حسین سیلانی
طراح، توسعه دهنده وسیستم ادمین

Book 14\100

Linux Professional Institute Certification Study Guide

LPIC-2
(201-450)
(202-450)

LPIC-2 REFERENCE
Study Guide 1st Edition

کتاب
مرجع
LPIC1
لینوکس

معرفی کوتاه کتاب

سطح دوم از سری گواهینامه‌های حرفه‌ای لینوکس تحت نظر سازمان LPI (Linux Professional Institute) است. این گواهینامه تخصصی برای کسانی طراحی شده است که پیش از این مهارت‌های پایه‌ای لینوکس را بدست آورده‌اند (با گذراندن LPIC-1) و اکنون می‌خواهند دانش خود را در زمینه اداره شبکه‌ها، خدمات شبکه، امنیت و نگهداری سیستم‌ها تقویت کنند. در واقع، LPIC-2 مرز بین مدیر سیستم محلی و متخصص سیستم‌های درون شبکه را نشان می‌دهد.باید بتواند شبکه‌ای ترکیبی از لینوکس و مایکروسافت را برنامه‌ریزی، پیاده‌سازی، حفظ کند و در شرایط بحرانی عیب‌یابی نماید. او باید بتواند سرویس‌های مهم شبکه را راه‌اندازی کند — مانند سرور داخلی با سرویس‌هایی نظیر DNS، NFS، Samba و DHCP؛ نقطه دسترسی به اینترنت با فایروال، VPN، SSH، پروکسی وب و سرویس ایمیل؛ و همچنین سرورهای خارجی شامل وب‌سرور با پروکسی معکوس و سرور FTP. علاوه بر آن، داوطلب باید توانایی نظارت بر دستیاران را داشته باشد

کتاب مرجع کامل مدرک بین‌المللی

LPIC-2

(201-450, 202-450)

Linux Professional Institute Certification Study Guide

حسین سیلانی

ویرایش اول

۱۴۰۴

سرشناسه	سیلانی، حسین، ۱۳۶۰،
عنوان و نام پدیدآور	Lpic2
	حسین سیلانی؛ ویراستاری حسین سیلانی.
مشخصات نشر	تهران: کیان دانش، ۱۴۰۴.
مشخصات ظاهری	600 ص.
شابک	۹۷۸،۶۲۲،۴۰۱،۰۵۵،۶
وضعیت فهرست نویسی	فیپا
موضوع	سیستم عامل لینوکس
	Operating System
شناسه افزوده	سیلانی، حسین، ۱۳۶۰ ،
رده بندی کنگره	۷۶/۷۶QA
رده بندی دیویی	۴۳۲/۰۰۵
شماره کتابشناسی ملی	۹۶۷۳۵۹۳
اطلاعات رکورد کتابشناسی	فیپا

نشر کیان دانش: ناشر کتاب‌های دانشگاهی

عنوان: مفاهیم سیستم عامل

تالیف: حسین سیلانی

صفحه آرایی و ویراستاری: حسین سیلانی

چاپ و صحافی: موسسه مهراد

طراحی جلد: حسین سیلانی

چاپ اول: ۱۴۰۴

شمارگان: ۱۰۰ جلد

شابک: ۶،۰۵۵،۴۰۱،۶۲۲،۶۲۹

خیابان ۱۲ فروردین، خیابان شهدای ژاندارمری، شماره ۱۲۶، طبقه چهارم

سخنی از نویسنده

با سلام و احترام به دوست گرامی

از اینکه این کتاب را انتخاب کرده‌اید و در این مسیر همراه من هستید صمیمانه سپاسگزارم. امیدوارم این اثر برای علاقه‌مندان به حوزه لینوکس مفید و اثربخش باشد. همچنین از شما درخواست دارم که اگر نواقص یا کاستی‌هایی در کتاب مشاهده کردید آن‌ها را نادیده نگیرید و با ارائه نظرات و پیشنهادات سازنده‌ی خود، به بهبود کیفیت این اثر کمک کنید.

شایان ذکر است که این کتاب یکی از صدها کتاب لینوکسی من است که بسیاری از آن‌ها در حال ترجمه یا نگارش هستند. این مجموعه در راستای پروژه‌های متن‌باز شخصی‌ام و با هدف طراحی و توسعه توزیع‌های لینوکس نوشته شده است. امیدوارم این تلاش‌ها گامی هرچند کوچک در جهت گسترش دانش و استفاده از سیستم‌عامل‌های متن‌باز باشد.

- https://predator.os.ir/
- https://little.Psycho.ir
- https://emperor.os.ir/
- https://hubuntu.ir

تصمیم دارم محتوای آموزشی فارسی در حوزه‌های لینوکس و نرم‌افزارهای متن‌باز را در قالب‌های متنوعی مانند کتابچه، کتاب، فلش‌کارت، فایل‌های صوتی و ویدئوهای آموزشی تولید کرده و از طریق یک سایت در دسترس عموم قرار دهم. هدف من از این کار، ایجاد یکی از بزرگترین و جامع‌ترین مراجع آموزشی فارسی در این زمینه است.برای دسترسی به سایت و اطلاع از آخرین اخبار و محتواهای آموزشی، می‌توانید از طریق راه‌های ارتباطی زیر با من در تماس باشید:

- **آدرس سایت:** learninghive.ir
- **کانال تلگرام:** @Linuxtnt
- **آیدی تلگرام:** @seilany
- **ایمیل:** h.seilany@gmail.com
- **گیت‌هاب:** hosseinseilani

منتظر حضور و همراهی شما هستم!

تقدیم به تو

یگانه فرزانه دلم

فهرست مطالب

مقدمه

پس از موفقیت و استقبال گسترده از کتاب مرجع دستورات لینوکسی تحت عنوان **۱۰۰۱ دستور لینوکس**، این بار با افتخار کتابی جامع و **کاربردی** با عنوان **کتاب مرجع کامل مدرک بین‌المللی LPIC-2** را در اختیار شما قرار می‌دهیم. این کتاب نه تنها آموزش گام‌به‌گام و کامل مباحث اصلی لینوکس را ارائه می‌دهد، بلکه با بهره‌گیری از منابع بین‌المللی معتبر و تجربه‌های عملی، مطالعه‌ای منسجم، عمیق و **کاربردی** را برای شما فراهم می‌آورد.

کتاب حاضر تمامی مفاهیم و سرفصل‌های مورد نیاز برای موفقیت در آزمون‌های **LPIC-2**را پوشش می‌دهد، به‌گونه‌ای که نیاز به مراجعه به کتاب‌ها و منابع دیگر به حداقل می‌رسد. این کتاب شامل محتوای جامع و به‌روز شده از منابع معتبر زیر است:

- **LPIC-2 Objectives V4.5 – Linux Professional Institute**
- **LPIC-2: Study Guide: Exam 201 and Exam 202 (Sample, 2nd Edition)**
- **LPIC-2 Cert Guide: (201-450 and 202-450 exams) (Pearson IT Certification)**
- **CompTIA LPIC-2 201, and LPIC-2 202 (Sample)**
- **GNU/Linux Advanced Administration**

با مطالعه این کتاب، شما نه تنها برای آزمون‌های LPIC-2آماده خواهید شد، بلکه دانش تئوری و مهارت عملی شما در لینوکس به سطحی حرفه‌ای و قابل اعتماد ارتقا می‌یابد. این مجموعه می‌تواند هم به‌عنوان یک راهنمای آموزشی گام‌به‌گام و هم به‌عنوان یک مرجع سریع و کاربردی در کنار شما باشد و نیازهای شما را در محیط‌های آموزشی، کاری و حرفه‌ای به طور کامل پوشش دهد.

مخاطبین هدف این دوره چه کسانی هستند؟

این دوره به‌گونه‌ای طراحی شده است که طیف وسیعی از علاقه‌مندان و متخصصان فعال در حوزه فناوری اطلاعات را پوشش دهد. محتوای آموزشی از مبانی پایه شروع شده و تا موضوعات پیشرفته و **کاربردی** در محیط‌های enterprise ادامه می‌یابد، بنابراین برای افراد تازه‌کار و همچنین حرفه‌ای‌هایی که قصد تثبیت دانش و پر کردن خلاءهای علمی خود را دارند، کاملاً مناسب است.

- **تازه‌واردان به دنیای لینوکس:** افرادی که هیچ پیش‌زمینه‌ای ندارند و می‌خواهند **یادگیری لینوکس را از پایه‌ترین مفاهیم** به صورت اصولی و **کاربردی** آغاز کنند.
- **جامعه توسعه‌دهندگان و مهندسین نرم‌افزار: برنامه‌نویسان، توسعه‌دهندگان وب و مهندسین نرم‌افزار** که برای توسعه، استقرار (Deploy) و عیب‌یابی (Debug) برنامه‌های خود نیاز به تسلط بر محیط سرورهای لینوکس و خط فرمان (Terminal) دارند.
- **متخصصان DevOps و ابر (Cloud): کارشناسان DevOps، مهندسین (SRE) Site Reliability و متخصصان Cloud** که اسکریپت‌نویسی، اتوماسیون، مدیریت زیرساخت به‌عنوان کد (IaC) و کار با سرویس‌های ابری مانند AWS، Azure و GCP در هسته مرکزی فعالیت آن‌ها قرار دارد.
- **متخصصان و مدیران شبکه: مدیران شبکه، مهندسین زیرساخت IT، کارشناسان NOC و آنالیزورهای شبکه** که برای مدیریت روترها، سوئیچ‌ها، فایروال‌ها و سرویس‌های تحت شبکه) مانند (DNS, DHCP, Proxy به لینوکس به‌عنوان یک ابزار حیاتی نیاز دارند.
- **حرفه‌ای‌های امنیت سایبری: کارشناسان تست نفوذ(Penetration Testers)، هکرهای قانونمند(Ethical Hackers)، متخصصان امنیت اطلاعات (SOC Analysts) و پاسخ به حوادث (Incident Responders)** که برای تحلیل تهدیدات، بررسی لاگ‌ها و استفاده از ابزارهای امنیتی، نیازمند تسلط عمیق بر سیستم عامل لینوکس هستند.
- **تیم‌های پشتیبانی فنی و عملیات: تکنسین‌های پشتیبانی(Help Desk)، متخصصین پشتیبانی سرور و اپراتورهای دیتاسنتر** که وظیفه عیب‌یابی، مانیتورینگ سلامت سرویس‌ها و انجام عملیات اولیه را بر عهده دارند.
- **مشاوران و مدیران فناوری اطلاعات: مشاوران IT و شبکه، مدیران فنی و مدیران ارشد فناوری اطلاعات (CTOs)** که برای تصمیم‌گیری‌های استراتژیک، انتخاب پلتفرم‌های مناسب و درک قابلیت‌های سیستم‌عامل لینوکس نیاز به دانش کافی دارند.
- **دانشجویان و پژوهشگران: دانشجویان مقاطع مختلف در رشته‌های کامپیوتر، فناوری اطلاعات، مهندسی نرم‌افزار و شبکه** که می‌خواهند دانش آکادمیک خود را با **مهارت‌های عملی و مورد نیاز بازار کار** تکمیل کنند.

فصل یکم

فصل ۱

تاریخچه سیستم‌عامل لینوکس

برگرفته از خانواده سیستم‌عامل‌های **یونیکس** است. برخلاف بسیاری از سیستم‌عامل‌های انحصاری، لینوکس تحت مالکیت یا نظارت یک شرکت یا سازمان واحد نیست؛ بلکه حاصل همکاری گسترده میان شرکت‌های تجاری، سازمان‌های غیرانتفاعی و هزاران توسعه‌دهنده مستقل در سراسر جهان است. این همکاری جمعی باعث شده لینوکس به یکی از انعطاف‌پذیرترین و قابل‌اعتمادترین سیستم‌عامل‌های موجود تبدیل شود.

تاریخچه سیستم‌عامل یونیکس

سرآغاز داستان لینوکس را باید در **یونیکس (UNIX)** جستجو کرد. نخستین نسخه از یونیکس در سال **۱۹۶۹** توسط **کن تامپسون**، **دنیس ریچی** و همکارانشان در آزمایشگاه‌های بل (Bell Labs) توسعه یافت. (جالب است بدانید که همان سال، **لینوس تورووالدز**، خالق لینوکس، به دنیا آمد.در ابتدا، یونیکس به زبان **اسمبلی** برای رایانه‌های مینی (Mini Computers) نوشته شد. اما در اوایل دهه ۷۰، **دنیس ریچی** زبان برنامه‌نویسی **C**را معرفی کرد که یک زبان سطح بالا و قابل‌حمل بود. این تحول باعث شد که بخش‌های زیادی از کرنل یونیکس به زبان C بازنویسی شوند و این امر قابلیت **انتقال‌پذیری (Portability)** سیستم‌عامل را به‌طور چشمگیری افزایش داد.

از یونیکس تا لینوکس — روایتی از تولد یک جنبش نرم‌افزاری آزاد

آغاز ماجرا: تولد یونیکس ۱۹۶۹

در اواخر دهه ۱۹۶۰، در آزمایشگاه‌های **بل(Bell Labs)** ، سه برنامه‌نویس برجسته به نام‌های **کن تامپسون**، **دنیس ریچی** و **داگلاس مک‌ایلروی** در حال کار بر روی پروژه‌های بزرگ و پیچیده رایانه‌ای بودند. نتیجه تلاش‌های آنان در سال **۱۹۶۹** به تولد سیستم‌عامل **یونیکس (UNIX)** انجامید. جالب است بدانید که همان سال، در سوی دیگری از جهان، **لینوس تورووالدز**، کسی که بعدها لینوکس را خلق کرد، به دنیا آمد. نسخه اولیه یونیکس به زبان **اسمبلی** نوشته شد و برای اجرا روی رایانه‌های **PDP-11** طراحی شده بود. این سیستم‌عامل به دلیل سادگی، کارایی و قابلیت چندکاربره بودن، خیلی زود توجه محققان و دانشگاه‌ها را به خود جلب کرد.

زبان C و تحولی بزرگ ۱۹۷۳

در سال **۱۹۷۳**، **دنیس ریچی** زبان **C**را طراحی کرد. این زبان سطح بالا به اندازه کافی قدرتمند بود که بتواند هسته سیستم‌عامل را پیاده سازی کند و در عین حال، قابلیت انتقال بین سخت‌افزارهای مختلف را فراهم آورد. نسخه چهارم یونیکس، نخستین نسخه‌ای بود که تمام هسته آن به زبان C نوشته شد؛ اتفاقی که انقلاب بزرگی در دنیای نرم‌افزار ایجاد کرد.

گسترش در دانشگاه‌ها و شکل‌گیری شاخه‌ها

کد منبع یونیکس همراه با مستندات آن به دانشگاه‌ها ارسال شد. این اقدام باعث شد هزاران دانشجو و محقق بتوانند ساختار و کد سیستم‌عامل را مطالعه، تغییر و بهبود دهند. تا اواخر دهه ۷۰، یونیکس در بیش از **۱۲۰ دانشگاه** و **۵۰۰ مرکز تحقیقاتی** در سراسر جهان نصب شده بود. در این زمان، یونیکس به دو شاخه اصلی تقسیم شد:

۱. **BSD (Berkeley Software Distribution)**توسعه‌یافته در دانشگاه کالیفرنیا، برکلی، با تمرکز بر قابلیت‌های شبکه و ابزارهای پیشرفته.

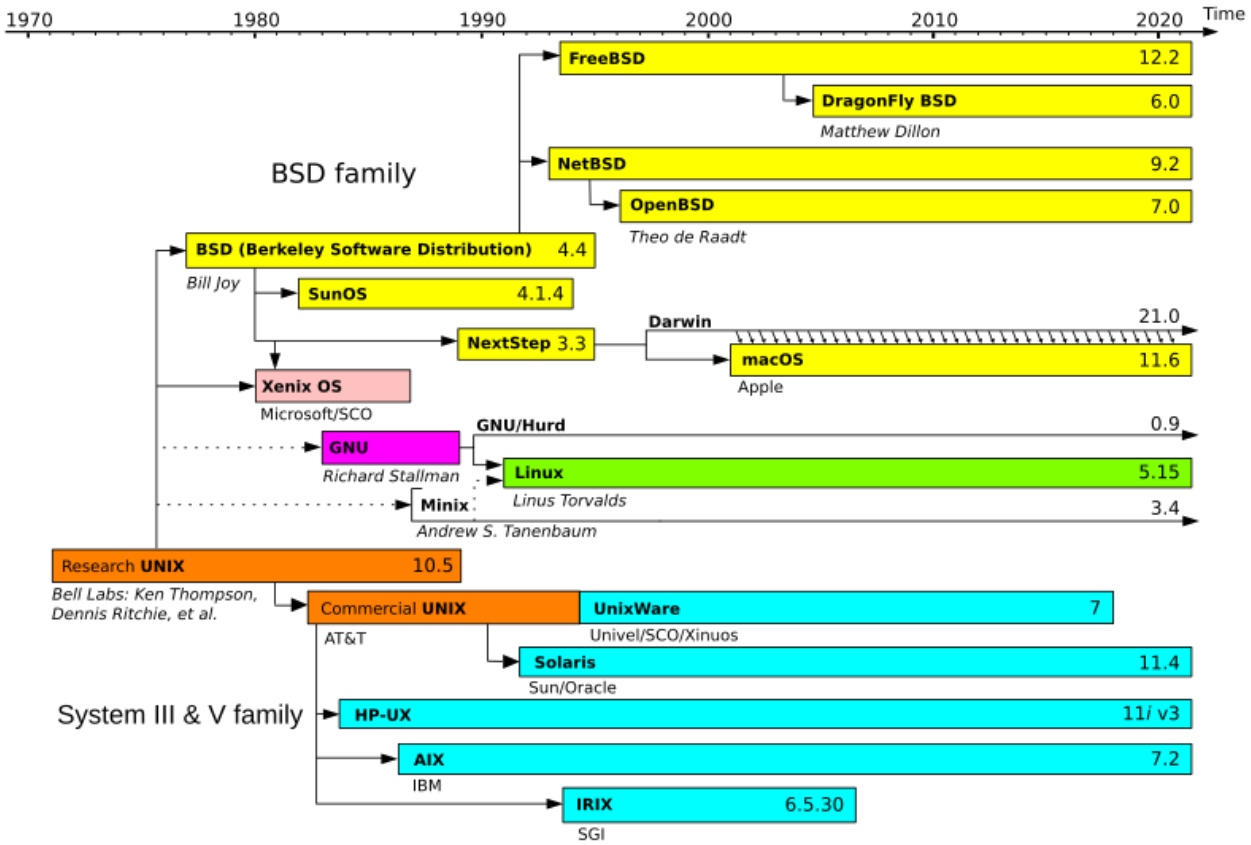
۲. **System V** توسعه‌یافته توسط AT&T ، با ساختار سازمانی و استانداردسازی گسترده.

ظهور مینیکس — پلی میان آموزش و عمل ۱۹۸۷

دهه ۸۰ با رشد سریع رایانه‌های شخصی همراه بود، اما بیشتر سیستم‌عامل‌ها یا انحصاری و گران بودند (مثل محصولات مایکروسافت) یا پیچیده و گران‌قیمت (مثل یونیکس). در این میان، **اندرو تاننباوم**، استاد علوم رایانه در هلند، سیستم‌عامل **مینیکس (MINIX)** را طراحی کرد. مینیکس یک نسخه کوچک و آموزشی از یونیکس بود که همراه با کتاب تاننباوم منتشر می‌شد. این کتاب حاوی توضیحات کامل و حدود **۱۲ هزار خط کد** به زبان C و اسمبلی بود و به دانشجویان اجازه می‌داد منطق داخلی یک سیستم‌عامل واقعی را درک کنند. مینیکس متن‌باز بود (هرچند با محدودیت‌های مجوزی اولیه) و همین باعث شد افراد زیادی از جمله **لینوس تورووالدز** با آن آشنا شوند.

ریچارد استالمن و پروژه گنو) ۱۹۸۳

در سال **۱۹۸۳**، **ریچارد ا ستالمن**، برنامه‌نویس آمریکایی و فعال حوزه آزادی نرم‌افزار، پروژه‌ای را به نام **GNU** آغاز کرد. هدف او ساخت یک سیستم‌عامل کاملاً آزاد و سازگار با یونیکس بود. پروژه گنو ابزارها، کامپایلرها، ویرایشگر متن و بسیاری از اجزای مهم سیستم‌عامل را تولید کرد، اما هسته (Kernel) آن — به نام Hurd — کامل نشد. این کمبود بعدها با هسته لینوکس جبران شد.



فصل دوم

فصل ۲

فرایند Boot(بوت) و Startup(راه‌اندازی)

دقیقاً مانند راهنمای مطالعه LPIC-1 ، این کتاب شامل توضیحات مفصل در مورد تمامی اهداف آزمون LPIC-2 است. هدف این کتاب کمک به شما برای موفقیت در هر دو آزمون LPIC-2 ، یعنی ۲۰۱ و ۲۰۲ است. این آزمون‌ها موضوعات پیشرفته‌تری نسبت به آزمون LPIC-1 را پوشش می‌دهند؛ از جمله هسته لینوکس، راه‌اندازی سیستم، فایل‌سیستم‌ها، عملیات شبکه، سرورهای DNS ، سرورهای وب، سرورهای فایل، سرورهای ایمیل، مدیریت کلاینت‌های شبکه و امنیت. این کتاب شما را گام‌به‌گام در تمامی این مباحث همراهی می‌کند تا برای پرسش‌های آزمون LPIC-2 آماده شوید.

هدف برنامه **LPIC-2** از سوی *Linux Professional Institute (LPI)* این است که سطح دانش لازم برای مدیریت شبکه‌های کوچک تا متوسطِ ترکیبی (ترکیب‌شده از مایکروسافت و لینوکس) را تعریف کند، با تمرکز ویژه بر روی سیستم‌عامل لینوکس. این برنامه برای متخصصانی طراحی شده که می‌خواهند دانش خود را که در سطح **LPIC-1** کسب کرده‌اند، توسعه دهند و به مرحله‌ای بالاتر برسانند.

انتظار می‌رود داوطلبان پیش از ورود به آزمون LPIC-2 ، آزمون **Linux Essentials** (اختیاری) و آزمون (**LPIC-1** یا آزمون معادل آن یعنی **CompTIA Linux+** را گذرانده باشند.

فردی که در آزمون LPIC-2 موفق شود، باید در کمترین حالت دارای دانش و تجربه در زمینه‌های زیر باشد:

- مدیریت چندین سرور لینوکس
- ارائه مشاوره به مدیران در زمینه مکانیزه‌سازی و خرید تجهیزات
- برنامه‌ریزی و مدیریت یک شبکه کوچکِ ترکیبی، شامل:
  - **LAN Server** سرور شبکه محلی
  - مدیریت کلاینت‌ها
  - سرویس **DHCP** اختصاص‌دهی پویا به IP ها
  - سرویس **DNS** سامانه نام دامنه
  - **NFS** اشتراک فایل در لینوکس
  - **Samba** ارتباط لینوکس با ویندوز برای اشتراک فایل و پرینتر
  - **Internet Gateway** دروازه اینترنت
    - **Firewall** دیوار آتش
    - سرویس‌های (**Mail** ایمیل
    - **OpenSSH** ارتباط امن و مدیریت راه دور
    - **VPN** شبکه خصوصی مجازی
    - **Web Cache/Proxy** کش وب و پراکسی برای مدیریت و کنترل دسترسی
  - **Internet Server** سرور اینترنت
    - **FTP Server** انتقال فایل
    - **Web Server** راه‌اندازی و مدیریت وب‌سایت‌ها
    - وب‌سرور با **Reverse Proxy** جهت افزایش امنیت و کارایی

ساختار **ESP**از سیستم‌فایل قدیمی **FAT – File Allocation Table**(جدول تخصیص فایل) مایکروسافت استفاده می‌کند تا بوت‌لودرها را ذخیره کند. در سیستم‌های لینوکسی، این پارتیشن معمولاً در مسیر / **boot/efi**مونت (mount) می‌شود و فایل‌های بوت‌لودر معمولاً با پسوند **efi** . ذخیره می‌گردند.

UEFI Boot Manager **مدیر بوت** UEFI

میان‌افزار **UEFI**دارای یک بوت‌لودر کوچک داخلی است که اغلب به آن **Boot Manager**(مدیر بوت) گفته می‌شود. این ابزار داخلی به شما اجازه می‌دهد که تعیین کنید کدام فایل بوت‌لودر باید اجرا شود. البته، همه توزیع‌های لینوکس از **UEFI Firmware**پشتیبانی نمی‌کنند. بنابراین اگر از سیستمی مبتنی بر **UEFI**استفاده می‌کنید، باید مطمئن شوید توزیع لینوکسی انتخابی شما آن را پشتیبانی می‌کند. در **UEFI**لازم است که هر فایل بوت‌لودری را که می‌خواهید در منوی بوت نمایش داده شود، در **Boot Manager Interface Menu**(منوی رابط مدیر بوت) ثبت کنید. برای این کار، در لینوکس برنامه‌ای به نام **efibootmgr**وجود دارد که به شما امکان می‌دهد ورودی‌های بوت را ایجاد یا حذف کنید و ترتیب بوت را تغییر دهید. رابط **UEFI**همچنین شامل یک محیط **Shell**(شل یا محیط دستوری داخلی) است که به شما امکان می‌دهد دستورات لازم برای تغییر تنظیمات بوت را وارد کنید یا مشخص کنید که هنگام هر بار بوت، کدام بوت‌لودر اجرا شود.

پشتیبانی از SSD و Nvme

یکی دیگر از فناوری‌های محبوب و جدید که معمولاً در سیستم‌های مبتنی بر UEFIیافت می‌شود، جایگزینی Legacy Hard Drives (هاردهای سنتی) با SSD – Solid State Drive(درایو حالت جامد) است. سخت‌افزارهای SSDاز کنترلرهای قدیمی هارد استفاده نمی‌کنند، بلکه از استاندارد **Nvme** – Non-Volatile Memory Express(رابط حافظه غیرفرار پرسرعت) بهره می‌برند.

Linux Kernel(هسته لینوکس) از نسخه ۳,۳ به بعد از سیستم‌های **Nvme**پشتیبانی می‌کند و فرایند بوت **UEFI**نیز در هنگام استفاده از سخت‌افزار **SSD**در سیستم‌های لینوکسی بدون مشکل کار می‌کند.

Linux Bootloaders **بوت‌لودرهای لینوکس**

برنامه **Bootloader**(بوت‌لودر) نقش پل ارتباطی میان **Firmware**(میان‌افزار سیستم) و **Linux Kernel**(هسته لینوکس) را ایفا می‌کند. در لینوکس چندین انتخاب برای بوت‌لودر وجود دارد. پرکاربردترین آن‌ها که احتمالاً با آن‌ها روبه‌رو خواهید شد عبارتند از:

LILO – Linux Loader (**بارگذار لینوکس**)

GRUB Legacy – Grand Unified Bootloader Legacy (**بوت‌لودر قدیمی گراب**)

GRUB2 – Grand Unified Bootloader v2 (**نسخه دوم بوت‌لودر گراب**)

LILO – Linux Loader

در نسخه‌های اولیه لینوکس، تنها بوت‌لودر موجود LILOبود. این بوت‌لودر قابلیت‌های محدودی داشت، اما هدف اصلی خود را برآورده می‌کرد: یعنی بارگذاری Linux Kernelاز طریق راه‌اندازی BIOS

LILO (Linux Loader)یکی از اولین بوت‌لودرهای سیستم‌عامل لینوکس بود که در دهه ۱۹۹۰ میلادی برای بارگذاری هسته لینوکس طراحی شد. در آن زمان، سیستم‌های مبتنی بر BIOS (Basic Input/Output System) رایج بودند و LILO به‌عنوان ابزاری ساده و کارآمد برای بارگذاری هسته لینوکس از دیسک سخت به حافظه عمل می‌کرد. در ابتدا، LILOبه‌عنوان بوت‌لودر پیش‌فرض در بسیاری از توزیع‌های لینوکس مورد استفاده قرار می‌گرفت. فایل پیکربندی آن معمولاً در مسیر / **etc/lilo.conf**قرار داشت و کاربران می‌توانستند سیستم‌عامل‌ها و هسته‌های مختلف را در این فایل تعریف کنند. LILO به‌صورت مستقیم از طریق BIOS سیستم بارگذاری می‌شد و نیازی به پشتیبانی از ویژگی‌های پیشرفته‌ای مانند UEFI (Unified Extensible Firmware Interface) نداشت.

تعامل با GRUB2

بوت‌لودر **GRUB2** هنگام راه‌اندازی سیستم، یک **منوی بوت Boot Menu** (نمایش می‌دهد که بسیار شبیه به نسخه‌ی قدیمی‌تر یعنی **GRUB Legacy** است. از طریق این منو می‌توانید سیستم‌عامل موردنظر خود را برای بوت انتخاب کرده یا تنظیمات بوت را تغییر دهید.

کلیدهای کاربردی در منوی بوت GRUB2

- **کلیدهای جهت‌نما (Arrow Keys) :**

برای جابه‌جایی بین گزینه‌های مختلف بوت از کلیدهای بالا و پایین استفاده کنید.

- **کلید E :**

برای **ویرایش موقت تنظیمات بوت** (مانند تغییر پارامترهای کرنل یا مسیر بوت) روی گزینه‌ی انتخاب‌شده فشار دهید.

پس از انجام تغییرات، با فشردن کلید **Ctrl + X** یا **F10** سیستم را با تنظیمات جدید بوت کنید.

- **کلید C :**

وارد **خط فرمان GRUB Command Line (GRUB2)** می‌شوید.

در این حالت می‌توانید دستورات تعاملی مانند ls(برای فهرست دیسک‌ها و پارتیشن‌ها) یا set(برای مشاهده متغیرهای بوت) را وارد کنید.

نکته در مورد رابط‌های گرافیکی

در برخی توزیع‌های لینوکس با رابط گرافیکی (مثل **Ubuntu**، منوی بوت GRUB2 معمولاً **پنهان** است تا فرآیند بوت سریع‌تر و زیباتر باشد.

اگر بخواهید این منو را ببینید یا به تنظیمات آن دسترسی پیدا کنید، کافی است **در لحظه‌ی روشن شدن سیستم، کلید Shift را نگه دارید** در سیستم‌های (BIOS یا **کلید Esc را چند بار فشار دهید** در سیستم‌های(EFI / UEFI .

بوت‌لودرهای جایگزین

هرچند **GRUB Legacy** و **GRUB2** محبوب‌ترین بوت‌لودرهای لینوکس هستند، اما در برخی توزیع‌ها یا شرایط خاص ممکن است با بوت‌لودرهای دیگر نیز مواجه شوید.

Systemd-boot

بوت‌لودر **Systemd-boot**(سیستم‌دی–بوت) به‌ویژه در توزیع‌هایی که از روش **systemd init** راه‌اندازی (systemd استفاده می‌کنند، محبوبیت پیدا کرده است. این بوت‌لودر یک **Boot Menu** ایجاد می‌کند و می‌تواند هر **EFI Boot Image** ایمیج بوت EFI را بارگذاری کند.

U-Boot

بوت‌لودر **U-Boot**(یوبوت) توانایی بوت از هر نوع **Disk**(دیسک) و بارگذاری هر نوع **Boot Image**(ایمیج بوت) را دارد.

پروژه سیسلینوکس Syslinux Project

پروژه **Syslinux** شامل پنج بوت‌لودر جداگانه است که هرکدام کاربرد خاصی دارند:

- SYSLINUX بوت‌لودر مخصوص سیستم‌هایی که از) Microsoft FAT Filesystem فایل‌سیستم FAT مایکروسافت استفاده می‌کنند. این بوت‌لودر معمولاً برای بوت از) USB Memory Stick فلش (USB رایج است.
- EXTLINUX یک مینی بوت‌لودر برای بوت از فایل‌سیستم‌های ext2، ext3، ext4 یا btrfs.

برنامه**systemctl** با استفاده از **دستورات (Commands)** مختلف مشخص می‌کند که چه عملیاتی روی یک واحد انجام شود. این دستورات می‌توانند وضعیت سرویس‌ها را بررسی کنند، سرویس‌ها را شروع یا متوقف کنند، فایل‌های پیکربندی را بارگذاری مجدد کنند و حتی سطح اجرای سیستم یا Target فعلی را تغییر دهند. جدول زیر مهم‌ترین دستورات**systemctl** و کاربرد آن‌ها را نشان می‌دهد:

دستور	توضیح
list-units	نمایش وضعیت کنونی تمام واحدهای پیکربندی شده و فعال سیستم
default	تغییر به Target پیش‌فرض سیستم، یعنی سطحی که سیستم در هنگام بوت به آن می‌رود
isolate	اجرای یک واحد مشخص و توقف تمام واحدهای دیگر؛ برای تغییر کامل سطح اجرایی سیستم
start name	شروع واحد مشخص شده، مانند یک سرویس یاTarget
stop name	متوقف کردن واحد مشخص شده
reload name	بارگذاری مجدد فایل پیکربندی واحد مشخص شده بدون توقف کامل آن
restart name	متوقف و دوباره راه‌اندازی کردن واحد مشخص شده؛ برای اعمال تغییرات اساسی یا رفع مشکلات
status name	نمایش وضعیت واحد مشخص شده؛ شاملPID ، وضعیت فعال یا غیرفعال بودن و مسیر فایل پیکربندی
enable name	پیکربندی واحد برای شروع خودکار هنگام بوت بعدی
disable name	پیکربندی واحد برای عدم شروع خودکار هنگام بوت بعدی

یک جدول با **مثال برای هر دستور systemd**

دستور	توضیح	مثال
list-units	نمایش وضعیت کنونی تمام واحدهای پیکربندی شده و فعال سیستم	**systemctl** list-units
default	تغییر به Target پیش‌فرض سیستم، یعنی سطحی که سیستم در هنگام بوت به آن می‌رود	**systemctl** get-default **systemctl** set-default graphical.target
isolate	اجرای یک واحد مشخص و توقف تمام واحدهای دیگر؛ برای تغییر کامل سطح اجرایی سیستم	**systemctl** isolate multi-user.target
start name	شروع واحد مشخص شده، مانند یک سرویس یاTarget	**systemctl** start nginx.service
stop name	متوقف کردن واحد مشخص شده	**systemctl** stop nginx.service
reload name	بارگذاری مجدد فایل پیکربندی واحد مشخص شده بدون توقف کامل آن	**systemctl** reload nginx.service
restart name	متوقف و دوباره راه‌اندازی کردن واحد مشخص شده؛ برای اعمال تغییرات اساسی یا رفع مشکلات	**systemctl** restart nginx.service
status name	نمایش وضعیت واحد مشخص شده؛ شاملPID ، وضعیت فعال یا غیرفعال بودن و مسیر فایل پیکربندی	**systemctl** status nginx.service
enable name	پیکربندی واحد برای شروع خودکار هنگام بوت بعدی	**systemctl** enable nginx.service
disable name	پیکربندی واحد برای عدم شروع خودکار هنگام بوت بعدی	**systemctl** disable nginx.service

فصل سوم

فصل ۳

نگهداری و پایداری سیستم لینوکس

پایدار نگه داشتن یک سیستم لینوکس کار ساده‌ای نیست و مجموعه‌ای از وظایف مهم و روزمره را شامل می‌شود. مدیر سیستم باید همیشه مراقب باشد که داده‌ها در امنیت کامل باشند و در صورت بروز مشکل بتوان آن‌ها را بازیابی کرد. این یعنی **پشتیبان‌گیری منظم (Backup)** به یکی از وظایف کلیدی تبدیل می‌شود. از طرف دیگر، سیستم باید توانایی پاسخ‌گویی به نیازهای کاربران را داشته باشد؛ بنابراین **ظرفیت‌سنجی (Capacity Planning)** اهمیت ویژه‌ای پیدا می‌کند. علاوه بر این، مدیر سیستم باید همیشه نرم‌افزارهای جدید را نصب کند، برنامه‌های قدیمی را به‌روزرسانی کند، و در عین حال کاربران سیستم را از وضعیت فعلی مطلع سازد. ترکیب این کارهاست که یک سیستم لینوکس را در حالت پایدار و قابل‌اعتماد نگه می‌دارد.

اطلاع‌رسانی به کاربران

ارتباط درست با کاربران سیستم یکی از ارکان اصلی مدیریت موفق به شمار می‌رود. کاربران می‌توانند مدیر سیستم‌های دیگر، برنامه‌نویسان، مشتریان یا حتی افراد عادی باشند. اطلاع‌رسانی صحیح باعث می‌شود اعتماد آن‌ها جلب شود و در مواقع ضروری همکاری بیشتری داشته باشند.بسیاری از سازمان‌ها سیاست‌های مشخصی برای اطلاع‌رسانی دارند؛ مثلاً ارسال ایمیل‌های گروهی، پیامک‌های خودکار یا اطلاعیه روی صفحات داخلی شرکت. اما لینوکس ابزارهای قدرتمند و درونی خود را نیز در اختیار مدیران قرار می‌دهد. این ابزارها نه‌تنها سریع و کارآمد هستند بلکه به‌صورت مستقیم با سیستم‌عامل و کاربران فعال تعامل دارند.

ابزارهای اطلاع‌رسانی در لینوکس

چندین فایل و دستور مهم در لینوکس وجود دارد که برای ارتباط با کاربران استفاده می‌شوند:

- **/etc/issue** : پیامی که قبل از ورود (login) در ترمینال محلی نمایش داده می‌شود.
- **/etc/issue.net** : پیامی که قبل از ورود از طریق SSH یا اتصال شبکه‌ای نمایش داده می‌شود.
- **/etc/motd**: پیام روز (Message of the Day) که بعد از ورود کاربر نمایش داده می‌شود.
- **/sbin/shutdown**: دستور خاموش یا ریبوت کردن سیستم، همراه با امکان ارسال پیام به کاربران.
- **/usr/bin/notify-send** : دستور ارسال اعلان گرافیکی (Popup Notification) در محیط‌های دسکتاپ.
- **/usr/bin/wall** : دستور ارسال پیام به همه‌ی کاربران فعال در ترمینال‌ها.

این ابزارها هرکدام در سناریوهای خاصی به کار می‌آیند. مثلاً اگر بخواهید پیامی برای همه کاربران قبل از ورود نمایش داده شود، باید از /etc/issue یا /etc/issue.net استفاده کنید. اگر هدفتان پیام‌های زنده و لحظه‌ای باشد، wall یا shutdown مناسب‌تر هستند.

پیام‌رسانی لحظه‌ای (Fluid Messaging)

یکی از روش‌های مهم ارتباط با کاربران، پیام‌رسانی لحظه‌ای است. این روش بیشتر برای زمانی استفاده می‌شود که یک اتفاق در حال رخ دادن است و کاربران باید بلافاصله در جریان قرار بگیرند. برای نمونه، وقتی سیستم قرار است خاموش شود یا وقتی یک سرویس حیاتی دچار مشکل می‌شود، مدیر سیستم باید پیام را به‌طور زنده به کاربران منتقل کند.

پیام‌رسانی لحظه‌ای معمولاً در دو حالت استفاده می‌شود:

۱. **شرایط اضطراری**، مثل خاموشی ناگهانی یا قطع سرویس حیاتی.
۲. **اطلاع‌رسانی تکمیلی**، وقتی که از قبل اطلاعیه داده شده اما نیاز است کاربران فعال دوباره هشدار بگیرند.

دستور wall

یکی از کاربردی‌ترین ابزارها برای اطلاع‌رسانی زنده، دستور **wall** است. این دستور پیام شما را برای همه کاربران فعال ارسال می‌کند.

است باعث از دست رفتن داده‌ها یا نارضایتی شدید شود. اما اگر با استفاده از ابزارهایی مثل wallیا shutdownاطلاع‌رسانی شود، کاربران زمان کافی برای ذخیره‌سازی کارهای خود خواهند داشت.

دستور write در لینوکس

در سیستم‌های لینوکس، دستور **write** یک ابزار ساده و در عین حال قدرتمند برای ارسال پیام مستقیم به یک کاربر خاص است. این دستور برای مواقعی استفاده می‌شود که شما به‌عنوان مدیر سیستم یا حتی یک کاربر بخواهید به فردی مشخص روی همان سیستم پیام فوری ارسال کنید. ساختار کلی دستور به صورت زیر است:

```
write username terminal_id
```

در این دستور، **username** نام کاربری کسی است که می‌خواهید پیام را برایش ارسال کنید و **terminal_id** شناسه ترمینالی است که آن کاربر در حال استفاده از آن است. شناسه ترمینال می‌تواند tty1, tty2برای ترمینال‌های فیزیکی و pts/0, pts/1برای ترمینال‌های شبیه‌سازی‌شده در محیط گرافیکی (GUI) باشد.

مثال استفاده از دستور write

۱. **مشاهده کاربران فعال و ترمینال‌های آن‌ها:**

ابتدا باید ببینیم چه کاربرانی وارد سیستم شده‌اند و روی کدام ترمینال هستند.

```
$ who
alice   pts/1   10:30
bob     pts/2   10:35
```

در اینجا کاربر aliceروی pts/1و کاربر bobروی pts/2فعال است.

۲. **ارسال پیام به یک کاربر مشخص:**

می‌خواهیم به کاربر alice پیام بفرستیم. ترمینال او pts/1است.

```
$ write alice pts/1
```

۳. سلام! امروز جلسه فنی ساعت ۳ برگزار می‌شود.

```
Ctrl+D
```

- بعد از تایپ پیام، **با زدن Ctrl+D پیام ارسال می‌شود.**
- کاربر alice پیام شما را فوراً روی ترمینال خود مشاهده می‌کند.

۴. **نکات مهم:**

- دستور write فقط پیام فوری ارسال می‌کند و پیام‌ها ذخیره نمی‌شوند.
- قبل از ارسال، مطمئن شوید پیام‌رسانی ترمینال مقصد فعال است(mesg y) .
- اگر پیام‌رسانی غیرفعال باشد(mesg n) ، پیام شما ارسال نمی‌شود و پیغام خطا دریافت می‌کنید.

بررسی دسترسی پیام با دستور who -T

قبل از استفاده از write، بهتر است بررسی کنید که کاربر مورد نظر امکان دریافت پیام را دارد یا خیر. برای این کار از دستور زیر استفاده می‌شود:

```
who -T
```

این دستور وضعیت دسترسی نوشتن (Write Access) هر کاربر وارد شده به سیستم را نمایش می‌دهد. علامت‌ها به این صورت هستند:

۲. **پشتیبان افزایشی Incremental Backup**

فقط داده‌هایی که از آخرین عملیات پشتیبان‌گیری تغییر کرده‌اند، کپی می‌شوند. سرعت ایجاد آن بالا و نیاز به فضای کمتر است، اما بازیابی داده‌ها طولانی‌تر است زیرا ابتدا باید **پشتیبان کامل** و سپس تمام پشتیبان‌های افزایشی بازگردانی شوند.

۳. **پشتیبان تفاضلی Differential Backup**

کپی تمام داده‌هایی که از آخرین پشتیبان کامل تغییر کرده‌اند. سرعت ایجاد آن بین پشتیبان کامل و افزایشی است و برای بازیابی، تنها نیاز به پشتیبان کامل و آخرین پشتیبان تفاضلی دارید.

۴. **پشتیبان لحظه‌ای Snapshot Backup**

یک روش ترکیبی است که ابتدا یک **کپی کامل** از داده‌ها ایجاد می‌کند و سپس با استفاده از **اشاره‌گرها و لینک‌ها**، تنها تغییرات جدید ذخیره می‌شوند. این روش فضای کمتری مصرف می‌کند و امکان بازگشت به هر نقطه زمانی برای بازیابی کامل سیستم را فراهم می‌کند. برخی نرم‌افزارها مانند **rsync** از این روش استفاده می‌کنند.

پشتیبان‌گیری لحظه‌ای (Snapshot) می‌تواند به صورت **Copy-on-Write** باشد که تنها تغییرات ذخیره می‌شوند، یا به صورت **Split-Mirror** که داده‌ها روی یک دستگاه آینه‌ای نگهداری می‌شوند و در هر پشتیبان‌گیری، کپی کاملی از داده‌ها ایجاد می‌شود.

بررسی بازیابی داده‌ها

وقتی صحبت از پشتیبان‌گیری می‌شود، مرحله‌ای حیاتی وجود دارد که اغلب کمتر مورد توجه قرار می‌گیرد: **بازیابی داده‌ها**. هیچ کس دوست ندارد به این موضوع فکر کند، اما واقعیت این است که در طول عمر یک سیستم، وقوع **حوادث ناخواسته** و نیاز به بازیابی داده‌ها اجتناب‌ناپذیر است. فایل‌ها به اشتباه حذف می‌شوند، هارد دیسک‌ها ممکن است خراب شوند، و فجایع بزرگ گاهی رخ می‌دهند. همانطور که مرگ و مالیات اجتناب‌ناپذیر هستند، بازیابی داده‌ها هم جزو واقعیات دنیای فناوری است.**برنامه‌ریزی بازیابی داده‌ها** باید جزو بخش اصلی استراتژی پشتیبان‌گیری باشد. جزئیات این برنامه به شدت به انتخاب‌های دیگر شما در برنامه پشتیبان‌گیری بستگی دارد، اما در هر حالت، باید برای سناریوهای زیر برنامه‌ریزی شود:

بازیابی فایل‌های منفرد یا چندگانه

این رایج‌ترین سناریو در بازیابی داده‌ها است. کاربران به طور تصادفی فایل‌های خود را حذف می‌کنند یا داده‌های خود را خراب می‌کنند. ارائه یک راهکار که کاربران بتوانند **خودشان فایل‌ها را بازیابی کنند**، بسیار مفید است. همچنین آموزش کاربران در پیشگیری از چنین مشکلاتی، بخش مهمی از استراتژی محافظت از داده‌ها است.

بازیابی پارتیشن یا دیسک کامل

اگر ساختارهای مناسب مانند **RAID** داشته باشید، احتمال وقوع چنین مشکلاتی کمتر می‌شود، اما همچنان امکان‌پذیر است. در این سناریو، **Recovery Time Objective (RTO)** شما نقش بسیار مهمی دارد و مشخص می‌کند که چه مدت زمان برای بازیابی داده‌ها قابل تحمل است. برنامه بازیابی باید دقیقا مشخص کند که چطور و چه زمانی پارتیشن‌ها یا دیسک‌های کامل بازیابی خواهند شد.

بازیابی سیستم

این حالت، بدترین سناریوی ممکن است؛ زمانی که کل سیستم و تمام داده‌ها باید بازیابی شوند. بازیابی سیستم به دو روش انجام می‌شود:

۱. **بازیابی کامل سیستم:** شامل بوت کردن سیستم عامل از یک توزیع زنده (Live Distribution) و سپس بازگردانی تمام فایل‌های لازم برای عملکرد لینوکس، از جمله **فایل‌های پیکربندی سیستم و تمام داده‌ها**.

۲. **بازیابی پیکربندی و داده‌ها:** ابتدا لینوکس دوباره نصب می‌شود و سپس فایل‌های پیکربندی سیستم و داده‌ها بازیابی می‌شوند.

- باینری‌ها و فایل‌های کتابخانه‌ای و مستندات را به مسیر مناسب منتقل می‌کند
- دسترسی‌ها و مجوزهای فایل‌ها را تنظیم می‌کند

پس از نصب، می‌توانید فایل tarball و دایرکتوری نصب موقت را پاک کنید، اما نگه داشتن tarball برای احتمالی حذف برنامه (uninstall) مفید است.

برای تست برنامه:

```
curl --version
```

مثال عملی: نصب Geany از سورس

مراحل نصبGeany IDE :

۱. با یک کاربر عادی وارد سیستم شوید و مرورگر وب را باز کنید.

۲. به آدرس www.geany.orgبروید و tarball برنامه را دانلود کنید (نام آن شبیه geany-#.#.tar.gzاست).

۳. پس از دانلود، وارد حساب کاربری با دسترسی **super user**شوید.

4. ابزارهای کامپایل و توسعه را نصب کنید:

```
RHEL/RedHat/Fedora:
yum groupinstall "Development Tools"
Debian/Ubuntu:
sudo apt-get install build-essential
```

۵. فایل tarball را به دایرکتوری فعلی کپی کنید.

۶. فایل را استخراج کنید:

```
tar -zxvf geany-#.#.tar.gz
```

اگر فرمت متفاوت است، گزینه‌های tar را مطابق فرمت تغییر دهید.

۷. وارد دایرکتوری نصب شوید:

```
cd geany-#.#
```

۸. فایل‌های README وINSTALL را بخوانید و بررسی کنید چه وابستگی‌هایی نیاز است.

۹. نصب کتابخانه GTK2 و) intltool برای(Debian/Ubuntu :

```
sudo apt-get install gtk+2.0 intltool
```

برایRHEL :

```
yum install gtk2-devel
```

۱۰. آماده‌سازی برای کامپایل:

```
./configure
```

در صورت خطای وابستگی، آن‌ها را نصب و دوباره configure را اجرا کنید.

۱۱. کامپایل برنامه:

```
make
```

۱۲. نصب نهایی با دسترسیsuper user :

```
sudo make install
```

۱۳. بررسی نسخه برنامه:

collectdیک **Daemon**است که مصرف منابع IT را پایش می‌کند. این نرم‌افزار به زبان **C**نوشته شده و قابلیت جمع‌آوری داده‌های **Local** و **Remote** را دارد.

- تنظیمات از طریق فایل **collectd.conf**در مسیر / /etcیا / /etc/collectdانجام می‌شود.
- گزینه **LoadPlugin**مشخص می‌کند چه پلاگین‌هایی فعال باشند.
- نیاز به ابزارهای نمایش داده‌ها مانند **Front End**دارد.
- اطلاعات بیشتر:collectd

3. MRTG (Multi Router Traffic Grapher)

MRTGنرم‌افزاری برای جمع‌آوری و گراف شبکه است.

- به زبان **Perl**نوشته شده
- قابلیت گراف گرفتن از تقریباً هر دستگاه شبکه‌ای را دارد
- صفحات **HTML**تولید می‌کند که نمودارهای شبکه را نمایش می‌دهند
- می‌تواند همراه با **RRDTool**استفاده شود
- اطلاعات بیشتر:MRTG

4. Nagios

Nagiosیکی از محبوب‌ترین نرم‌افزارهای پایش منابع است که در دو نسخه عرضه می‌شود:

- **Nagios Core**:رایگان و متن‌باز
- **Nagios XI**:نسخه تجاری

ویژگی‌ها:

- پایش سیستم‌ها، دستگاه‌های شبکه و سرویس‌ها
- قابلیت استفاده از **Plugins**برای تعریف بررسی‌های سفارشی
- نمایش وضعیت سیستم‌ها در یک **مرکز کنترل مرکزی(Centralized Dashboard)**
- ارسال هشدار از طریق ایمیل یا پیامک
- داده‌های جمع‌آوری شده می‌توانند با ابزارهای گرافینگ مانند **PNP4Nagios**و **nagiosgraph**تجسم شوند
- اطلاعات بیشتر:Nagios
- نسخه کاملاً خودکار:Nagios Community

5. Icinga

Icingaشاخه‌ای از **Nagios**است که پس از توسعه به دو محصول تقسیم شد:

- **Icinga1** : نسخه اصلی فورک شده ازNagios
- **Icinga2** : بازنویسی کامل، با رابط کاربری متفاوت و توسعه سریع‌تر

مزیت‌ها:

- سازگار باNagios
- رابط کاربری بهتر و توسعه سریع‌تر
- اطلاعات بیشتر:Icinga

6. RRDTool

RRDToolمخفف **Round-Robin Database Tool**است.

```
$ sudo iptraf -i eth0
$ sudo iptraf -s            # نمایش آمار خلاصه
$ sudo iptraf -b            # اجرای بدون رنگ برای لاگ
$ sudo iptraf -L logfile.txt # ذخیره گزارش در فایل
```

lsof

توضیح: نمایش فایل‌های باز توسط پروسه‌ها.

```
$ lsof
$ lsof -i :22
$ lsof -u username
$ lsof /var/log/syslog      # نمایش پروسه‌های باز روی فایل مشخص
$ lsof -c sshd              # نمایش پروسه‌ها بر اساس نام
$ lsof +D /home/user        # نمایش همه فایل‌های باز در دایرکتوری
```

mpstat

توضیح: نمایش آمار مصرف CPU بر اساس هسته‌ها.

```
$ mpstat
$ mpstat -P ALL 2
$ mpstat -u 5 3      # هر ۵ ثانیه، ۳ بار، آمار CPU
$ mpstat -I ALL      # نمایش وقایع اینتراپت
$ mpstat -A          # نمایش تمام آمار CPU
```

mtr

توضیح: ترکیب traceroute و **ping** برای بررسی مسیر شبکه.

```
$ mtr google.com
$ mtr -r -c 5 google.com
$ mtr --report-wide google.com
$ mtr -i 2 google.com     # فاصله ۲ ثانیه بین پینگ‌ها
$ mtr -z google.com       # مرتب‌سازی مسیرها بر اساس زمان پاسخ
```

netstat

توضیح: نمایش پورت‌ها، اتصال‌ها و سوکت‌ها.

```
$ netstat
$ netstat -tulnp
$ netstat -i
$ netstat -s
$ netstat -rn
$ netstat -an | grep ESTABLISHED
```

ntop

توضیح: مانیتورینگ ترافیک شبکه از طریق وب.

```
$ sudo ntop
```

مدیریت حافظه سیستم(System Memory Management)

یکی از مهم‌ترین وظایف هسته لینوکس، **مدیریت حافظه (Memory Management)** است. این وظیفه به هسته اجازه می‌دهد که **نحوه اجرای برنامه‌ها و ابزارهای سیستم در محدوده حافظه موجود** را کنترل کند.

هسته علاوه بر مدیریت حافظه **فیزیکی(Physical Memory)** ، قادر است **حافظه مجازی (Virtual Memory)** نیز ایجاد و مدیریت کند. حافظه مجازی، حافظه‌ای است که به طور واقعی وجود ندارد اما روی **هارد دیسک** ایجاد شده و مانند حافظه واقعی با آن رفتار می‌شود.

مفهوم Swap Space

برای این منظور، هسته از بخشی از هارد دیسک به نام **Swap Space** استفاده می‌کند. وقتی یک برنامه نیاز به حافظه بیشتری دارد و RAM پر است، هسته **صفحات حافظه (Memory Pages)** غیر فعال را به Swap منتقل می‌کند. این فرآیند به نام **Swapping Out** شناخته می‌شود.

وقتی برنامه‌ای به صفحه حافظه‌ای نیاز دارد که در Swap قرار دارد، هسته باید فضای کافی در RAM ایجاد کند، معمولاً با **جابجایی (Swapping In)** صفحه‌ای دیگر به Swap. واضح است که این عملیات زمان‌بر است و می‌تواند سرعت برنامه را کاهش دهد.

صفحات حافظه(Memory Pages)

حافظه به بلوک‌هایی به نام **Memory Pages** تقسیم می‌شود. هسته یک **جدول حافظه (Page Table)** نگهداری می‌کند که نشان می‌دهد کدام صفحات در RAM قرار دارند و کدام صفحات به Swap منتقل شده‌اند.

- **Swapping Out** انتقال صفحات غیر فعال به Swap
- **Swapping In** بازگرداندن صفحات مورد نیاز به RAM

این فرآیند به طور مداوم در طول اجرای سیستم ادامه دارد و یکی از دلایل کاهش عملکرد سیستم در مواقع کمبود RAM است.

مثال عملی

فرض کنید یک سرور لینوکس با ۴ گیگابایت RAM و Swap ۲ گیگابایت دارید. اگر یک برنامه سنگین مثل **MySQL یا Apache با بیش از ٦ گیگابایت مصرف حافظه** اجرا شود، هسته مجبور است صفحات غیر فعال برنامه‌های دیگر را به Swap منتقل کند. در نتیجه:

- دسترسی به حافظه کندتر می‌شود
- Disk I/O افزایش می‌یابد
- عملکرد کل سیستم کاهش می‌یابد

ابزارهای مرتبط برای پایش حافظه:

- **free –h** نمایش میزان حافظه آزاد و استفاده شده
- **vmstat 2 5** نمایش وضعیت حافظه، پردازنده و Swap هر ۲ ثانیه، ۵ بار
- **sar –r** نمایش گزارش مصرف حافظه به صورت تاریخی

نکات کلیدی مدیریت حافظه

۱. **تنظیم اندازه Swap**: اندازه Swap باید متناسب با RAM و نیاز برنامه‌ها باشد. معمولاً برای سرورها، Swap دو برابر RAM توصیه می‌شود.

۲. **حافظه فشرده (Compressed Memory)**: برخی توزیع‌ها از قابلیت **zswap** برای فشرده‌سازی صفحات Swap استفاده می‌کنند، که می‌تواند سرعت سیستم را در هنگام Swapping افزایش دهد.

مثال عملی: اگر بخواهید دو نسخه هسته مختلف روی یک سیستم داشته باشید، می‌توانید پوشه‌های جداگانه‌ای مانند / **usr/src/linux-4.3.3** و / **usr/src/linux-4.4.0** ایجاد کرده و هر بار لینک سمبلیک را به نسخه مورد نظر تغییر دهید. این کار باعث می‌شود بتوانید **همزمان چند نسخه هسته را روی سیستم تست کنید** بدون اینکه نسخه‌های قبلی حذف شوند.

ایجاد فایل پیکربندی هسته لینوکس(Creating the Kernel Configuration File)

قبل از اینکه بتوانید هسته جدید را از کد منبع کامپایل کنید، لازم است مشخص کنید کدام ویژگی‌های هسته (Kernel Features) را می‌خواهید در فایل باینری هسته جدید (Kernel Binary File) خود داشته باشید. این کار با استفاده از فایل پیکربندی هسته (Kernel Configuration File) انجام می‌شود. فایل پیکربندی هسته، یک فایل متنی (Text File) است که شامل خطوط جداگانه برای هر ویژگی هسته است و در آن، نام ویژگی (Feature Name) و تنظیمات آن (Setting) مشخص شده است. این فایل به صورت پیش‌فرض در مسیر / usr/src/linux/**.config** ذخیره می‌شود. اگر شما کد منبع هسته را از قبل نصب کرده‌اید، می‌توانید به این فایل نگاه کنید تا ببینید چه ویژگی‌هایی برای پیکربندی در دسترس هستند.

مثالی از بخشی از فایل **config** . در سیستم Ubuntu :

```
CONFIG_64BIT=y
CONFIG_X86_64=y
CONFIG_X86=y
CONFIG_INSTRUCTION_DECODER=y
CONFIG_PERF_EVENTS_INTEL_UNCORE=y
CONFIG_OUTPUT_FORMAT="elf64-x86–64"
CONFIG_ARCH_DEFCONFIG="arch/x86/configs/x86_64_defconfig"
CONFIG_LOCKDEP_SUPPORT=y
CONFIG_STACKTRACE_SUPPORT=y
CONFIG_HAVE_LATENCYTOP_SUPPORT=y
CONFIG_MMU=y
CONFIG_NEED_DMA_MAP_STATE=y
CONFIG_NEED_SG_DMA_LENGTH=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_GENERIC_BUG=y
CONFIG_GENERIC_BUG_RELATIVE_POINTERS=y
CONFIG_GENERIC_HWEIGHT=y
CONFIG_ARCH_MAY_HAVE_PC_FDC=y
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
```

همانطور که مشاهده می‌کنید، هر ویژگی با **CONFIG_** شروع می‌شود و مقدار آن می‌تواند **y**(فعال) یا **n**(غیرفعال) باشد.

ایجاد خودکار فایل پیکربندی

تنظیم دستی هر خط از فایل پیکربندی بسیار زمان‌بر و خطاپذیر است. خوشبختانه، می‌توان از **اسکریپت‌های خودکار (Automated Scripts)** برای ایجاد فایل **config** . استفاده کرد. این اسکریپت‌ها سوالاتی درباره ویژگی‌های مختلف هسته از شما می‌پرسند و بر اساس پاسخ‌ها، فایل پیکربندی را ایجاد می‌کنند.

دستور **make** در کد منبع، این اسکریپت‌ها را اجرا می‌کند. دستور make با استفاده از **targets** مشخص می‌کند کدام اسکریپت اجرا شود. چند targetپرکاربرد برای ایجاد فایل پیکربندی وجود دارد:

۷. **به‌روزرسانی هسته**: به مرور زمان، هسته‌ها برای **رفع اشکال و افزایش امنیت** باید به‌روزرسانی شوند. این کار می‌تواند با دانلود بسته‌های جدید هسته یا اعمال **Patch**ها انجام شود.

کار با فایل‌های ماژول(Working with Module Files)

در لینوکس، ماژول‌ها بخش بسیار مهمی از هسته سیستم را تشکیل می‌دهند. برای اینکه بتوانید به درستی با ماژول‌ها کار کنید، باید با چند فایل کلیدی آشنا شوید. همانطور که در مراحل قبل دیدید، ماژول‌های مورد نیاز برای پشتیبانی از هسته **در مسیر /lib/modules/** ذخیره می‌شوند. هر نسخه هسته پوشه مخصوص به خود را دارد، به عنوان مثال:

```
/lib/modules/4.3.3
```

این ساختار به شما امکان می‌دهد که برای هر نسخه هسته ماژول‌های جداگانه ایجاد کنید و در صورت نیاز از ماژول‌های سفارشی برای نسخه‌های مختلف هسته استفاده کنید. این موضوع به ویژه زمانی مهم است که سیستم شما چند هسته مختلف را پشتیبانی می‌کند یا می‌خواهید از ماژول‌های آزمایشی استفاده کنید بدون اینکه روی هسته اصلی تاثیر بگذارید.

فایل‌های بارگذاری ماژول‌ها هنگام بوت

ماژول‌هایی که هسته هنگام **بوت سیستم** بارگذاری می‌کند، در فایل‌های مختلف ذخیره می‌شوند:

- در توزیع‌های **Debian**، این ماژول‌ها در فایل / etc/modulesلیست شده‌اند.
- در توزیع‌های **Red Hat–based** مانند **Fedora**و**RedHat** ، این ماژول‌ها در پوشه / etc/modules-load.d/ذخیره می‌شوند.

در اکثر سیستم‌ها، بسیاری از ماژول‌های سخت‌افزاری به صورت **دینامیک** بارگذاری می‌شوند، زیرا سیستم به طور خودکار دستگاه‌های سخت‌افزاری متصل شده را شناسایی می‌کند. بنابراین، در بسیاری از مواقع، فایل‌های پیکربندی ماژول‌ها تعداد کمی ماژول دارند.

پیکربندی ماژول‌ها

گاهی نیاز است که یک **ماژول هسته** را با تنظیمات خاص سفارشی کنید، مانند تنظیمات سخت‌افزاری که برای عملکرد درست دستگاه لازم است. پیکربندی‌های ماژول‌ها در فایل / etc/conf.modulesذخیره می‌شوند. این فایل شامل پارامترهای مختلف ماژول‌ها و دستورالعمل‌های بارگذاری آن‌ها است.برخی از ماژول‌ها به **ماژول‌های دیگر وابسته** هستند و باید قبل از آن‌ها بارگذاری شوند. این وابستگی‌ها در فایل **modules.dep** تعریف شده‌اند و در مسیر / lib/modules/version/ذخیره می‌شوند، جایی که **version**نسخه هسته است. فرمت هر خط در این فایل به شکل زیر است:

```
modulefilename: dependencyfilename1 dependencyfilename2 ...
```

زمانی که دستور make modules_installاجرا می‌شود، برنامه **depmod**به صورت خودکار وابستگی ماژول‌ها را شناسایی کرده و فایل modules.depرا ایجاد می‌کند. اگر بعد از آن ماژول جدیدی اضافه یا تغییر دهید، باید به صورت دستی دستور زیر را اجرا کنید تا فایل modules.depبه‌روزرسانی شود:

```
# depmod
```

مدیریت ماژول‌های سفارشی باDKMS

اگر ماژول‌های خود را خارج از ماژول‌های استاندارد هسته کامپایل کنید، با یک مشکل مواجه خواهید شد: هر بار که هسته سیستم **به‌روزرسانی** می‌شود، ماژول‌های سفارشی نیز باید دوباره کامپایل شوند تا با هسته جدید سازگار باشند.راه حل این مشکل استفاده از **Dynamic Kernel**

- 1 → استفاده می‌شود (/) معمولاً برای ریشه
- 2 → برای بقیه فایل‌سیستم‌ها

مثال عملی

فرض کنید می‌خواهیم یک پارتیشن با UUID مشخص را روی / mnt/dataبه صورت دائمی مانت کنیم:

```
UUID=1234-ABCD /mnt/data ext4 defaults 0 2
```

این رکورد می‌گوید:

- فایل‌سیستم باUUID 1234-ABCD
- روی دایرکتوری / mnt/dataمانت شود
- نوع فایل‌سیستم ext4باشد
- از تنظیمات پیش‌فرض استفاده شود
- در پشتیبان‌گیری dumpلحاظ نشود(0)
- هنگام بوت، توسط fsckبعد از ریشه بررسی شود(2)

بیایید یک مثال کامل‌تر برای فایل / etc/fstabبا یک پارتیشن اصلی (/dev/sda1) و یک پارتیشن **Swap**بنویسیم.

مثال/etc/fstab

```
# <file system>    <mount point>    <type>     <options>        <dump>   <pass>

# پارتیشن ریشه
/dev/sda1          /                ext4       defaults         1        1

# پارتیشن سواپ
/dev/sda2          none             swap       sw               0        0
```

توضیح رکوردها

۱. **پارتیشن ریشه(/dev/sda1)**

- mount point: / → ریشه سیستم
- type: ext4 → نوع فایل‌سیستم
- options: defaults → استفاده از تنظیمات پیش‌فرض
- dump: 1 → پشتیبان‌گیری شود dumpمی‌توان توسط دستور
- pass: 1 → در هنگام بوت اولویت چک فایل‌سیستم بالاست (معمولاً ریشه اولین است)

۲. **پارتیشن(/dev/sda2) Swap**

- mount point: none → سواپ نیازی به دایرکتوری مانت ندارد
- type: swap → نوع فایل‌سیستم مخصوص سواپ
- options: sw → فعال کردن فضای سواپ
- dump: 0 → پشتیبان‌گیری نشود
- pass: 0 → بررسی نشود fsckهنگام بوت توسط

در این مثال ابتدا UUID فعلی نمایش داده می‌شود، سپس با دستور uuidgen یک UUID جدید ساخته می‌شود. در مرحله بعدی با استفاده از tune2fs و گزینه –U، UUID جدید روی پارتیشن اعمال می‌شود. در نهایت با اجرای دوباره blkid می‌توان تغییر انجام‌شده را بررسی کرد.

ابزارهای مدیریت XFS

سیستم‌فایل **XFS** یکی از قدرتمندترین فایل‌سیستم‌هاست که به دلیل کارایی بالا و قابلیت مقیاس‌پذیری، در توزیع‌هایی مانند RHEL و RedHat به‌طور گسترده استفاده می‌شود. این فایل‌سیستم به‌ویژه برای کار با فایل‌های بزرگ و محیط‌های سازمانی مناسب است.

ابزار xfs_admin برای تغییر ویژگی‌های فایل‌سیستم مانند UUID و برچسب استفاده می‌شود. این ابزار مشابه e2label یا tune2fs در ext عمل می‌کند.

ابزار xfs_fsr وظیفه بهینه‌سازی چیدمان فایل‌ها در سیستم‌فایل را دارد. با گذر زمان و حذف و اضافه شدن فایل‌ها، داده‌ها ممکن است به صورت پراکنده در دیسک ذخیره شوند. (fragmentation) این ابزار با بازآرایی داده‌ها به افزایش کارایی کمک می‌کند.

ابزار xfs_growfs امکان گسترش اندازه‌ی سیستم‌فایل را فراهم می‌کند. یکی از مزایای مهم XFS این است که می‌توان بدون نیاز به **unmount** کردن سیستم‌فایل، آن را گسترش داد. این قابلیت در محیط‌های سازمانی که زمان downtime بسیار حیاتی است، اهمیت زیادی دارد. برای مثال اگر یک سرور فایل دارید که همیشه باید در دسترس باشد، می‌توانید به سادگی فضای آن را افزایش دهید بدون اینکه سرویس از دسترس خارج شود.

ابزار xfs_admin

ابزار xfs_admin برای تغییر ویژگی‌های سیستم‌فایل XFS مانند UUID و برچسب استفاده می‌شود.

مثال‌ها:

- نمایش برچسب فعلی سیستم‌فایل:

```
sudo xfs_admin -l /dev/sda1
```

- تغییر برچسب سیستم‌فایل به "DataPartition" :

```
sudo xfs_admin -L DataPartition /dev/sda1
```

- تغییر UUID سیستم‌فایل به "۱۲۳۴۵۶۷۸-۹abc-def0-1234-56789abcdef0":

```
sudo xfs_admin -U 12345678-9abc-def0-1234-56789abcdef0 /dev/sda1
```

توجه: برای استفاده از xfs_admin، سیستم‌فایل باید unmounted باشد.

ابزار xfs_fsr

ابزار xfs_fsr برای بهینه‌سازی چیدمان فایل‌ها در سیستم‌فایل XFS و کاهش پراکندگی (fragmentation) استفاده می‌شود.

مثال‌ها:

- بهینه‌سازی تمام فایل‌های سیستم‌فایل‌های XFS متصل:

```
sudo xfs_fsr
```

- بهینه‌سازی یک فایل خاص:

```
sudo xfs_fsr /path/to/file
```

توجه: اجرای xfs_fsr ممکن است زمان‌بر باشد و بهتر است در زمان‌های کم‌ترافیک سیستم انجام شود.

بررسی یک آرایه RAID

پس از آنکه یک آرایه **RAID** ساخته شد، اولین گام مهم این است که مطمئن شویم همه چیز درست کار می‌کند. ساده‌ترین ابزار برای این بررسی، فایل / **proc/mdstat** است. این فایل در لینوکس وضعیت فعلی تمامی آرایه‌های RAID فعال را نشان می‌دهد.

برای مثال، اگر یک آرایه **RAID سطح ٦** به نام / dev/md0 ایجاد کرده باشیم، با اجرای دستور زیر می‌توانیم وضعیت آن را ببینیم:

```
# cat /proc/mdstat
```

خروجی نمونه:

```
Personalities: [raid6] [raid5] [raid4]
md0: active raid6 sdi1[3] sdh1[2] sdg1[1] sdf1[0]
     2093056 blocks super 1.2 level 6, 512k chunk, [...]

unused devices: <none>
```

در این مثال می‌بینیم که:

- آرایه / dev/md0 به صورت **فعال (active)** در حال کار است.
- چهار پارتیشن / dev/sdi1, /dev/sdh1, /dev/sdg1, /dev/sdf1 به طور کامل در آرایه حضور دارند.
- سطح RAID برابر **6** است و اندازه **chunk** برابر با ۵۱۲ کیلوبایت تنظیم شده است.

این نشانه‌ها می‌گویند که آرایه سالم و آماده‌ی استفاده است.

مفهوم Chunk و Stripe در RAID

یکی از مهم‌ترین مفاهیم در RAID ، درک ساختار **Stripe** و **Chunk** است.

- **Stripe نواری** :(زمانی که داده روی دیسک‌ها توزیع می‌شود، هر قطعه از داده در قالب یک نوار به چندین دیسک تقسیم می‌شود.
- **Chunk قطعه** :(کوچک‌ترین واحد داده در هر دیسک داخل آن Stripe است.

در مثال بالا، اندازه **Chunk = 512KB** است. این یعنی هر بار که داده‌ای نوشته می‌شود، به بلوک‌هایی ۵۱۲ کیلوبایتی تقسیم شده و سپس به دیسک‌ها توزیع می‌گردد.

هرچه اندازه **Chunk** بزرگ‌تر باشد، برای فایل‌های بزرگ عملکرد بهتری خواهیم داشت (مثل فایل‌های ویدئویی یا دیتابیس‌های حجیم). در مقابل، برای فایل‌های کوچک، Chunk کوچک‌تر می‌تواند کارایی را بهبود دهد.

بررسی جزئیات بیشتر با mdadm

علاوه بر فایل /proc/mdstat، می‌توان از حالت **Miscellaneous** ابزار **mdadm** برای بررسی دقیق‌تر استفاده کرد. مثال زیر، اطلاعات کامل آرایه / dev/md0 را نمایش می‌دهد:

```
# mdadm --misc --detail /dev/md0
```

یا حتی کوتاه‌تر:

```
# mdadm --detail /dev/md0
```

برای دیدن دیسک‌های **Nvme** متصل وnamespace های آن‌ها:

```
sudo nvme list
```

خروجی مشابه خواهد بود با جدولی شاملDevice ، Namespace(s)، Size، و مسیرها.

دستور id-ctrl — Identify Controller

برای گرفتن اطلاعات کامل کنترلر**Nvme** :

```
sudo nvme id-ctrl /dev/nvme0
```

برای نمایش به شکل خوانا با هدر:

```
sudo nvme id-ctrl -H /dev/nvme0
```

خروجی شامل فیلدهایی مانند mn (Model Name)، sn (Serial Number)، ver (Firmware Version)، nn (Number of Namespaces)و غیره خواهد بود.

دستور id-ns — Identify Namespace

برای گرفتن اطلاعات یک namespace خاص، مثلا namespace شماره ۱:

```
sudo nvme id-ns /dev/nvme0n1
```

یا با گزینه خوانا:

```
sudo nvme id-ns -H /dev/nvme0n1
```

خروجی فیلدهایی مانند nsze (Namespace Size)، lbaf (LBA Format), flbas، nuse، ncapو غیره را نمایش می‌دهد.

ایجاد و حذفNamespace

اگر دستگاه **Nvme** اجازه دهد، می‌توانی namespace جدید بسازی یا حذف کنی:

- **ایجادNamespace** :

```
sudo nvme create-ns /dev/nvme0 -s 100G -c 100G
```

در این مثال یک namespace با ظرفیت ۱۰۰ گیگابایت ساخته می‌شود.

- **حذفNamespace** :

فرض کن namespace با شناسه 3را بخوای حذف کنی:

```
sudo nvme delete-ns /dev/nvme0 -n 3
```

اتصال و جداکردن(attach / detach) Namespace

برای اتصال یک namespace به کنترلر:

```
sudo nvme attach-ns /dev/nvme0 -n 2 -c 0
```

در این مثال، namespaceبا شناسه ۲ به کنترلر شماره ۰ متصل می‌شود.

برای جدا کردن(detach) :

```
sudo nvme detach-ns /dev/nvme0 -n 2
```

درک و پیاده‌سازی این روش ذخیره‌سازی قبل از نیاز واقعی به آن اقدامی بسیار هوشمندانه است. در اینجا با مفاهیم اصلیLVM ، ساختار آن و همچنین مراحل عملی پیاده‌سازی آشنا خواهیم شد.

آشنایی باLVM

مدیریت حجم منطقی (Logical Volume Management – LVM) یک فناوری است که در سیستم‌های **لینوکس** به‌طور گسترده برای مدیریت دیسک و پارتیشن‌ها به کار می‌رود. این فناوری به شما اجازه می‌دهد چندین **پارتیشن** یا حتی چندین **دیسک فیزیکی** را به یکدیگر متصل کرده و به‌صورت یکپارچه از آنها استفاده کنید. نتیجه این کار یک فضای ذخیره‌سازی منعطف است که می‌توان آن را **فرمت** کرد، روی ساختار دایرکتوری مجازی لینوکس **مونت (mount)** نمود، داده‌ها را روی آن ذخیره کرد و حتی در آینده آن را تغییر اندازه داد.

ویژگی مهم LVM این است که یک **لایه انتزاعی (Abstraction Layer)** ایجاد می‌کند تا چندین پارتیشن یا دیسک فیزیکی به شکل یک واحد مستقل دیده شوند. این لایه باعث می‌شود که مدیریت داده‌ها ساده‌تر و بسیار منعطف‌تر از پارتیشن‌بندی سنتی باشد.

به عنوان مثال، فرض کنید در ابتدا یک دیسک ۵۰۰ گیگابایتی دارید و یک **حجم منطقی** روی آن ساخته‌اید. بعد از مدتی نیاز به فضای بیشتر پیدا می‌کنید. در سیستم‌های سنتی مجبور می‌شدید پارتیشن‌های جدید بسازید یا حتی داده‌ها را به دیسک دیگری منتقل کنید. اما در LVM تنها کافی است یک دیسک جدید (مثلاً ۱ ترابایت) اضافه کنید و آن را به همان حجم منطقی متصل نمایید. در نتیجه بدون نیاز به جابه‌جایی داده‌ها، فضای ذخیره‌سازی شما افزایش پیدا می‌کند.

اجزای اصلیLVM

LVMاز سه بخش اصلی تشکیل شده است که هر کدام نقشی کلیدی در ایجاد و نگهداری حجم‌های منطقی دارند:

۱. حجم فیزیکی(Physical Volume – PV)

حجم فیزیکی (PV) اولین مرحله در ساخت LVM است. یک PV معمولاً یک **پارتیشن دیسک** یا حتی کل یک **هارد دیسک** است که با دستور / sbin/pvcreateآماده استفاده در LVM می‌شود. هنگام اجرای این دستور، داده‌های ساختاری LVM شامل **برچسب حجم (Volume Label)**و **متادیتا (Metadata)** روی پارتیشن نوشته می‌شوند. این متادیتا اطلاعاتی مانند شناسه یکتا، وضعیت PV و تنظیمات پایه را ذخیره می‌کند.

نکته مهم این است که پس از ایجاد یکPV ، نمی‌توانید آن را به‌طور مستقیم برای ذخیره داده استفاده کنید. بلکه ابتدا باید آن را وارد یک **گروه حجم (VG)** کنید.

۲. گروه حجم(Volume Group – VG)

گروه حجم (VG) مجموعه‌ای از یک یا چند **PV**است که با دستور / sbin/vgcreateساخته می‌شود. VG مانند یک **استخر ذخیره‌سازی (Storage Pool)**عمل می‌کند که از آن برای ایجاد حجم‌های منطقی استفاده می‌کنیم. ویژگی مهم VG این است که شما می‌توانید چندین PV را به آن اضافه کنید و در نتیجه یک فضای ذخیره‌سازی بزرگ و یکپارچه بسازید.

به عنوان مثال، اگر سه دیسک ۵۰۰ گیگابایتی داشته باشید، می‌توانید آنها را به یک VG اضافه کنید و یک فضای کلی ۱.۵ ترابایتی ایجاد نمایید. سپس حجم‌های منطقی خود را بر اساس این فضا بسازید.هر PV فقط می‌تواند عضو یک VG باشد، اما یک دیسک می‌تواند چندین پارتیشن داشته باشد که هر کدام بهVG های مختلف اختصاص داده شوند.

۳. حجم منطقی(Logical Volume – LV)

جدول کامل پرتکل‌ها و پورت‌ها

لایه شبکه	پروتکل / سرویس	پورت TCP/UDP	توضیحات کاربردی
(Application) لایه کاربردی	HTTP	80 TCP	انتقال صفحات وب بدون رمزگذاری
	HTTPS	443 TCP	انتقال صفحات وب امن با TLS/SSL
	FTP	20 TCP (Data), 21 TCP (Control)	انتقال فایل
	SFTP	22 TCP	انتقال امن فایل با SSH
	SSH	22 TCP	دسترسی امن ریموت به سرور
	Telnet	23 TCP	دسترسی ریموت غیر امن (امروزه کمتر استفاده می‌شود)
	SMTP	25 TCP	ارسال ایمیل
	SMTPS	465 TCP	SMTP با رمزگذاری SSL
	IMAP	143 TCP	دریافت ایمیل با مدیریت پوشه‌ها
	IMAPS	993 TCP	IMAP امن با SSL
	POP3	110 TCP	دریافت ایمیل ساده
	POP3S	995 TCP	POP3 امن با SSL
	DNS	53 TCP/UDP	ترجمه نام دامنه به IP و بالعکس
	DHCP	67/68 UDP	تخصیص خودکار IP به کلاینت‌ها
	NTP	123 UDP	هماهنگی زمان در شبکه
	SNMP	161/162 UDP	مدیریت و مانیتورینگ شبکه
	LDAP	389 TCP/UDP	مدیریت دایرکتوری شبکه
	LDAPS	636 TCP	LDAP امن با SSL
	SIP	5060 TCP/UDP	پروتکل سیگنالینگ VoIP
	SIPS	5061 TCP	SIP امن با TLS
	HTTP/2	443 TCP	نسخه جدید HTTP با بهینه‌سازی
	MySQL	3306 TCP	اتصال به دیتابیس MySQL
	PostgreSQL	5432 TCP	اتصال به دیتابیس PostgreSQL
	Redis	6379 TCP	کش و NoSQL پایگاه داده
	MQTT	1883 TCP	پروتکل پیام‌رسانی سبک IoT
	MQTTs	8883 TCP	MQTT امن با TLS
(Transport) لایه انتقال	TCP	–	پروتکل اتصال‌گرا، تضمین تحویل داده‌ها
	UDP	–	پروتکل بدون اتصال، سریع برای صدا و ویدئو
	SCTP	–	پروتکل انتقال پیام مقاوم برای همزمانی
(Network) لایه شبکه	IP	–	مسیریابی داده‌ها، IPv4 و IPv6
	ICMP	–	کنترل پیام و تست وضعیت شبکه (ping)
	ARP	–	تبدیل آدرس IP به MAC در شبکه LAN
	RARP	–	معکوس ARP، پیدا کردن IP بر اساس MAC
	IGMP	–	مدیریت گروه‌های multicast
(Physical) لایه فیزیکی	Ethernet	–	شبکه سیم‌دار، استاندارد کابل مسی و فیبر
	Wi-Fi (802.11)	–	شبکه بی‌سیم، شامل باندهای ۲٫۴ و ۵ گیگاهرتز
	Fiber-optic	–	انتقال داده با نور، سرعت بالا و فاصله طولانی

مثال عملی:

```
# mtr www.google.com
```

```
                 My traceroute  [v0.85]
Host                    Loss%   Snt   Last   Avg  Best  Wrst StDev
1. 10.0.2.2              0.0%    24    7.0   1.5   0.0   7.0   1.7
2. homeportal            0.0%    24    5.4   5.3   0.8  17.4   3.9
3. 162-194-188-2.lightspeed.ip  8.3%    23   14.15 14.08 14.43 14.33 3.6
```

ویژگی‌های مهم mtr :

- نمایش درصد بسته‌های از دست رفته **(Loss%)** در هر هاپ
- نمایش زمان آخرین پکت، میانگین، بهترین و بدترین زمان
- بروزرسانی لحظه‌ای اطلاعات شبکه
- کمک به شناسایی مشکلات عملکردی در مسیر شبکه

آزمایش اتصال کلاینت و سرور

گاهی اوقات، ارسال پکت‌های **ICMP** کافی نیست و نیاز دارید **اتصال واقعی داده‌ها** را شبیه‌سازی کنید. دستور **nc (Netcat)** برای این کار عالی است.

Netcat می‌تواند هم به عنوان **کلاینت** و هم **سرور** عمل کند و به شما اجازه می‌دهد داده‌ها را از طریق شبکه ارسال و دریافت کنید.

گزینه	توضیح
4 –	استفاده از **IPv4**
6 –	استفاده از **IPv6**
b –	اجازه ارسال پکت‌های **Broadcast**
C –	ارسال **CRLF** در انتهای هر خط
D –	فعال کردن حالت **Debug**
d –	عدم خواندن از ورودی استاندارد
h –	نمایش راهنما
I len –	اندازه بافر دریافت **TCP**
i int –	فاصله زمانی بین ارسال و دریافت
k –	ادامه گوش دادن بعد از بسته شدن اتصال
l –	گوش دادن به یک پورت مشخص
n –	غیرفعال کردن جستجوی **DNS**
O len –	اندازه بافر خروجی **TCP**
p port –	تعیین پورت مبدأ
q sec –	تعداد ثانیه انتظار پس از **EOF**
s addr –	تعیین آدرس **IP** اینترفیس ارسال‌کننده
t –	اجازه اجرای **Telnet** اسکریپتی

۴. **کدهای پاسخ عملکردی(Action Response Codes)**

- **کدهای خطا** به جزئیات دقیق اشاره نمی‌کنند و تنها یک ایده کلی از مشکل ارائه می‌دهند.
- **کدهای اطلاعاتی** برای نمایش اطلاعات اضافی درباره یک دستور استفاده می‌شوند.
- **کدهای سرویس** وضعیت سرویس **SMTP** در اتصال را نشان می‌دهند.
- **کدهای عملکردی** نتیجه تلاش سرور برای اجرای دستورات **MAIL**، **RCPT**و **DATA** را نشان می‌دهند و به کلاینت اطلاع می‌دهند که چه اقدام بعدی انجام شود.

جدول نمونه کدهای SMTP:

کد	دسته‌بندی	توضیح
500	خطا	خطای نحوی، دستور شناسایی نشد
501	خطا	خطای نحوی در پارامترها
502	خطا	دستور پیاده‌سازی نشده
503	خطا	ترتیب دستورات اشتباه است
504	خطا	پارامتر دستور پیاده‌سازی نشده
211	اطلاعاتی	وضعیت سیستم یا کمک سیستم
214	اطلاعاتی	پیام راهنما
220	سرویس	سرویس آماده است
221	سرویس	سرویس در حال بسته شدن است
421	سرویس	سرویس در دسترس نیست
250	عملکردی	دستور ایمیل موفقیت‌آمیز بود
251	عملکردی	کاربر محلی نیست، پیام به مسیر هدایت می‌شود
354	عملکردی	شروع ورود پیام، پایان با <CRLF>.<CRLF>
450	عملکردی	عملیات انجام نشد، **Mailbox**در دسترس نیست
451	عملکردی	عملیات متوقف شد، خطا در پردازش
452	عملکردی	عملیات انجام نشد، فضای ذخیره‌سازی کافی نیست
550	عملکردی	عملیات انجام نشد، **Mailbox**غیرقابل دسترس
551	عملکردی	کاربر محلی نیست، مسیر جایگزین پیشنهاد شد
552	عملکردی	عملیات متوقف شد، حجم ذخیره‌سازی بیش از حد
553	عملکردی	عملیات انجام نشد، نام **Mailbox** مجاز نیست
554	عملکردی	تراکنش شکست خورد

نکته عملی: دانستن این کدها برای **عیب‌یابی ایمیل‌ها** حیاتی است. اگر یک پیام ایمیل بازگردد**(Bounce)** ، معمولاً کد پاسخ دقیق نیز همراه با ایمیل ارائه می‌شود تا علت شکست مشخص شود.

Extended SMTP (ESMTP)

با گذر زمان، محدودیت‌های **SMTP** استاندارد آشکار شد. به جای جایگزینی پروتکل قدیمی که در سطح جهانی استفاده می‌شد، **Extended SMTP**یا **ESMTP** معرفی شد تا **ویژگی‌های جدید** را به **SMTP** اضافه کند و در عین حال **سازگاری با پروتکل قدیمی** حفظ شود.

ویژگی‌های ESMTP:

- اضافه کردن دستورات اختیاری برای قابلیت‌های ویژه
- کلاینت و سرور باید **مذاکره کنند** که کدام دستورات استفاده شود

فصل ۷

سیستم نام دامنه (DNS) و نرم‌افزار BIND

سیستم نام دامنه (DNS) یک **پروتکل شبکه‌ای** است که در قلب بسیاری از فعالیت‌های اینترنتی قرار دارد. این سیستم وظیفه دارد **آدرس‌های IP**را به سیستم‌های مختلفی که خدمات ارائه می‌دهند اختصاص دهد و برای انجام این کار از یک **پایگاه داده توزیع‌شده** استفاده می‌کند. برای سیستم‌های **لینوکس**، نرم‌افزار **BIND**یکی از محبوب‌ترین بسته‌ها برای راه‌اندازی خدمات **DNS** است.

راه‌اندازی **DNS** از طریق **BIND** می‌تواند فرآیندی پیچیده باشد، زیرا نیاز به **ویرایش فایل‌های پیکربندی**، ایجاد و مدیریت **زون‌ها**، و همچنین **تأمین امنیت سرور DNS** و تراکنش‌های مختلف آن دارد. در این فصل، تمامی این موضوعات و موارد مرتبط با آن پوشش داده شده است.

پیکربندی یک سرور DNS

تا حدی، شما باید از **DNS** تشکر کنید که می‌توانید آدرس یک وب‌سایت، مانند **(www.ivytech.edu** یا نسخه کوتاه‌تر آن، مانند **ivytech.edu)** را در مرورگر وارد کرده و مرورگر شما را به سایت مورد نظر برساند. در این بخش، شما با نحوه عملکرد **DNS**، نصب نرم‌افزار **BIND**برای ارائه خدمات**DNS** ، فایل‌های پیکربندی مختلف**BIND** ، راه‌اندازی و توقف آن، تنظیم **لاگ‌گیری** در**BIND** ، و حتی پیاده‌سازی یک **سرور DNS محلی فقط با حافظه کش** آشنا خواهید شد.

درک DNS و BIND

قبل از اینکه وارد نحوه نصب و پیکربندی **BIND** شویم، بهتر است ابتدا نحوه کار **DNS**را درک کنیم. **DNS** در سال ۱۹۸۳ طراحی شد و ساختاری پیچیده اما منظم دارد که آزمون زمان را به خوبی پشت سر گذاشته است.

نگاه کلی به DNS

معمولاً وقتی کسی می‌خواهد به یک وب‌سایت دسترسی پیدا کند، مرورگر را باز کرده و حروف یا کلمات را که با نقطه جدا شده‌اند، در نوار آدرس وارد می‌کند، مانند:

```
www.ivytech.edu
```

این رشته‌های جدا شده با نقطه، یک **نام دامنه کامل یا FQDN (Fully Qualified Domain Name)** را تشکیل می‌دهند.**FQDN** ها آدرس‌های قابل فهم برای انسان هستند که افراد از طریق آن‌ها به خدمات مختلف مانند صفحات وب دسترسی پیدا می‌کنند.

اما سیستم‌های متصل به شبکه، مانند اینترنت، از**FQDN** ها استفاده نمی‌کنند؛ آن‌ها از **آدرس‌های IP** برای شناسایی یکدیگر استفاده می‌کنند. بنابراین نیاز است که**FQDN** ها به **IP** و برعکس تبدیل شوند. این نیاز معمولاً توسط **DNS**برآورده می‌شود. فرآیند تبدیل بین **FQDN** یک سیستم و آدرس **IP** آن، **حل نام یا Name Resolution** نامیده می‌شود.

در گذشته، فایل محلی **/etc/hosts**در سیستم‌ها برای حل نام استفاده می‌شد. هنوز هم در شبکه‌های کوچک محلی که شامل تعداد کمی سیستم هستند، می‌توان از این فایل برای حل نام‌ها استفاده کرد.

اگر **DNS** تنها از یک پایگاه داده در یک مکان استفاده می‌کرد، نتایج فاجعه‌بار می‌شد؛ شبکه‌های بزرگ هنگام انتظار برای حل نام‌ها دچار کندی می‌شدند. خوشبختانه، **پایگاه داده DNS** دارای ساختار **توزیع‌شده** و **قابلیت مقیاس‌پذیری بالا** است که عملکرد لازم برای حل نام‌ها را فراهم می‌کند.

ساختار پایگاه داده DNS

ساختار پایگاه داده **DNS** که به آن **فضای نام دامنه یا Domain Name Space** نیز گفته می‌شود، به صورت یک **درخت وارونه یا معکوس** است. این ساختار تا حدودی شبیه به ساختار **دایرکتوری مجازی لینوکس** است.

در**DNS** ، یک میزبان وب ممکن است با **نام میزبان و دامنه‌ها** که با نقطه از هم جدا شده‌اند، مانند **www.example.com**مشخص شود، در حالی که در ساختار دایرکتوری لینوکس، فایل‌ها و دایرکتوری‌ها با **اسلش‌های مستقیم** / جدا می‌شوند، مانند:

```
/home/rich/filea.txt
```

نصب و ابزارهای BIND

قبل از پیکربندی، باید نرم‌افزار BIND روی سیستم نصب شده باشد. حتی اگر فقط یک **سرور caching-only** پیکربندی می‌کنید، بهتر است **ابزارهای کمکی BIND** مانند بسته **bind-utils**نیز نصب شوند. این ابزارها شامل ابزارهایی مانند dig، nslookup و named-checkconfهستند که برای تست و عیب‌یابی ضروری‌اند.در برخی توزیع‌ها، فایل‌های پیکربندی BIND از پیش برای ارائه سرویس **DNS caching-only به کلاینت‌ها** تنظیم شده‌اند و شما تنها نیاز به تغییرات جزئی دارید. در توزیع‌های دیگر، ممکن است چندین فایل پیکربندی نیاز به تغییر داشته باشند تا سرور caching-only راه‌اندازی شود.

نکته عملی: برخی توزیع‌ها مانند **Ubuntu** با **dnsmasq**پیش‌نصب شده و فعال هستند. اگر قصد استفاده از BIND دارید، ممکن است تداخل و نتایج غیرمنتظره‌ای در کش رخ دهد. بنابراین قبل از راه‌اندازیBIND ، باید dnsmasq غیرفعال شود.

نصب BIND

در ابتدا باید بسته‌های مورد نیاز را نصب کنیم. بسته به توزیع لینوکس، دستور نصب متفاوت است:

در توزیع‌هایDebian/Ubuntu :

```
sudo apt update
sudo apt install bind9 bind9utils bind9-doc -y
```

در توزیع‌هایRHEL/CentOS/Fedora :

```
sudo dnf install bind bind-utils -y
```

پس از نصب، سرویس BIND به‌صورت پیش‌فرض فعال است، ولی بهتر است وضعیت آن را بررسی کنیم:

```
sudo systemctl status named
```

اگر سرویس فعال نبود:

```
sudo systemctl start named
sudo systemctl enable named
```

ساختار فایل‌های پیکربندی

BINDاز چند فایل اصلی برای پیکربندی استفاده می‌کند:

توضیح	فایل
فایل اصلی پیکربندی	*etc/bind/named.conf/*
تنظیمات عمومی، مانندforwarders	*etc/bind/named.conf.options/*
تعریفzone های محلی	*etc/bind/named.conf.local/*
محل کشDNS	*var/cache/bind/*

پیکربندیcaching-only DNS

یک **DNS caching-only**فقط درخواست‌ها را ذخیره و پاسخ می‌دهد و معمولاً برای سرعت‌بخشی به درخواست‌های DNS شبکه داخلی استفاده می‌شود.

فصل ۸

استفاده از لینوکس به عنوان وب سرور

یکی از رایج‌ترین کاربردهای **لینوکس** در محیط‌های سروری، استفاده از آن به عنوان **وب سرور (web server)** است. وب سرورهای لینوکس بر اینترنت غالب هستند و همچنین در میزبانی **برنامه‌های سازمانی (corporate intranet applications)** نیز بسیار محبوب‌اند. از ارائه صفحات استاتیک وب گرفته تا میزبانی **برنامه‌های وب داینامیک(dynamic web applications)** ، اغلب روی پلتفرم لینوکس با سرعت و کارایی بالاتری اجرا می‌شوند.

این فصل به شما نشان می‌دهد چگونه سیستم لینوکس خود را برای پشتیبانی از انواع محیط‌های وب سرور پیکربندی کنید. ابتدا توضیح می‌دهد **وب سرور چیست** و چه نرم‌افزارهایی برای تبدیل لینوکس به وب سرور لازم است. سپس به نصب و پیکربندی نرم‌افزار **Apache**می‌پردازد که محبوب‌ترین وب سرور حال حاضر است. از آنجا که Apache بسیار انعطاف‌پذیر است، پیکربندی آن ممکن است کمی پیچیده باشد، اما این فصل مراحل لازم برای راه‌اندازی وب‌سایت را به صورت مرحله به مرحله شرح می‌دهد.

پس از آن، **وب سرور پراکسی Squid** معرفی می‌شود. استفاده از پراکسی وب سرور در محیط‌هایی که نیاز به **کشینگ صفحات وب (web page caching)**برای صرفه‌جویی در پهنای باند شبکه و افزایش کارایی دارند، بسیار مفید است. در نهایت، **وب سرور nginx** بررسی می‌شود؛ محصولی که به سرعت در حال محبوب شدن است و ویژگی‌های جدید و جذابی ارائه می‌کند.

وب سرور چیست؟

قدرت اینترنت به **انتقال آزاد و باز اطلاعات** برمی‌گردد. در اوایل دوران اینترنت، دسترسی به فایل‌های داده روی سیستم‌های راه دور نیازمند داشتن **حساب کاربری (user login account)** و استفاده از برنامه‌های پیچیده **انتقال فایل (file transfer)** بود. برای تسهیل **به اشتراک‌گذاری سریع و باز اطلاعات**، نیاز به روشی جدید بود.

HTTPو تولد وب مدرن

- پروتکل **Hypertext Transfer Protocol (HTTP)**توسعه یافت تا انتقال داده‌ها بین سیستم‌ها به صورت ناشناس و ساده انجام شود.
- برای داده‌های عمومی که نیاز به حفاظت ندارند، HTTPاجازه می‌دهد **کلاینت** به سرور متصل شود، فایل‌های داده را دریافت کند و سپس ارتباط را قطع کند.
- این روش باعث شد سرعت دریافت داده‌ها از سیستم‌های راه دور افزایش یابد و به محبوب شدن اینترنت به عنوان محیطی برای **به اشتراک‌گذاری اطلاعات** کمک کرد.

HTMLو سازماندهی داده‌ها

با وجودHTTP ، هنوز مشکل این بود که پیدا کردن اینکه چه اطلاعاتی در کدام فایل متنی ذخیره شده، و چه فایل‌هایی باید از چه سروری دریافت شود، دشوار بود. برای حل این مشکل، نیاز به روشی برای **لینک دادن فایل‌های مرتبط** و ایجاد **استانداردی جدید** بود.

Hypertext Markup Language (HTML)دقیقاً این کار را انجام داد:

- HTMLبه نویسندگان اجازه می‌دهد **اسناد خود را با استفاده از لینک‌های ابرمتنی (hyperlinks)** به اسناد دیگر متصل کنند.
- لینک‌ها فایل‌های راه دور را با مشخص کردن **نام سرور** و **مسیر فایل** ارجاع می‌دهند.
- نویسندگان تنها کافی بود لینک‌ها را در متن خود قرار دهند تا کاربران بتوانند به آسانی از یک سند به سند دیگر بروند.

همچنین HTML امکان **فرمت‌دهی متن** را فراهم می‌کند:

کدهای رایج پاسخ HTTP

در جدول زیر تعدادی از **کدهای استاندارد پاسخ سرور HTTP** و معنای آن‌ها آورده شده است:

کد وضعیت	پیام متنی	توضیح
100	Continue	ادامه درخواست
101	Switch Protocols	تغییر پروتکل
102	Processing Request	پردازش درخواست
200	OK	موفقیت آمیز
201	Created	ایجاد شد
202	Accepted	پذیرفته شد
206	Partial Content	محتوای جزئی
207	Multi-Status	وضعیت چندگانه
208	Already Reported	قبلاً گزارش شده
226	IM Used	از IM استفاده شد
300	Multiple Choices	گزینه‌های متعدد
301	Moved Permanently	منتقل شد
302	Found	پیدا شد
303	See Other	مشاهده مورد دیگر
304	Not Modified	تغییر نکرده
305	Use Proxy	از پراکسی استفاده شود
307	Temporary Redirect	تغییر مسیر موقت
308	Permanent Redirect	تغییر مسیر دائمی
400	Bad Request	درخواست اشتباه
401	Unauthorized	غیرمجاز
403	Forbidden	ممنوع
404	Not Found	پیدا نشد
405	Method Not Allowed	متد مجاز نیست
406	Not Acceptable	غیرقابل قبول
408	Request Timeout	زمان درخواست تمام شد
409	Conflict	تعارض
410	Gone	حذف شده
500	Internal Server Error	خطای داخلی سرور
501	Not Implemented	پیاده‌سازی نشده

پس از نصـــب بســـته‌های ســـرور و وابســـتگی‌ها، می‌توانید فرآیند پیکربندی را شـــروع کنید. قبل از پیکربندی اشـــتراک فایل‌ها، مرور اجمالی بر **دایرکتوری‌ها و ابزارهای Samba** توصیه می‌شود.

مثال عملی: ایجاد یک Samba Share ساده

فرض کنید می‌خواهید یک پوشه /srv/samba/sharedرا بین کاربران لینوکس و ویندوز به اشتراک بگذارید.

۱. نصب بسته‌های مورد نیاز:

```
sudo yum install samba samba-client cifs-utils
```

۲. ایجاد پوشه به اشتراک گذاشته شده:

```
sudo mkdir -p /srv/samba/shared
sudo chown nobody:nogroup /srv/samba/shared
sudo chmod 0775 /srv/samba/shared
```

۳. افزودن اشتراک در فایل پیکربندی /etc/samba/smb.conf:

```
[shared]
  path = /srv/samba/shared
  browsable = yes
  writable = yes
  guest ok = yes
  read only = no
```

۴. راه‌اندازی سرویسSamba :

```
sudo systemctl start smb
sudo systemctl enable smb
sudo systemctl restart nmb
```

۵. دسترسی کاربران ویندوز:

- از **Run**ویندوز:\\server-ip\shared
- فایل‌ها را مشاهده و ویرایش کنید.

بررسی دایرکتوری‌ها و فایل‌هایSamba

علاوه بر دایرکتوری مستندات (Documentation Directory) که پیش‌تر توضیح داده شد، سه دایرکتوری اصلی Samba وجود دارند که فایل‌های مختلف Samba در آن‌ها قرار دارند. یکی از این دایرکتوری‌ها، محل ذخیره فایل‌های لاگ Samba (Log Files) اســـت: //var/log/samba.

Samba همچنین از دایرکتوری /var/lib/samba/و زیرشـــاخه های آن برای ذخیره اطلاعاتی مانند پایگاه داده کاربران (User Database)اســـتفاده می‌کند. به عنوان مثال، اطلاعات حســـاب کاربران Samba می‌تواند در یکی از پایگاه‌های داده smbpasswd، tdbsamیا ldapsamذخیره شود که معمولاً در همین دایرکتوری قرار دارند.

- برای پیاده‌سازی‌های کوچک Samba (کمتر از ۲۵۰ کاربر)، پایگاه داده tdbsamتوصیه می‌شود.
- پیاده‌ســـازی های بزرگ‌تر باید از پایگاه داده ldapsamهمراه با (Active Directoryدایرکتوری فعال یا LDAP – Lightweight Directory Access Protocol)پروتکل سبک دسترسی به دایرکتوری (استفاده کنند.
- پایگاه داده smbpasswdدر حال منسوخ شدن است و تنها برای سازگاری با نسخه‌های قدیمی Samba استفاده می‌شود.

راه‌اندازی سرویس Samba در زمان بوت

برای اینکه سرویس Samba هنگام **بوت شدن سیستم** خودکار اجرا شود، از دستور زیر استفاده می‌کنیم:

```
# systemctl enable smb
```

نکته امنیتی: در برخی توزیع‌ها ممکن است نیاز به تنظیم **SELinux Booleans** برای عملکرد صحیح Samba باشد. برای تست، می‌توان به‌طور موقت SELinux را غیرفعال کرد:

```
# sestatus
# nano /etc/selinux/config
# SELINUX=disabled
# reboot
```

پیکربندی فایروال برای Samba

برای عملکرد صحیح Samba با کلاینت‌ها، فایروال باید پورت‌های مورد نیاز را باز کند. برخی از پورت‌های مهم Samba در جدول زیر آمده‌اند:

شماره پورت	پروتکل‌ها	توضیح
53	TCP, UDP	DNS داخلی
88	TCP, UDP	Kerberos
135	TCP	End Point Resolution
137-139	TCP, UDP	خدمات NetBIOS
445	TCP	SMB over TCP
389	TCP, UDP	LDAP
464	TCP, UDP	Kerberos kpasswd
636	TCP	LDAP over SSL (LDAPS)
901	TCP, UDP	Samba Web Administration Tool (SWAT)
1024-5000	TCP	Dynamic RPC Ports
3268-3269	TCP	Microsoft Global Catalog
5353	TCP, UDP	Multicast DNS

برای مشاهده پورت‌های استفاده شده توسط Daemon های Samba ، از دستور **ss** استفاده می‌کنیم:

```
# systemctl status smb | grep PID
# ss -utlpn | grep <PID>
```

پیکربندی کلاینت Samba

پس از آماده شدن اشتراک‌های سرور، می‌توان کلاینت لینوکس را تنظیم کرد. مراحل کلی به شرح زیر است:

۱. نصب یا به‌روزرسانی بسته **Samba client** و ابزارهای جانبی.
۲. ایجاد حساب کاربری محلی مشابه سرور) همان UID) .
۳. ورود به سرویس Samba با **smbclient** و مشاهده اشتراک‌ها.
۴. ایجاد **Mount Point** برای اشتراک.
۵. نصب موقت اشتراک با دستور **mount**.

```
  option routers 192.168.1.1;
  option domain-name-servers 8.8.8.8, 8.8.4.4;
  default-lease-time 600;
  max-lease-time 7200;
}
```

در این مثال، کلاینت‌ها می‌توانند آدرس‌های بین ۱۹۲,۱۶۸,۱,۱۰ تا ۱۹۲,۱۶۸,۱,۵۰ دریافت کنند و روتر پیش‌فرض ۱۹۲,۱۶۸,۱,۱ و سرورهای DNS گوگل برای نام‌گذاری دامنه‌ها ارائه شده است.

نرم‌افزار DHCP در لینوکس

امروزه می‌توانید انواع مختلفی از دستگاه‌ها (Devices) را در شبکه پیکربندی کنید تا به عنوان سرور DHCP (DHCP Server) عمل کنند. بیشتر روترها (Routers) این سرویس را ارائه می‌دهند و همچنین اکثر سیستم‌های عامل سرور (Server-Oriented Operating Systems)مانند ویندوز سرور (Windows Server) و البته سرورهای لینوکس نیز این قابلیت را دارند.

محبوب‌ترین بسته نرم‌افزاری سرور DHCP لینوکس توسط Internet Systems Consortium (ISC – کنسرسیوم سیستم‌های اینترنتی) نگهداری می‌شود و به نام DHCPdشناخته می‌شود. تقریباً همه توزیع‌های سرور لینوکس این بسته را در مخازن نرم‌افزاری (Software Repositories)خود دارند، از جمله توزیع‌های مبتنی بر Debianو Red Hat.

زمانی که یک سرور DHCP روی شبکه خود راه‌اندازی کردید، باید به کلاینت‌های لینوکس (Linux Clients) خود بگویید که از این سرور برای دریافت آدرس‌های شبکه (Network Addresses) استفاده کنند. برای این کار به یک بسته نرم‌افزاری کلاینت DHCP (DHCP Client Software Package)نیاز دارید. سه بسته محبوب کلاینت DHCP برای لینوکس وجود دارد:

- **dhcpcd**
- **pump**
- **dhclient**

بیشتر توزیع‌های مبتنی بر **Debian**و **Red Hat**از بسته **dhclient**استفاده می‌کنند و حتی در برخی مواقع این بسته به طور پیش‌فرض نصب می‌شود، وقتی که یک **کارت شبکه (Network Card)** در حین فرآیند نصب سیستم شناسایی می‌شود. در اکثر محیط‌های دسکتاپ لینوکس، نیازی به انجام هیچ پیکربندی شبکه‌ای ندارید؛ برنامه نصب نرم‌افزار **کلاینت DHCP** را به طور خودکار نصب و فعال می‌کند. تنها کاری که باید انجام دهید این است که کابل شبکه را وصل کنید یا به شبکه بی‌سیم متصل شوید.

نصب سرور DHCP در لینوکس

اگر نیاز دارید یک **سرور DHCP** روی شبکه محلی خود پیاده‌سازی کنید، باید بسته **ISC DHCPd**را روی سرور لینوکس خود نصب کنید. برنامه DHCPd به اندازه‌ای محبوب است که در اکثر **مخازن توزیع‌های لینوکس** به طور پیش‌فرض موجود است، اما متأسفانه در توزیع‌های مختلف لینوکس با نام‌های متفاوتی شناخته می‌شود.

در حال حاضر، توزیع‌های مبتنی بر **Debian**این بسته را با نام **isc-dhcp-server**می‌شناسند (البته در نسخه‌های قدیمی‌تر، آن را **dhcp3-server**می‌نامیدند). برای نصب برنامهDHCPd ، می‌توانید از ابزار استاندارد **apt-get**استفاده کنید:

```
$ sudo apt-get install isc-dhcp-server
```

برای توزیع‌های مبتنی بر **Red Hat**، نام بسته به سادگی **dhcp**است:

```
# yum install dhcp
```

PAMاز ماژول‌ها (Modules) استفاده می‌کند تا درخواست احراز هویت از اپلیکیشن را به فرمت مناسب سیستم احراز هویت زیرین تبدیل کند و سپس پاسخ را به فرمت استاندارد قابل فهم برای اپلیکیشن‌ها تبدیل نماید.

مثال ماژول‌هایPAM

کلید PAM ماژول‌های احراز هویت زیرین هستند. هر ماژول از یک روش احراز هویت پشتیبانی می‌کند، اما شما می‌توانید چندین ماژول را در یک نصب PAM نصب کنید تا از چندین روش احراز هویت پشتیبانی شود.

ماژول(Module)	توضیح(Description)
pam_unix.so	استفاده از فایل‌های استاندارد / **etc/shadowوetc/passwd**
pam_krb5.so	استفاده از سیستم احراز هویت **Kerberos 5**
pam_ldap.so	استفاده از سرور **Lightweight Directory Access Protocol – LDAP**
pam_nis.so	استفاده از سرور **NIS**
pam_sss.so	استفاده از **System Security Services Daemon – SSSD**
pam_userdb.so	استفاده از فایل . **db**استاندارد برای احراز هویت

علاوه بر ماژول‌های احراز هویت، PAM**ماژول‌های کتابخانه‌ای اضافی (Library Modules)** دارد که ویژگی‌های اضافی برای مدیریت حساب کاربری و سیاست‌های امنیتی ارائه می‌دهند، مانند:

ماژول(Module)	توضیح(Description)
pam_access.so	ارائه **ورود ناشناس (Anonymous Logins)** برای سرورهای FTP عمومی
pam_chroot.so	ایجاد یک **محیط محدود (Locked Down Environment)** برای ورود
pam_console.so	فراهم کردن محیط **ورود کنسول(Console Login Environment)**
pam_cracklib.so	بررسی **رمز عبور(Password Strength)**
pam_deny.so	جلوگیری از **ورود به سیستم – (Login Attempts)** معمولاً به عنوان پیش‌فرض
pam_env.so	تنظیم یا حذف **متغیرهای محیطی(Environment Variables)**
pam_lastlog.so	ارائه زمان آخرین ورود کاربر
pam_limits.so	اعمال **محدودیت منابع (Resource Limits)** مانند تعداد فایل‌های باز
pam_listfile.so	اجازه یا جلوگیری از دسترسی بر اساس **فایل لیست(List File)**

تقریباً همه توزیع‌های لینوکس **PAMرا به طور پیش‌فرض نصب و استفاده می‌کنند**، حتی اگر فقط از فایل استاندارد / **etc/passwd** برای احراز هویت استفاده شود. نحوه استفاده هر اپلیکیشن از PAM توسط **فایل پیکربندی (PAM Configuration File) PAM** کنترل می‌شود.

پیکربندی(PAM (Configuring PAM

هر اپلیکیشنی (Application) که از PAM استفاده می‌کند، باید یک پیکربندی (Configuration Setup) داشته باشد که مشخص کند کدام ماژول احراز هویت (Authentication Module) باید استفاده شود و چگونه باید استفاده شود. متأسفانه، توزیع‌های مختلف لینوکس چندین روش متفاوت برای مدیریت سیستم پیکربندی PAM دارند.

روش اول: فایل واحد پیکربندی/etc/pam.conf

یکی از روش‌ها این است که همه پیکربندی‌های اپلیکیشن‌ها در یک فایل واحد (/etc/pam.conf) قرار داده شود. هر خط در این فایل دارای فرمت زیر است:

```
service type control module arguments
```

در پایگاه داده سلسله‌مراتبی، اشیاء به صورت درختی (Tree-Like) به یکدیگر متصل هستند. هر درخت LDAP باید یک شیء ریشه (Root Object)داشته باشد و اشیاء دیگر به ترتیب به آن متصل شوند. اشیاء از طریق ارجاع (Reference) به سایر اشیاء متصل می‌شوند، مشابه سیستم نام‌گذاریDNS .

- مزیت ساختار سلسله‌مراتبی: زمان خواندن داده‌ها سریع است، اما زمان نوشتن کند است. این روش فرض می‌کند که یک شیء پس از درج در پایگاه داده، چندین بار خوانده می‌شود اما به ندرت به‌روزرسانی می‌شود. این برای اکثر شبکه‌ها مناسب است، زیرا نام کاربری یکبار ایجاد می‌شود و سرورها هر بار که کاربر وارد می‌شود، آن را می‌خوانند.

اجزای پایگاه دادهLDAP

پایگاه داده LDAP شامل یک یا چند اسکیما (Schema) است. اسکیما تعریف می‌کند که چه اشیایی (Objects) در پایگاه داده وجود دارند و چگونه به هم مرتبط هستند. هر شیء پایگاه داده توسط کلاس شیء (Object Class) تعریف شده و یک شناسه منحصر به فرد (Unique Object ID)به آن اختصاص داده می‌شود.

- کلاس شیء: یک قالب شامل مجموعه‌ای از ویژگی‌ها (Attributes) که شیء دارد.
- ویژگی‌ها: اطلاعات واقعی شیء را ذخیره می‌کنند، مانند نام کاربری برای شیء کاربر یا نام سرور برای شیء سرور. هر ویژگی شامل نام (Name)و مقدار داده مرتبط (Associated Data Value) است.

نامگذاری اشیاء درLDAP

درLDAP ، **درخت LDAP** از استاندارد **X.500**برای نامگذاری اشیاء استفاده می‌کند. مثال:

```
o="Sample, Inc.", c="US"
```

این نام به شیء شرکت **Sample, Inc.**که در ایالات متحده واقع شده اشاره دارد.

oو cبه ویژگی‌های شیء اشاره دارند: سازمان (Organization) و کشور(Country) .

برای ساده‌تر کردن نام‌ها، بسیاری از شرکت‌ها از **نام دامنه ثبت شده خود** برای درخت LDAP استفاده می‌کنند، به شکل زیر:

```
dc="sample", dc="com"
```

dcمخفف **Domain Component**است.

سپس می‌توان اشیاء اضافی زیر درخت شرکت ایجاد کرد. هر شیء باید با یک نام **منحصر به فرد (Distinguished Name - DN)** در پایگاه داده تعریف شود، مانند:

```
cn="rich", dc="engineering", dc="mycompany", dc="com"
```

این روش امکان ایجاد **شاخه‌ها (Branches)** و **اشیاء نگهدارنده (Container Objects)** را برای مدیریت بهتر پایگاه داده فراهم می‌کند.

فرمت تبادل دادهLDAP (LDIF)

با توجه به چندین شیء و ویژگی‌ها، گاهی پیدا کردن داده مورد نظر دشوار است. **LDAP Data Interchange Format (LDIF)** یک استاندارد متنی برای تعریف اشیاء LDAP و ویژگی‌های آن‌ها است.

```
dn: <distinguished name>
<attrtype>: <attrvalue>
<attrtype>: <attrvalue>
...
```

contact me:

seilany.ir
learninghive.ir
emperor-os.ir
Hubuntu.ir
predator-os.ir
telegram:@seilany

HosseinSeilani

Designer,Developer and Linux sysadmin

Founder and Developer of Emperor-OS, Hubuntu and Predator-OS. I bring significant experience as a Linux/Windows sysadmin and graphical web design,UX/UI to the Open Source Community.

LITTLE-PSYCHO
DESTRUCTIVE LINUX

Emperor-OS

Predator-OS