্রথমে একটা সিস্টেম কল থাকবে ফাইলটাকে চেনানোর। ফাইলটার মধ্যে হার্ডডিস্কের শরীরের কত কত নম্বর সেক্টর থেকে কত কত কোন কোন বাইট তথ্য ভরা আছে, সেটা রয়েছে সেটা লেখা রয়েছে ফাইলসিস্টেমে। পরে আমরা দেখব ফাইলের ডিটেইলস লেখা এই তালিকাকে বলে আইনোড টেবিল।

এই আইনোড টেবিল থেকে ফাইলের খুঁটিনাটি মিলিয়ে হার্ডডিস্ক থেকে ফাইলটা পড়ার জন্যে একটা সিস্টেম কল থাকবে। একটা সিস্টেম কল থাকবে যা ঠিক করবে এবার হার্ডডিস্কের অন্য কোন জায়গায় এই বাইটগুলো নতুন করে লিখে ফেলা হবে, এবং লিখে ফেলবে বাইটগুলো। একটা সিস্টেম কল থাকবে কত গুলো বাইট অব্দি এই এক জায়গা থেকে পড়া আর অন্য জায়গায় লেখার প্রক্রিয়াটা চালিয়ে যেতে হবে। নতুন তৈরি হওয়া ফাইলটাকে ফাইলসিস্টেমের আইনোড তালিকায় আপডেট করে নেওয়ার জন্যেও একটা সিস্টেম কল থাকবে। ইত্যাদি।

আমরা আগে, চার নম্বর দিনে, ইউনিক্স-এর গঠনের ব্যাপারে যে আন্তর্জাতিক পোজিক্স স্ট্যান্ডার্ডের কথা বলেছি, সেই মোতাবেক সবগুলো ফ্লেভারের ইউনিক্স, মানে বেল ল্যাবের ইউনিক্স, সিস্টেম ভি, বিএসডি, গ্নু-লিনাক্স ইত্যাদি — এদের প্রত্যেকেরই সিস্টেম কলের গঠন মোটামুটি একরকম — সামান্য কিছু খুঁটিনাটির তফাত আছে। সিস্টেম কলের গোটা ব্যাপারটাই ভয়ানকভাবে মেশিন-নির্ভর — এখানে সফটওয়ারের ভূমিকা অত্যন্ত কম। এবং প্রায়শই এই সিস্টেম কলগুলো একটা অপারেটিং সিস্টেমে দেওয়া হয় অ্যাসেম্বলি কোডে লিখে। তারপর সি-থেকে তাদের ব্যবহার করার ব্যবস্থা করে দেওয়া হয়। সিস্টেমে এদের রাখা থাকে মূলত কিছু সিস্টেম কল ফাংশনের লাইব্রেরি হিশেবে। মানে সি থেকে কিছু নির্দিষ্ট ফাংশন ব্যবহার করা যায় যারা আসলে অ্যাসেম্বলি কোডে লেখা এই সিস্টেম কলগুলোকে কাজে লাগায়। শুধু সি নয় যদিও, অন্য অন্য কম্পিউটার ভাষা দিয়েও কাজে লাগানো যায় এই সিস্টেম কলগুলোকে।

এখানে একটা ব্যাপার মাথায় রাখুন, একটা এক-প্রসেসর কম্পিউটারের সিপিইউ বা সেন্ট্রাল প্রসেসিং ইউনিট একটা একক মুহূর্তে কেবলমাত্র একটি নির্দেশই পালন করতে পারে। কী করে সময়কে খুব ছোট ছোট টুকরোয় ভেঙে তার এক একটা টুকরোয় এক একটা নির্দেশ পালন করে মোটের উপর অনেকগুলো নির্দেশ একই সঙ্গে পালন করার একটা মায়া তৈরি করে কম্পিউটার তা আমরা মাল্টিপ্লেক্সিং-নিয়ে আলোচনায় আগেই বলেছি। তাই, এবার কোনো একটা নির্দিষ্ট ইউজার প্রোগ্রামের যদি কোনো সিস্টেম রসদ ব্যবহারের প্রয়োজন পড়ে, যেমন ডিস্কে লেখা কোনো ফাইল থেকে পড়া ইত্যাদি, সে তখন সিস্টেম কল নামক দরখাস্ত পাঠায় কারনেলের কাছে এবং তার নিজের কাজের নিয়ন্ত্রণ সেই মুহূর্তে ছেড়ে দেয় অপারেটিং সিস্টেমের হাতে। অপারেটিং সিস্টেম তখন তার প্রয়োজনের বিভিন্ন উপাদান আর মাত্রাগুলোকে বিচার করে সেইমত এই প্রোগ্রামের হয়ে ভৌত রসদটুকু ব্যবহার করে। তারপর ফের প্রোগ্রামের নিয়ন্ত্রণ ফেরত দেয় সেই ইউজার প্রোগ্রামের নিজের হাতে। সিস্টেম কলের কাজ শেষ হয়।

১১।। আর একটু খোসা চর্চা

বঙ্কিম বা রামকৃষ্ণ বা রাজশেখরের বালখিল্য পর্বতমুনি যতই খোসা-অন্তপ্রাণ হতে বারণ করুন, আপাতত আমাদের একটু খোসা-চর্চা করতেই হবে। আগেই বলেছি, এই খোসা বা শেলই সেই মারাত্মক মিডলম্যান, আমাদের কমান্ড ইন্টারপ্রিটার — আমাদের কমান্ড যে ইন্টারপ্রিট করে পৌঁছে দেয় কারনেলের কাছে — এবং কারনেল তখন আমাদের চালানো প্রোগ্রামগুলোর চাহিদা এবং প্রয়োজন মোতাবেক রসদ সরবরাহ করে। তার মানে, আর একভাবে দেখলে, শেল আমাদের কাছ থেকে কারনেলকে লুকিয়েও রাখে, অপারেটিং সিস্টেমের খুঁটিনাটি জটিলতাগুলোকে গোপনে জনান্তিকে অগোচরে রেখে দেয়। আমাদের আর মাথা ঘামাতে হয়না। এক নম্বর দিনে আমরা যে প্রতীকী বা সিম্বলিক ভারচুয়াল ভূমির কথা বলেছিলাম, ভৌত উপাদানগুলোর ভিত্তির উপর দাঁড়িয়ে কী করে স্তরে স্তরে সেটাকে বুনে তোলা হয় — এখন সেটা একটু একটু করে বুঝতে পারছেন?

আমাদের খোসা বা শেলও একটা ইউজার প্রসেস। সে আমাদের কিবোর্ডে চাবি টেপার, ইঁদুরের পেট কচলে দেওয়ার নানা ইভেন্টের বা ঘটনার জন্যে অপেক্ষা করে। এই অপেক্ষা করা মানে একমনে আইও পোর্টগুলো পড়তে থাকা — কিছু সেখানে ঘটল কিনা। এই ক্রিয়াটা গোটাটাই ঘটে কারনেলের মধ্যে দিয়ে। যেই কারনেল দেখে আপনি কোনো একটা কমান্ড লিখে এন্টার মারলেন — সেই এন্টার মারার আগের গোটা লাইনের টেক্সটা সে পাঠিয়ে দিল শেলের কাছে। শেল তার জানা কমান্ডগুলোর সঙ্গে এবার মিলিয়ে মিলিয়ে দেখে নিল আপনি যা যা টাইপ করেছেন সেই চাবিগুলোকে। ধরুন আমরা টাইপ করেছি '`ls`', তারপর এন্টার মেরেছি। কারনেল এন্টারের আগেরটুকু মানে 'ls' অংশটাকে পাঠিয়ে দিয়েছে শেলের কাছে। শেল এবার মিলিয়ে দেখতে শুরু করে দিয়েছে, এই 'ls' আকারের কোনো কমান্ড আছে কিনা যাকে সে চেনে। পরে আমরা দেখব, শেলের এই খোঁজাটা যে গোটা মেশিন মানে হার্ড ডিস্ক জুড়ে

ঘটে তা কিন্তু নয়। নির্দিষ্ট কিছু ডিরেক্টরিতে খুঁজে দেখে শেল। যে ডিরেক্টরিগুলোর নাম তার পাথ বা পথনির্দেশ-এ দেওয়া আছে। শেলের কাছে দেওয়া এই পথনির্দেশে কী কী ডিরেক্টরি রয়েছে সেটা আমরা জানতে পারি 'echo' কমান্ড ব্যবহার করে। আর, পথনির্দেশ বলে সিস্টেমে যেটা থাকে, তার টেকনিকাল পরিচিতি হল একটা '$PATH' নামে একটা শেল ভ্যারিয়েবল। অনেক কিছুই এখানে বোঝার আছে। নয় আর দশ নম্বর দিনে আসব এই প্রসঙ্গে। আমরা কমান্ড দেব, 'echo $PATH'। এরপর এন্টার মারলেই শেল আমাদের দেখাবে, কী কী ডিরেক্টরিতে তার খুঁজে দেখার কথা। যারা উইন্ডোজ প্রতিবেশে 'autoexec.bat' ফাইলটাকে কখনো কাজে লাগিয়েছেন তারা দেখবেন এর মধ্যে দেওয়া থাকে উইন্ডোজ ওএস-এর পথনির্দেশটা। উইন্ডোজ প্রতিবেশে পথনির্দেশ জানতে হলে ডস-প্রম্পটে কমান্ড দিতে হয়, 'path'।

গ্নু-লিনাক্সে পথনির্দেশের কাঠামোটার সঙ্গে একটু পরিচিত হয়ে নিন এবার। আমার গ্নু-লিনাক্স সিস্টেমে আমি সাধারণ ইউজার হয়ে, মানে, 'dd' হয়ে, 'echo $PATH' কমান্ড দেওয়ায় স্ক্রিনে ফুটে উঠল এই রকম কিছু —

```
/home/dd/bin:/usr/local/bin:/usr/bin:/usr/X11R6/bin:/bin:/usr/games:
```

আবার 'su' করে রুট পাসওয়ার্ড দিয়ে রুট হয়ে ওই একই কমান্ড দেওয়ায় ফুটে উঠল —

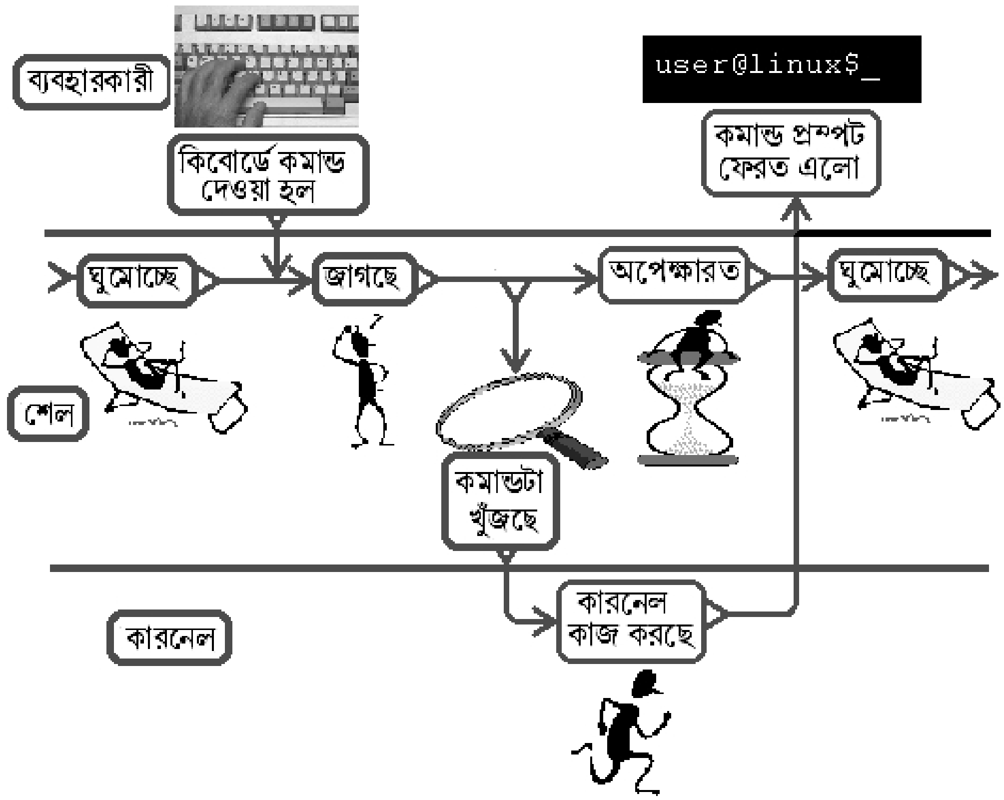
```
/sbin:/usr/sbin:/usr/local/sbin:/root/bin:/usr/local/bin:/usr/bin:/usr/X11
R6/bin:/bin:/usr/games
```

বাইনারি ফাইল মানে, আগেই বলেছি, যে ফাইলগুলোর নাম লিখে এন্টার মারলে নিজে নিজেই চলে। যেমন, 'ls' একটা বাইনারি, কারণ, 'ls' লিখে এন্টার মারলে সে নিজে নিজেই কাজ করে এবং ফাইল আর ডিরেক্টরির তালিকা ফুটিয়ে তোলে। উইনডোজ প্রতিবেশে '*.exe' ইত্যাদি এক্সিকিউটেবল ফাইলগুলো যা করে। 'dd'-র হোম ডিরেক্টরি '/home/dd'। সেখানের বাইনারি ডিরেক্টরি হল '/home/dd/bin', 'dd'-র ব্যবহারযোগ্য বাইনারি ফাইলগুলো যেখানে থাকে, সেটা রয়েছে 'dd'-র পথনির্দেশে। আর রুটের হোম '/root', সেখানের বিন ডিরেক্টরি '/root/bin', সেটা আছে রুটের পথনির্দেশে। এর বাইরে রুটের পথনির্দেশে আছে তিনটে আলাদা সিস্টেম বাইনারির মানে সিস্টেম প্রোগ্রামের ডিরেক্টরি, '/sbin:/usr/sbin:/usr/local/sbin'। এগুলো 'dd'-র নেই, কারণ ওই কাজগুলো করার, ওই প্রোগ্রামগুলো চালানোর অনুমতি সাধারণ ইউজারের থাকেনা। আর সেই ':' চিহ্নটা এখানেও ব্যবহার হচ্ছে যতিচিহ্ন হিশেবে।

শেলের কথায় ফেরত আসা যাক। আমি 'ls' লিখে এন্টার মেরেছি। শেল খোঁজা শুরু করেছে সেই ডিরেক্টরিগুলোয় এই মাত্র 'dd'-র 'echo $PATH' কমান্ড থেকে যে পথনির্দেশ পেলাম। এই পথগুলোর মধ্যে যদি এবার শেল 'ls' নামে কোনো বাইনারি ফাইল পেয়ে যায়, সেই 'ls' ফাইলটাকে চালিয়ে দেয়। এবং তার আউটপুট মানে ফলাফলটাকে পৌঁছে দেয় স্ক্রিনে। যাতে 'ls' কমান্ড চালানো ইউজার সেটা দেখতে পায়। এই ক্ষেত্রে, এই 'ls' নামের বাইনারিটাকে শেল পেয়ে যাবে '/bin' ডিরেক্টরিতে, যা পথেই আছে। পাওয়া মাত্র শেল কারনেলকে আবেদন জানায় একটা '/bin/ls' প্রসেস চালু করার। মানে, মোদ্দা কথায়, 'ls' প্রোগ্রামটা চালানোর। এই আবেদনটা জানাতে হল কেন? নিজেই ভাবুন তো? প্রতিমুহূর্তে নিজেকে জেরা করতে করতে এগোন। এখানে আবেদনটা জানাতেই হবে। 'ls' প্রোগ্রামটা চলা মানেই তো ডিস্ক থেকে ডিরেক্টরির আর ফাইলের তালিকা বানানো, তার মানেই ডিস্ক পড়তে হবে, তার মানেই ডিস্ক নামক ভৌত রসদের সঙ্গে কথা বলার, আর সেই কথা বলার অধিকার একমাত্র কারনেলের আছে। এবার এই '/bin/ls' প্রসেসটা একটা লিস্ট বা তালিকা তৈরি করে যে ডিরেক্টরি থেকে 'ls' কমান্ডটা দেওয়া হয়েছিল, সেই ডিরেক্টরিতে মোট উপস্থিত ডিরেক্টরি এবং ফাইলের। এই '/bin/ls' চালিয়ে তালিকা বানিয়ে সেই তালিকাটা স্ক্রিনে পাঠানোর দায়িত্বটা কারনেলকে দিয়ে শেল ঘুমোতে যায়। কেন, আবার কারনেল কেন? যদি না-বুঝতে পারেন, একটু গান শুনুন, পরে পড়বেন। যতটুকু সময় ধরে '/bin/ls' কাজ করে ততক্ষণ ঘুমোয় শেল। এই তালিকাটা এবার কারনেল পৌঁছে দেয় ইউজারের কাছে। কনসোল একটা ভৌত রসদ, তাই, এটা আর শেলের এলাকা নয়, কারনেলের কাছে সিস্টেম কল নামক দরখাস্ত করাটুকুই তার কাজ।

'/bin/ls' প্রক্রিয়াটা শেষ হয়, শেষ হওয়ার সংবাদ কারনেলের কাছে পৌঁছে দেয় '/bin/ls' নিজেই। এটা প্রোগ্রামেরই অংশ — একজিট (exit) নামক সিগনাল কারনেলের কাছে পৌঁছে দেওয়া। এই সিগনাল হাতে পেয়ে কারনেল এবার শেলকে ফের জাগিয়ে তোলে। আবার জেগে থাকতে বলে। জেগে থাকা মানে অতন্দ্র প্রহরা — কিবোর্ডে কিছু

নতুন কমান্ড এল কিনা। বা মাউসে। কোনো নতুন ঘটনা। শেল কারনেলের আদেশ মোতাবেক জেগে ওঠে, আবার স্ক্রিনে ফুটে ওঠে কমান্ড প্রম্পট। যার মানে শেল ফের অপেক্ষা করছে নতুন কমান্ড টাইপ করে এন্টার মারার। আমরা স্ক্রিনে দেখলাম 'ls' টাইপ করার পর এন্টার মারা মাত্রই কমান্ড প্রম্পট উঠে গিয়ে স্ক্রিনে ফুটে উঠল ফাইল আর ডিরেক্টরির তালিকাটা। তারপর ফের ফেরত এল কমান্ড প্রম্পট। মানে 'ls' আদেশটা পালন করা হয়ে গেছে।



এবার যদি কোনো একটা স্টেপে কিছু ঘেঁটে যায়? বা অকারণ দেরি হয়? ধরুন, যে ডিরেক্টরিতে 'ls' মেরেছেন সেটা বেধড়ক লম্বা, কয়েক কোটি ফাইল আর ডিরেক্টরি আছে সেখানে, এবং আপনার প্রসেসরও একটু মাঠো, মার্কিন কায়দায় বললে 'স্টেট-অফ-দি-আর্ট' নয় মোটেই, সেক্ষেত্রে কমান্ড প্রম্পট তো ফিরে আসছে না, দেরি করছে — আপনি এখন কী করবেন? আগেই বলেছি, কন্ট্রোল আর অল্ট সুইচদুটোকে টিপে রেখে এফ-ওয়ান থেকে এফ-সিক্স অব্দি যেকোনো একটা এফ-সুইচ টিপলেই আপনি সেই নম্বর ভারচুয়াল কনসোলে চলে যাবেন। ম্যানড্রেক বা সুজে ডিস্ট্রিবিউশনে এই কাজটা উইন্ডোজ চাবিটা টিপেই করা যায়, মানে গলে যেতে থাকা বা উড়ে যেতে থাকা উইনডোজের লোগো আঁকা সুইচ-দুটো। ডানদিকেরটা টিপলে পরের ভারচুয়াল কনসোলে আর বাঁদিকেরটা টিপলে আগের ভারচুয়াল কনসোলে যাওয়া যায়। অন্য কনসোলে গিয়ে অন্য কোনো একটা কাজ দিন মেশিনকে।

উইন্ডোজ থেকে যারা এসেছেন তাদের কাছে এটা সত্যিই একটা নতুন অভিজ্ঞতা — সত্যিকারের মাল্টিটাস্কিং কাকে বলে দেখুন। উইন্ডোজ নিজেকে মাল্টিটাস্কিং ওএস বলেই দাবি করে। কিন্তু সেই মাল্টিটাস্কিং-টা উপর থেকে চাপানো। সেটা বোঝা যায় গ্নু-লিনাক্স করতে গেলে। এমনও হয়েছে আমার সিস্টেমে যে, একটা কনসোলে প্রম্পট আর সেদিনের মত ফেরতই আসেনি। অথচ আমি অন্য কনসোল থেকে এক্স-উইন্ডোজ অব্দি চালিয়েছি। সেখানে অন্যান্য কাজ করে গেছি। পরের বার যখন বুট করেছি তখন সেই হারিয়ে যাওয়া কনসোলটাকে ফেরত পেয়েছি। আমরা যারা উইন্ডোজে কাজ করেছি অনেকগুলো বছর, তাদের বেলায় এই অভিজ্ঞতাটা বেশ ব্যাপক। গ্নু-লিনাক্স তথা যে কোনো ইউনিক্সের শেল নামক বস্তুটা ভীষণ রিজনেবল। আমি এই রিজন ঘেঁটে যেতে দেখিনি কখনো। উইনডোজে বেশিদিন একটানা কাজ করার পরই নিজের বুদ্ধি এবং যুক্তিবোধ দুটোর উপরই ভরসা কমে যেতে থাকে, কেন কোনটা কখন কী হয় — এটা প্রায় জীবনরহস্যের মতই অবোধ্য লাগে।

এখানে আমার নিজের একটা মজা পাওয়ার গল্প করে নিই। আমার মেশিনে একটা ব্যাক-আপের কাজ চলছিল। আমাদের কলকাতা লাগ-এর ইন্দ্র আর তথাগত এসেছিল আমার একটা ঘেঁটে যাওয়া স্কাসি কনফিগারেশন ঠিক করার চেষ্টার নাম করে আড্ডা মারতে আর কাবাব ইত্যাদি খেতে। হঠাৎ কারেন্ট চলে গেল। ইউপিএস-টা চলার কথা পনেরো মিনিট, বারো সাড়ে বারোর বেশি চলেনা। অথচ ব্যাক আপে তার চেয়ে বেশি সময় লাগবে। কিছুটা বিদ্যুৎ বাঁচানোর প্ল্যানে তথাগত কিবোর্ডে টাইপ করল ‘পাওয়ারঅফ’ (poweroff) এবং এন্টার মারল, স্ক্রিনে তখন লাইনের পর লাইন ব্যাকআপশীল ফাইলের তালিকা হুড়হুড় করে নেমে যাচ্ছে, তারপর মনিটরটা বাইরে থেকে অফ করে দিল। মানে পাওয়ারঅফ কমান্ডটা সত্যিই মেশিনের কাছে পৌঁছল কিনা আমরা তা দেখতে পেলাম না। এই পাওয়ারঅফ কমান্ডটা তার নাম অনুযায়ী সত্যিই পাওয়ারকে অফ করে। একসময় মেশিন থেমে গেল। এবং পরে কারেন্ট এলে বুট করে দেখলাম সত্যিই ব্যাকআপের কাজ শেষ করে মেশিন অফ হয়েছিল। মানে আগের কমান্ডটা শেষ হওয়ার পরে প্রম্পট ফেরত এসেছিল, মানে আসেনি কারণ কনসোল তখন অফ ছিল, এবং সেখানে চালু হয়েছিল পাওয়ারঅফ কমান্ডটা এবং তার পরের এন্টার। যে অংশটা ততক্ষণ মেশিনের বাফারে ছিল। এটা এমন কিছু ঘটনা না, কিন্তু আমি ভারি আমোদ পেয়েছিলাম। গ্নু-লিনাক্সে এটা করা চলে। যতক্ষণ খুশি যতবড় খুশি কমান্ড টাইপ করে চলা যায়। আমি স্ক্রিনে হয়ত দেখতে পাচ্ছিনা, কিন্তু তাতে কাজের কোনো অসুবিধা হয়না। আমরা আগেই উল্লেখ করেছি এটাই ইউনিক্স-এর ফুল-ডুপ্লেক্স সিস্টেমের সুবিধে। কিন্তু এখন না এসব, পরে আসবে, আমরা যখন আরো বড় করে ইনপুট আউটপুটের আলোচনায় আসব।

বুট প্রক্রিয়ার প্রাথমিক পরিচয়টা আমরা পেলাম, এবং শেলের ক্রিয়াপদ্ধতির। পরের দিন আমরা হার্ডডিস্কের পার্টিশনিং এবং ফাইলসিস্টেমকে নিয়ে এই বুটপ্রক্রিয়াটাকে আরো একটু বিশদে বুঝব। এবং কিছু কমান্ডকেও। শেষ অব্দি সরাসরি লিনাক্সে আমরা পৌঁছলাম, এখনো বিশ্বাস হচ্ছে না, যেরকম ঘুরপথের পর ঘুরপথ চলছিল। পাঠমালাটা পড়া এবং তার কমান্ডগুলোকে করে করে দেখা কিন্তু চলতেই থাকবে। নইলে মাথায় বসবে না। পড়ার জায়গাগুলো মোটামুটি একই আছে। সুমিতাভ দাস, কারনিঘ্যান পাইক, স্ট্যানফিল্ড আর স্মিথের ‘লিনাক্স — সিস্টেম অ্যাডমিনস্ট্রেশন’, তার সঙ্গে ইন্ট্রোডাকশন টু লিনাক্স বলে মাখটেল্ট গারেলস-এর একটা বই, জিএলটির সিডিতে আছে, একটু ওল্টানো যেতে পারে, মানে, ইয়ে, পেজ আপ ডাউন করা যেতে পারে।

glt-mad@ilug-cal.org

MADHYAMGRAM