



The Philosopher Penguin

LINUX AND THE UNIX Philosophy ^{1st} Edition

معرفی کوتاه کتاب

بر خلاف بسیاری از کتاب‌هایی که صرفاً بر آموزش نحوه‌ی استفاده از لینوکس تمرکز دارند، کتاب «لینوکس و فلسفه‌ی یونیکس» به بررسی «شیوه‌ی تفکر لینوکسی» می‌پردازد و توضیح می‌دهد چرا لینوکس پیاده‌سازی برتری از این سیستم عامل قدرتمند است. این کتاب نسخه‌ای بازنگری شده و گسترش یافته از یک اثر کلاسیک در علوم کامپیوتر است و هر فصل آن به‌طور کامل با محتوای مرتبط با لینوکس به‌روزرسانی شده است. تا به امروز هیچ کتابی به فارسی در فضای یونیکس یا لینوکس این موضوع را به‌طور خاص مورد بررسی قرار نداده است.

مرجع
فلسفه
سیستم
عامل

فلسفه لینوکس و یونیکس

Linux and Unix Philosophy

حسین سیلانی

ویرایش اول

۱۴۰۴



سرشناسه	سیلانی، حسین،
عنوان و نام پدیدآور	فلسفه لینوکس و یونیکس
مشخصات نشر	حسین سیلانی؛ ویراستاری حسین سیلانی.
مشخصات ظاهری	تهران: انتشارات یافته، ۱۴۰۴.
شابک	۲۷۰ ص.
وضعیت فهرست نویسی	۹۷۸، ۶۲۲، ۴۰۰، ۱۵، ۶
موضوع	فیپا
موضوع	سیستم عامل
موضوع	Operating System
موضوع	سیستم‌های عامل (کامپیوتر)
موضوع	Operating systems (Computers)
شناسه افزوده	سیلانی، حسین، ۱۳۶۰،
رده بندی کنگره	۷۶/۷۶QA
رده بندی دیویی	۴۳۲/۰۰۵
شماره کتابشناسی ملی	۹۶۷۳۵۹۳
اطلاعات رکورد کتابشناسی	فیپا
انتشارات یافته: ناشر کتاب‌های دانشگاهی	

عنوان: مفاهیم سیستم عامل
تالیف: حسین سیلانی
صفحه آرای و ویراستاری: حسین سیلانی
چاپ و صحافی: موسسه مهرداد
طراحی جلد: حسین سیلانی
چاپ اول: ۱۴۰۴
شمارگان: ۱۰۰۰ جلد
شابک: ۶، ۰۱۵، ۴۰۰، ۶۲۲، ۳۷۸
حق چاپ محفوظ است.

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.Pscho.ir>
 <https://emperor.os.ir/>
 <https://ubuntu.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:

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

منتظر حضور و همراهی شما هستم!

تقديم به تو

يگانه فرزانه دلم

فهرست مطالب

مقدمه	۱۳-
فصل ۱	۱۸-
تاریخچه	۱۸-
ریشه‌های فلسفه یونیکس	۱۸-
کن تامپسون	۱۸-
دنيس ريچي	۱۸-
اصول کلیدی فلسفه یونیکس	۱۹-
تأثیر فلسفه یونیکس بر جنبش "ابزارهای نرم‌افزاری"	۱۹-
فلسفه یونیکس به عنوان یک فرهنگ توسعه نرم‌افزار	۲۰-
فلسفه یونیکس در دنیای امروز	۲۰-
مبدأ فلسفه یونیکس: از داگ مک‌ایلروی تا ریچی و تامپسون	۲۰-
داگلاس مک‌ایلروی: (Doug McIlroy) پیشگام فلسفه یونیکس و pipes	۲۰-
اصول فلسفه یونیکس بر اساس داگ مک‌ایلروی	۲۲-
فلسفه یونیکس توسط پیتراچ. سالوس	۲۲-
ملاحظات طراحی ریچی و تامپسون در مقاله سال ۱۹۷۴	۲۳-
برایان کرنیگان و فلسفه یونیکس: طراحی برنامه‌ها در محیط یونیکس	۲۵-
انتقاد از افزایش گزینه‌ها و ویژگی‌ها	۲۵-
فلسفه ابزارهای نرم‌افزاری: قدرت در ترکیب	۲۶-
مقایسه ابزارهای یونیکس با سیستم‌های دیگر	۲۶-
نمونه‌هایی از فلسفه یونیکس در عمل	۲۶-
فلسفه یونیکس: سادگی، ترکیب‌پذیری و مینیمالیسم	۲۷-
سادگی و مینیمالیسم: قلب فلسفه یونیکس	۲۷-
طراحی درست: به جای افزودن گزینه، به ریشه مشکل فکر کنید	۲۸-
فلسفه یونیکس در دنیای امروز	۲۸-
بازگشت به اصول	۲۸-

- ۲۹ - اصل DOTADIW در فلسفه یونیکس
- ۲۹ - تاریخچه و منشأ
- ۳۰ - انتقاد از: systemd: نقض اصل DOTADIW
- ۳۰ - مزایای اصل DOTADIW
- ۳۱ - چالش‌های اجرای اصل DOTADIW
- ۳۱ - ۱۷ قانون یونیکس اریک ریموند
- ۳۱ - فلسفه‌ای برای توسعه نرم‌افزارهای کارآمد
- ۳۳ - اریک اس. ریموند
- ۳۴ - زندگی و پیشینه
- ۳۴ - آثار و نوشته‌ها
- ۳۵ - فلسفه و دیدگاه‌ها
- ۳۷ - "بدتر بهتر است": فلسفه طراحی یونیکس و تأکید بر سادگی
- ۳۸ - فلسفه "بدتر بهتر است" در عمل: مثال‌هایی از یونیکس
- ۳۸ - مزایای رویکرد "بدتر بهتر است"
- ۳۹ - انتقادات به رویکرد "بدتر بهتر است"
- ۳۹ - انتقاد از فلسفه طراحی یونیکس: رابط کاربری و تجربه کاربری
- ۴۰ - تحلیل نورمن از رابط کاربری یونیکس از دیدگاه علوم شناختی
- ۴۰ - عدم همخوانی مدل شناختی کاربران با طراحی یونیکس
- ۴۰ - مشکل یادگیری و تعامل در یونیکس
- ۴۱ - اشتباهات فاجعه‌بار در یونیکس
- ۴۱ - مشکلات رابط کاربری یونیکس
- ۴۱ - نیاز به طراحی کاربرمحور
- ۴۲ - مقدمه‌ای کوتاه بر فلسفه یونیکس: ماژولار بودن و سادگی
- ۴۳ - ماژولار بودن: قلب فلسفه یونیکس
- ۴۴ - قانون ترکیب: قدرت در اتصال برنامه‌ها

- چرا قانون ترکیب اهمیت دارد؟ - ۴۵ -
- قاعده سکوت: کمتر گفتن، بیشتر انجام دادن - ۴۶ -
- زمینه تاریخی فلسفه یونیکس: سادگی و کارایی در برابر پیچیدگی - ۴۷ -
- تأثیر محدودیت‌های سخت‌افزاری بر طراحی یونیکس - ۴۸ -
- رشد و تحول سیستم‌عامل‌های شبه یونیکس - ۴۹ -
- کاهش سهم بازار یونیکس و تصورات نادرست - ۵۰ -
- نقش نرم‌افزار آزاد و کپی‌لفت در احیای یونیکس - ۵۱ -
- میراث پایدار فلسفه یونیکس - ۵۱ -
- کنتراست فلسفه یونیکس با سیستم‌های اختصاصی: سادگی در برابر پیچیدگی - ۵۳ -
- ویژگی‌های سیستم‌عامل‌های اختصاصی - ۵۴ -
- کد منبع: تفاوت در دسترسی و شفافیت - ۵۵ -
- تفاوت در اهداف و اولویت‌ها - ۵۵ -
- اهداف بازارمحور و رقابت آزاد - ۵۶ -
- انحراف از فلسفه ایده‌آل در سیستم‌عامل‌های شبه یونیکس - ۵۷ -
- جمع‌بندی: سادگی در برابر پیچیدگی - ۵۸ -
- احیای فلسفه یونیکس در لینوکس - ۵۹ -
- گسترش فلسفه یونیکس در عصر اینترنت - ۶۱ -
- اصول فلسفه یونیکس - ۶۲ -
- اصول فرعی فلسفه یونیکس - ۶۴ -
- تأثیر فلسفه یونیکس بر توسعه نرم‌افزارهای آزاد - ۶۶ -
- تأثیر فلسفه لینوکس: تغییر پارادایم در توسعه نرم‌افزار و جامعه - ۶۶ -
- فلسفه زندگی و فلسفه فناوری - ۶۷ -
- فلسفه سیستم‌عامل‌ها: از ویندوز تا لینوکس - ۶۸ -
- فلسفه لینوکس: آزادی، مشارکت و شفافیت - ۶۹ -
- تأثیر فلسفه لینوکس بر جامعه - ۷۰ -

- ۷۱- تغییر پارادایم در توسعه نرم افزار
- ۷۲- مفهوم رادیکال در فلسفه یونیکس و لینوکس
- ۷۳- تفاوت لینوکس با سایر سیستم عامل ها: آزادی در برابر محدودیت
- ۷۳- روشنگری در فلسفه لینوکس و یونیکس
- ۷۴- اصول کلیدی فلسفه یونیکس و لینوکس
- ۷۶- تأثیر فلسفه لینوکس بر دنیای فناوری
- ۷۶- روشنگری در دنیای لینوکس
- ۷۶- ترمینال و قدرت پنهان در لینوکس
- ۷۸- کنترل کامل: آزادی و مسئولیت در لینوکس
- ۸۰- دستورالعمل اصلی لینوکس: احترام، آزادی و قدرت
- ۸۱- لینوکس، یک اثر هنری نرم افزاری
- ۸۱- متن باز و فلسفه لینوکس: انقلابی در دنیای نرم افزار
- ۸۱- متن باز چیست و چرا اهمیت دارد؟
- ۸۲- مجوزهای متن باز: از GPL تا BSD و آپاچی
- ۸۳- مدل توسعه‌ی متن باز: تغییر روش ساخت نرم افزار
- ۸۳- نمونه‌های موفق متن باز در دنیای واقعی
- ۸۴- کتاب‌های مهم درباره‌ی متن باز و فلسفه‌ی آن
- ۸۴- متن باز: آینده‌ی توسعه‌ی نرم افزار؟
- ۸۵- لینوکس در برابر مدل‌های سنتی توسعه نرم افزار: نگاهی به تفاوت‌های اساسی
- ۸۵- مدل سنتی توسعه‌ی نرم افزار: کنترل، سلسله‌مراتب و سیاست‌های سخت‌گیرانه
- ۸۶- مدل لینوکس: رویکردی متفاوت و انقلابی در توسعه‌ی نرم افزار
- ۸۶- مقایسه‌ی توسعه‌ی نرم افزار سنتی و مدل لینوکس
- ۸۷- مدل توسعه‌ی چاپک (Agile) و شباهت آن با لینوکس
- ۸۷- چرا مدل توسعه‌ی لینوکس موفق است؟
- ۸۹- لینوکس: یک اکوسیستم توسعه‌ی باز و مشارکتی

نکاتی برای کاربران لینوکسی ----- ۹۳ -

نکاتی برای کاربران یونیکس ----- ۹۴ -

فصل ۲ ----- ۹۸ -

خلق کرنل لینوکس ----- ۱۰۲ -

نامگذاری ----- ۱۰۳ -

تاریخچه ----- ۱۰۴ -

لینوکس تحت gpl ----- ۱۰۸ -

تاکس ----- ۱۱۰ -

انجمن لینوکس ----- ۱۱۱ -

ازمایشگاه توسعه متن باز و بنیاد لیتوکس ----- ۱۱۲ -

شرکت ها ----- ۱۱۳ -

میزکار لینوکسی Bottom of Form ----- ۱۱۴ -

“لینوکس منسوخ شده است” ----- ۱۱۶ -

رقابت و همکاری مایکروسافت ----- ۱۱۸ -

گروه های SCO - The Santa Cruz Operation ----- ۱۲۱ -

حقوق علامت تجاری ----- ۱۲۲ -

کرونولوژی ----- ۱۲۴ -

جذب تجاری و مردمی ----- ۱۲۹ -

توسعه فعلی ----- ۱۳۲ -

طراحی ----- ۱۳۳ -

فصل ۳ ----- ۱۳۷ -

اجزای نصب شده در یک سیستم لینوکس ----- ۱۴۲ -

رابط کاربری ----- ۱۴۷ -

زیرساخت ورودی ویدیو ----- ۱۵۱ -

تاریخچه توسعه ----- ۱۵۲ -

- ۱۵۲ - توزیع لینوکس و نرم افزار
- ۱۵۲ - نقش هسته لینوکس در توسعه سیستم عامل
- ۱۵۳ - تفاوت لینوکس با یونیکس
- ۱۵۳ - توزیع های معروف لینوکس
- ۱۵۳ - مزایای استفاده از لینوکس
- ۱۵۳ - استانداردهای مربوط به لینوکس
- ۱۵۴ - آینده لینوکس
- ۱۵۵ - انجمن
- ۱۵۷ - لینوکس و فایل
- ۱۵۸ - اشیاء به عنوان توصیف کننده های فایل

فصل ۴ - برنامه نویسی در لینوکس

- ۱۶۴ - پشتیبانی سخت افزاری

فصل ۵ - سهم بازار و جذب

- ۱۶۷ - وب سرورها W3Cook
- ۱۶۷ - مزایای لینوکس
- ۱۶۸ - لینوکس در دنیای فناوری های نوین
- ۱۶۸ - آینده لینوکس
- ۱۶۸ - دستگاه های موبایل
- ۱۶۹ - تولید فیلم و نقش لینوکس
- ۱۶۹ - استفاده در دولت
- ۱۷۱ - حق چاپ، علامت تجاری، و نامگذاری
- ۱۷۲ - نام لینوکس

مقدمه

لینوکس، فراتر از یک سیستم‌عامل، نمادی از فلسفه نرم‌افزار آزاد و متن‌باز است که تأکید بر آزادی، شفافیت و همکاری جهانی دارد. این سیستم‌عامل که در اوایل دهه ۱۹۹۰ توسط لینوس توروالدز توسعه یافت، به سرعت به یکی از پرکاربردترین و تأثیرگذارترین سیستم‌عامل‌ها در جهان تبدیل شد. فلسفه لینوکس بر این اصل استوار است که نرم‌افزار باید آزادانه در دسترس باشد تا کاربران بتوانند آن را مطالعه، تغییر و توزیع کنند. این رویکرد، برخلاف مدل‌های نرم‌افزار مالکانه، به کاربران امکان می‌دهد تا به‌طور کامل بر نرم‌افزارهای خود کنترل داشته باشند و آن‌ها را بر اساس نیازهای خود سفارشی‌سازی کنند. ریچارد استالمن، بنیان‌گذار پروژه گنو و از پیشگامان جنبش نرم‌افزار آزاد، نقش بسزایی در شکل‌گیری این فلسفه داشت. او معتقد بود که محدودیت‌های اعمال‌شده توسط شرکت‌های نرم‌افزاری، مانعی برای پیشرفت و نوآوری است و با ایجاد نرم‌افزارهای آزاد، می‌توان این موانع را از میان برداشت. پروژه گنو با هدف ایجاد یک سیستم‌عامل کاملاً آزاد آغاز شد و هسته لینوکس که توسط توروالدز توسعه یافت، به‌عنوان تکمیل‌کننده این پروژه، منجر به تولد سیستم‌عامل گنو/لینوکس شد.

یکی از اصول اساسی فلسفه لینوکس، تأکید بر همکاری و مشارکت جهانی است. توسعه‌دهندگان از سراسر جهان می‌توانند در بهبود و توسعه این سیستم‌عامل مشارکت کنند و ایده‌ها و کدهای خود را با دیگران به اشتراک بگذارند. این مدل توسعه باز و مشارکتی، منجر به ایجاد یک جامعه پویا و فعال شده است که به‌طور مداوم در حال بهبود و ارتقاء لینوکس است. همچنین، این فلسفه به کاربران امکان می‌دهد تا با مطالعه کد منبع، درک عمیق‌تری از عملکرد نرم‌افزارها پیدا کنند و در صورت نیاز، آن‌ها را مطابق با نیازهای خود تغییر دهند. فلسفه لینوکس همچنین به دنبال حذف مرزها و محدودیت‌های جغرافیایی است. در دنیای لینوکس، یک توسعه‌دهنده از هر نقطه‌ای از جهان می‌تواند با دیگران همکاری کند و دانش و تجربیات خود را به اشتراک بگذارد. این رویکرد جهانی، منجر به تنوع و غنای بیشتر در توسعه نرم‌افزارها شده و به کاربران از فرهنگ‌ها و زبان‌های مختلف امکان می‌دهد تا در این جامعه جهانی سهیم باشند. در نهایت، فلسفه لینوکس بر این باور است که نرم‌افزار باید به‌عنوان یک دارایی مشترک انسانی در نظر گرفته شود که همه می‌توانند از آن بهره‌مند شوند و در توسعه آن مشارکت کنند. این دیدگاه، با تأکید بر آزادی، شفافیت و همکاری، به ایجاد یک اکوسیستم نرم‌افزاری منجر شده است که در آن نوآوری و پیشرفت به‌طور مداوم ادامه دارد و کاربران و توسعه‌دهندگان می‌توانند با هم در جهت بهبود و ارتقاء نرم‌افزارها تلاش کنند. یکی از ویژگی‌های برجسته فلسفه لینوکس، تأکید بر اصل «انجام یک کار و انجام آن به‌خوبی» است. این اصل که از فلسفه یونیکس نشأت می‌گیرد، توسعه‌دهندگان را تشویق می‌کند تا برنامه‌هایی بنویسند که وظایف خاصی را به‌طور مؤثر انجام دهند و در صورت نیاز، با دیگر برنامه‌ها ترکیب شوند. این رویکرد، سادگی و کارایی را ترویج می‌دهد و منجر به ایجاد ابزارهای کوچک و قابل ترکیب می‌شود که می‌توانند برای انجام وظایف پیچیده‌تر با یکدیگر همکاری کنند. فلسفه لینوکس همچنین بر اهمیت شفافیت و قابلیت بررسی تأکید دارد. با دسترسی آزاد به کد منبع، کاربران و توسعه‌دهندگان می‌توانند به‌طور مستقل امنیت و کارایی نرم‌افزار را ارزیابی کنند. این شفافیت، اعتماد به نرم‌افزار را افزایش می‌دهد و امکان شناسایی و رفع سریع‌تر اشکالات و آسیب‌پذیری‌ها را فراهم می‌کند. در مقابل، در نرم‌افزارهای مالکانه، کاربران به کد منبع دسترسی ندارند و باید به تولیدکنندگان نرم‌افزار اعتماد کنند، بدون اینکه بتوانند به‌طور مستقل صحت و امنیت آن را بررسی کنند. یکی دیگر از جنبه‌های مهم فلسفه لینوکس، تأکید بر

آموزش و یادگیری است. با دسترسی به کد منبع و امکان مطالعه و تغییر آن، کاربران می‌توانند مهارت‌های برنامه‌نویسی و درک خود از سیستم‌های کامپیوتری را بهبود بخشند. این رویکرد، به‌ویژه برای دانشجویان و علاقه‌مندان به فناوری مفید است، زیرا به آن‌ها امکان می‌دهد تا با ساختار داخلی نرم‌افزارها آشنا شوند و تجربه عملی کسب کنند. فلسفه لینوکس همچنین به دنبال ایجاد یک اکوسیستم پایدار و قابل اعتماد است. با مشارکت جامعه جهانی در توسعه و نگهداری نرم‌افزار، لینوکس توانسته است به یک سیستم‌عامل پایدار و امن تبدیل شود که در محیط‌های مختلف، از سرورها و ابررایانه‌ها گرفته تا دستگاه‌های تعبیه‌شده و تلفن‌های همراه، مورد استفاده قرار می‌گیرد. این پایداری و انعطاف‌پذیری، نتیجه مستقیم مدل توسعه باز و مشارکتی است که در فلسفه لینوکس مورد تأکید قرار گرفته است.

لینوکس به‌عنوان یکی از برجسته‌ترین سیستم‌عامل‌های جهان امروز، نه تنها از نظر فنی بلکه از منظر فلسفی نیز یک انقلاب در دنیای نرم‌افزار ایجاد کرده است. این سیستم‌عامل که در ابتدا توسط لینوس توروالدز در سال ۱۹۹۱ به‌عنوان یک پروژه متن‌باز آغاز شد، به سرعت جایگاه خود را در دنیای فناوری‌های اطلاعات و ارتباطات پیدا کرد و به یکی از پرکاربردترین سیستم‌عامل‌ها در محیط‌های مختلف از سرورها گرفته تا گوشی‌های هوشمند و دستگاه‌های اینترنت اشیا تبدیل شد. اما آنچه لینوکس را به یک پدیده منحصر به فرد تبدیل کرده است، تنها تکنولوژی پیچیده و قابلیت‌های فنی آن نیست، بلکه فلسفه و اصول بنیادی آن در زمینه نرم‌افزار آزاد و متن‌باز است که بنیاد سیستم‌عامل لینوکس را شکل داده‌اند. فلسفه لینوکس، که در حقیقت برگرفته از فلسفه جنبش نرم‌افزار آزاد است، بر مبنای اصولی همچون آزادی در استفاده، تغییر، توزیع و مطالعه نرم‌افزارها بنا شده است. این فلسفه، که به‌ویژه از طرف ریچارد استالمن و بنیاد نرم‌افزار آزاد (FSF) معرفی و ترویج شد، به دنبال مقابله با محدودیت‌هایی است که شرکت‌ها و سازمان‌های تجاری برای نرم‌افزارهای خود اعمال می‌کنند. در واقع، فلسفه نرم‌افزار آزاد این امکان را به کاربران می‌دهد تا به جای خریدن نرم‌افزارهای بسته و محدود، از نرم‌افزارهای آزاد و قابل دسترسی استفاده کنند و حتی آن‌ها را به‌طور آزادانه تغییر دهند و بهبود بخشند. در این راستا، لینوکس نه تنها به‌عنوان یک سیستم‌عامل بلکه به‌عنوان یک نماد برای آزادی در دنیای دیجیتال شناخته می‌شود.

در آغاز، لینوکس تنها یک هسته نرم‌افزاری بود که توسط لینوس توروالدز به‌طور مستقل توسعه داده شد. توروالدز که دانشجوی دانشگاه هلسینکی در فنلاند بود، تصمیم گرفت که یک هسته سیستم‌عامل جدید بسازد که قابلیت اجرا بر روی سخت‌افزارهای مختلف را داشته باشد. اما آنچه که این هسته را از دیگر هسته‌ها متمایز کرد، تصمیم او به انتشار آن به‌طور متن‌باز بود. این تصمیم به توسعه‌دهندگان نرم‌افزارهای دیگر در سراسر جهان این امکان را داد تا به‌طور آزادانه کدهای هسته را بررسی کرده، آن‌ها را بهبود دهند و نسخه‌های جدیدی از آن بسازند. این اقدام، که با استقبال گسترده‌ای از سوی جامعه برنامه‌نویسان روبرو شد، نقطه آغازین رشد سریع لینوکس بود. یکی از اصول اساسی فلسفه لینوکس این است که "نرم‌افزار باید آزاد باشد" تا کاربران بتوانند بدون هیچ محدودیتی از آن استفاده کنند. این به معنای آزادی در استفاده، مطالعه، اصلاح و توزیع نرم‌افزار است. این رویکرد به‌طور قابل‌ملاحظه‌ای با مدل‌های تجاری موجود در صنعت نرم‌افزار که بر اساس فروش نسخه‌های بسته‌بندی شده و با مجوزهای محدود است، تفاوت دارد. در این مدل‌های سنتی، کاربران حق تغییر کد منبع نرم‌افزار یا توزیع آن را ندارند، در حالی که در

دنیای لینوکس و نرم‌افزارهای آزاد، این آزادی‌ها جزء لاینفک از تجربه کاربری هستند. این آزادی‌ها نه تنها برای کاربران نهایی اهمیت دارند، بلکه برای توسعه‌دهندگان نیز فرصت‌هایی فراهم می‌آورند تا بر روی پروژه‌های بزرگ و پیچیده کار کنند. در سیستم‌های نرم‌افزاری معمولی، اگر یک توسعه‌دهنده نیاز به تغییر در نرم‌افزاری داشته باشد، باید با محدودیت‌های مختلفی از جمله دسترسی به کد منبع و حقوق مالکیت معنوی مواجه شود. اما در دنیای لینوکس، کد منبع به طور کامل در دسترس است و هر کسی می‌تواند آن را تغییر دهد، بهبود بخشد و به سایرین ارائه دهد. این فرهنگ به‌ویژه در جوامع متن‌باز مشهود است که در آن همکاری و اشتراک‌گذاری دانش به عنوان یکی از ارکان اصلی فلسفه لینوکس شناخته می‌شود. یکی از مهم‌ترین ویژگی‌های لینوکس این است که برخلاف سیستم‌عامل‌های تجاری، کاملاً رایگان و قابل توزیع است. این ویژگی به‌ویژه برای کشورهایی که به دنبال کاهش هزینه‌های زیرساختی در حوزه فناوری اطلاعات هستند، جذابیت دارد. لینوکس به عنوان یک سیستم‌عامل مبتنی بر اصول نرم‌افزار آزاد، به سازمان‌ها و نهادها این امکان را می‌دهد تا بدون نگرانی از پرداخت هزینه‌های لایسنس نرم‌افزار، از یک سیستم‌عامل قدرتمند و پایدار استفاده کنند. علاوه بر این، در صورت نیاز به هرگونه تغییر یا پیکربندی خاص، سازمان‌ها می‌توانند به راحتی به کد منبع دسترسی پیدا کنند و آن را مطابق با نیازهای خود تغییر دهند. فلسفه لینوکس همچنین تأکید زیادی بر شفافیت و همکاری دارد. در یک پروژه لینوکس، هر کسی می‌تواند به کد منبع دسترسی پیدا کند و در روند توسعه آن مشارکت داشته باشد. این باز بودن کدها و شفافیت در توسعه باعث می‌شود که مشکلات امنیتی سریع‌تر شناسایی و برطرف شوند و به طور کلی، کیفیت نرم‌افزار افزایش یابد. یکی از مهم‌ترین ویژگی‌های این شفافیت این است که کاربران و توسعه‌دهندگان می‌توانند به طور کامل به فرآیند توسعه سیستم‌عامل و نرم‌افزارهای مرتبط با آن دسترسی داشته باشند، و از این طریق در فرآیند تصمیم‌گیری‌ها و بهبودها مشارکت کنند.

با وجود این ویژگی‌های مثبت، فلسفه لینوکس همواره با چالش‌ها و نقدهایی مواجه بوده است. یکی از چالش‌های اصلی، عدم هماهنگی و یکپارچگی در توزیع‌های مختلف لینوکس است. به طور خاص، وجود توزیع‌های متعدد و متفاوت لینوکس ممکن است برای کاربران تازه‌کار یا کسانی که به دنبال استفاده از سیستم‌عامل برای کاربردهای خاص هستند، سردرگم‌کننده باشد. این تنوع در توزیع‌ها می‌تواند مزایای خاصی مانند تطبیق‌پذیری و انعطاف‌پذیری را به همراه داشته باشد، اما از سوی دیگر، باعث بروز مشکلاتی در خصوص پشتیبانی و هماهنگی بین توزیع‌ها می‌شود. علی‌رغم این چالش‌ها، لینوکس همچنان به عنوان یکی از موفق‌ترین سیستم‌عامل‌ها در دنیای امروز شناخته می‌شود. این سیستم‌عامل، که بر پایه اصولی همچون آزادی، شفافیت، همکاری و دسترسی به کد منبع ساخته شده است، نه تنها به عنوان یک ابزار نرم‌افزاری بلکه به عنوان یک جنبش فرهنگی و فلسفی در دنیای فناوری شناخته می‌شود. فلسفه لینوکس همچنان برای میلیون‌ها کاربر و توسعه‌دهنده در سرتاسر جهان الهام‌بخش است و به آنها این امکان را می‌دهد تا در ایجاد دنیای دیجیتال آزادتر و برابرتر سهم داشته باشند. فلسفه لینوکس بر چهار آزادی اساسی استوار است: آزادی اجرای برنامه برای هر هدفی، آزادی مطالعه و بررسی نحوه عملکرد برنامه، آزادی تغییر برنامه برای تطبیق با نیازهای شخصی، و آزادی توزیع نسخه‌های اصلاح‌شده برای کمک به جامعه. این چهار اصل، که توسط بنیاد نرم‌افزار آزاد (FSF) تعریف شده‌اند، نه تنها حقوق کاربران را تضمین می‌کنند، بلکه تعهدی اخلاقی برای توسعه‌دهندگان ایجاد می‌نمایند تا دانش خود را به عنوان میراثی مشترک به نسل‌های بعدی منتقل کنند. در

این چارچوب، هر خط کد نوشته شده تنها یک دستور رایانه‌ای نیست، بلکه بیانیه‌ای است در حمایت از شفافیت و اعتماد. یکی از جذاب‌ترین جنبه‌های فلسفه لینوکس، رد مفهوم سنتی «مالکیت» در تولید نرم‌افزار است. در دنیای لینوکس، هیچ‌کس نمی‌تواند ادعا کند که صاحب تمامی سیستم است. حتی لینوس توروالدز، با وجود نقش محوری‌اش، تنها به عنوان یک هماهنگ‌کننده جامعه توسعه‌دهندگان عمل می‌کند. این الگوی غیرمتمرکز، قدرت را از دست شرکت‌ها و افراد خاص خارج کرده و آن را میان میلیون‌ها کاربر و توسعه‌دهنده داوطلب در سراسر جهان توزیع می‌کند. چنین مدلی نه تنها از نظر فنی پایدارتر است، بلکه نمادی از یک دموکراسی واقعی در عصر اطلاعات محسوب می‌شود. جامعه لینوکس نمونه بی‌بدیلی از همکاری بین‌المللی است. توسعه‌دهندگانی از فرهنگ‌ها، زبان‌ها، و باورهای سیاسی متفاوت، بدون هیچ سلسله‌مراتب تحمیلی، کنار هم کار می‌کنند تا سیستمی بسازند که متعلق به همه است. این هماهنگی عظیم، که بدون رهبر مطلق و تنها با اتکا به شایسته‌سالاری و احترام متقابل پیش می‌رود، پاسخی عملی به این پرسش فلسفی است: «آیا انسان‌ها می‌توانند فارغ از منافع شخصی، برای هدفی بزرگ‌تر همکاری کنند؟» لینوکس ثابت کرده است که پاسخ مثبت است. مدل توسعه لینوکس، که به «روش بزار» (Bazaar) معروف است، در تقابل کامل با شیوه سنتی «کلیسای جامع» قرار دارد. در حالی که شرکت‌های انحصاری مانند مایکروسافت یا اپل نرم‌افزارهای خود را پشت دیوارهای بلند محرمانگی توسعه می‌دهند، لینوکس در بازاری باز رشد می‌کند که هر کسی می‌تواند محصول را بررسی، نقد، و بهبود بخشد. این شفافیتِ رادیکال، با وجود چالش‌هایش، منجر به ایجاد سیستمی شده که از نظر امنیت و پایداری بی‌نظیر است. فلسفه پشت این مدل ساده است: «چشم‌های زیاد، باگ‌ها را کم‌عمق می‌کنند».

لینوکس به ما می‌آموزد که کمال‌گرایی دشمن پیشرفت است. برخلاف سیستم‌های اختصاصی که سال‌ها در انتظار انتشار نسخه «بی‌نقص» می‌مانند، لینوکس به صورت تدریجی و از طریق انتشار مکرر نسخه‌های آزمایشی بهبود می‌یابد. این رویکرد تکاملی، که از نظریه داروین الهام گرفته، نشان می‌دهد که چگونه سیستم‌های پیچیده می‌توانند از طریق آزمون و خطا و مشارکت جمعی به بلوغ برسند. هر خطای کشف‌شده نه یک شکست، بلکه فرصتی برای یادگیری است.

فصل یکم

فصل ۱

تاریخچه

کن تامپسون و دنیس ریچی: بنیانگذاران فلسفه یونیکس و تأثیر آن بر دنیای نرم افزار

کن تامپسون و دنیس ریچی، دو چهره برجسته در تاریخ علوم کامپیوتر، نه تنها به عنوان توسعه دهندگان سیستم عامل یونیکس شناخته می شوند، بلکه به عنوان طرفداران کلیدی فلسفه ای که زیربنای این سیستم عامل است نیز شهرت دارند. فلسفه یونیکس، که توسط تامپسون و ریچی شکل گرفت، مجموعه ای از اصول فرهنگی و رویکردهای فلسفی است که بر توسعه نرم افزارهای مینیمالیستی و مدولار تأکید دارد. این فلسفه نه تنها بر فناوری یونیکس تأثیر گذاشت، بلکه به عنوان یک چارچوب فکری برای توسعه نرم افزار در سراسر جهان شناخته شد.

ریشه های فلسفه یونیکس

فلسفه یونیکس از تجربیات توسعه دهندگان پیشروی این سیستم عامل نشأت گرفت. در اواخر دهه ۱۹۶۰ و اوایل دهه ۱۹۷۰، تامپسون و ریچی در آزمایشگاه های بل مشغول کار بر روی سیستم عامل یونیکس بودند. در آن زمان، سیستم های عامل عموماً پیچیده و یکپارچه بودند و توسعه و نگهداری آنها دشوار بود. تامپسون و ریچی با رویکردی متفاوت، به دنبال ایجاد سیستمی بودند که ساده، قابل درک و انعطاف پذیر باشد. این رویکرد منجر به ایجاد فلسفه یونیکس شد که بر مینیمالیسم، مدولاریته و قابلیت استفاده مجدد تأکید داشت. کن تامپسون و دنیس ریچی از پیشگامان برجسته علوم رایانه هستند که با همکاری یکدیگر، سیستم عامل یونیکس را توسعه دادند و تأثیر عمیقی بر دنیای فناوری اطلاعات گذاشتند.

کن تامپسون

در ۴ فوریه ۱۹۴۳ در نیواورلئان، لوئیزیانا، ایالات متحده آمریکا متولد شد. او از کودکی علاقه مند به ریاضیات و الگوریتم ها بود و در دوران تحصیل، مسائل ریاضی را با استفاده از معادلات باینری حل می کرد. تامپسون در دانشگاه کالیفرنیا، برکلی تحصیل کرد و در سال ۱۹۶۶ مدرک کارشناسی ارشد خود را در رشته مهندسی برق و علوم رایانه دریافت نمود.

دنیس ریچی

در ۹ سپتامبر ۱۹۴۱ در برانکسویل، نیویورک، ایالات متحده آمریکا به دنیا آمد. او در سال ۱۹۶۳ مدرک کارشناسی خود را در رشته فیزیک از دانشگاه هاروارد دریافت کرد و سپس در رشته ریاضیات کاربردی ادامه تحصیل داد. ریچی در سال ۱۹۶۷ به مرکز تحقیقات علمی و محاسباتی آزمایشگاه های بل پیوست و در آنجا با کن تامپسون آشنا شد. در اواخر دهه ۱۹۶۰، تامپسون و ریچی در آزمایشگاه های بل روی پروژه سیستم عامل Multics کار می کردند. پس از پایان این پروژه، آنها تصمیم گرفتند سیستم عاملی ساده تر و کارآمدتر ایجاد کنند. نتیجه این تلاش ها، توسعه سیستم عامل یونیکس در سال ۱۹۶۹ بود که به دلیل سادگی، انعطاف پذیری و قابلیت حمل، به سرعت مورد توجه

پیچیده‌تری را انجام دهند. این در حالی است که در سیستم‌های دیگر، چنین عملکردهایی اغلب در یک برنامه واحد با ساختار پیچیده گنجانده می‌شدند. به عنوان مثال، در سیستم‌های دیگر، ممکن است یک برنامه واحد با نام "**سیستم فایل**" وجود داشته باشد که تمام عملکردهای مرتبط با فایل‌ها، مانند لیست کردن، حذف، تغییر نام و کپی کردن، را در خود جای دهد. این رویکرد، اگرچه ممکن است به ظاهر کاربردی باشد، اما باعث می‌شود که برنامه‌ها بزرگ و پیچیده شوند و استفاده از آن‌ها برای کاربران دشوارتر شود. در مقابل، فلسفه یونیکس بر این اصل استوار است که هر برنامه باید کوچک، ساده و مستقل باشد تا بتواند به راحتی با سایر برنامه‌ها ترکیب شود.

فلسفه یونیکس: سادگی، ترکیب‌پذیری و مینیمالیسم

داگ مک‌ایلروی (Doug McIlroy)، رئیس سابق مرکز تحقیقات علوم محاسباتی بل (Bell Labs) و یکی از چهره‌های کلیدی در توسعه سیستم عامل یونیکس، به عنوان یکی از معماران اصلی فلسفه یونیکس شناخته می‌شود. او نه تنها مخترع **لوله‌های یونیکس (Unix pipes)** بود، بلکه نقش مهمی در شکل‌گیری و ترویج اصولی داشت که امروزه به عنوان فلسفه یونیکس شناخته می‌شوند. مک‌ایلروی در سخنان خود، این فلسفه را به صورت خلاصه و گویا بیان کرده است: "**برنامه‌هایی بنویسید که یک کار را انجام دهند و آن را به خوبی انجام دهند. برای کار با هم برنامه بنویسید. برنامه‌هایی بنویسید تا جریان متن را مدیریت کنند، زیرا این یک رابط جهانی است**". این جملات کوتاه، اما عمیق، نه تنها اصول اصلی یونیکس را در بر می‌گیرند، بلکه راهنمایی ارزشمند برای توسعه‌دهندگان نرم‌افزار در سراسر جهان هستند.

سادگی و مینیمالیسم: قلب فلسفه یونیکس

مک‌ایلروی در کنار تأکید بر ترکیب‌پذیری و مدیریت جریان متن، بر **سادگی و مینیمالیسم** در برنامه‌نویسی یونیکس نیز تأکید کرده است. او معتقد بود که برنامه‌ها باید کوچک، ساده و متمرکز باشند و از افزودن پیچیدگی‌های غیرضروری اجتناب کنند. به گفته او، "**مفهوم پیچیدگی‌های پیچیده و زیبا تقریباً عجیب است**". در دنیای یونیکس، برنامه‌نویسان به جای تلاش برای ایجاد برنامه‌های بزرگ و پر از ویژگی‌های مختلف، برای ایجاد برنامه‌های "**ساده و زیبا**" با یکدیگر رقابت می‌کنند. این رویکرد نه تنها باعث می‌شود برنامه‌ها قابل درک و نگهداری باشند، بلکه استفاده از آن‌ها را نیز آسان‌تر می‌کند.

مک‌ایلروی در این زمینه به یک نکته مهم اشاره می‌کند: "**همه چیز کوچک بود... و وقتی اندازه لینوکس را می‌بینم، دلم برای لینوکس غرق می‌شود**". او با این جمله به تغییراتی که در سیستم‌عامل‌های مدرن مانند لینوکس رخ داده است، اشاره می‌کند. در گذشته، برنامه‌ها و دستورات یونیکس کوچک و ساده بودند، اما امروزه بسیاری از آن‌ها با هزاران گزینه و ویژگی اضافی همراه شده‌اند. به عنوان مثال، **صفحه‌های دستی (man pages)** که قبلاً مختصر و مفید بودند، اکنون حجم زیادی دارند و پر از گزینه‌های پیچیده هستند. مک‌ایلروی این روند را ناشی از طراحی نادرست می‌داند و معتقد است که افزودن گزینه‌های زیاد اغلب نشان‌دهنده این است که برنامه‌نویس به نقطه طراحی درستی نرسیده است.

زندگی و پیشینه

اریک استیون ریموند در ۴ دسامبر ۱۹۵۷ در بوستون، ماساچوست به دنیا آمد. او از کودکی به علوم کامپیوتر و برنامه‌نویسی علاقه‌مند بود و در نوجوانی شروع به یادگیری برنامه‌نویسی کرد. ریموند در دانشگاه پنسیلوانیا تحصیل کرد، اما تحصیلات رسمی خود را ناتمام گذاشت و به جای آن، به صورت خودآموز به یادگیری و فعالیت در حوزه‌های مختلف فناوری پرداخت. ریموند یکی از پیشگامان جنبش نرم‌افزارهای متن‌باز است و نقش مهمی در ترویج این فلسفه داشته است. برخی از مشارکت‌های مهم او عبارتند از:

۱. نگهداری از پروژه: fetchmail	ریموند توسعه‌دهنده اصلی پروژه fetchmail است، یک ابزار متن‌باز برای دریافت ایمیل از سرورهای راه‌دور. این پروژه به عنوان نمونه‌ای از توسعه نرم‌افزارهای متن‌باز در کتاب معروف او، «کلیسای جامع و بازار» (The Cathedral and the Bazaar)، مورد بررسی قرار گرفته است.
۲. نقش در شکل‌گیری جنبش متن‌باز:	ریموند یکی از بنیان‌گذاران ابتکار متن‌باز (Open Source Initiative) یا (OSI) بود که در سال ۱۹۹۸ تأسیس شد. این سازمان به ترویج و حمایت از نرم‌افزارهای متن‌باز پرداخت و اصطلاح «متن‌باز» (Open Source) «را به طور رسمی رایج کرد.
۳. همکاری با پروژه‌های بزرگ:	او در توسعه و بهبود پروژه‌های متن‌باز بزرگی مانند هسته لینوکس و ابزارهای گنو مشارکت داشته است.

آثار و نوشته‌ها

ریموند به عنوان یک نویسنده، آثار متعددی در مورد فرهنگ هکرها، توسعه نرم‌افزار، و فلسفه متن‌باز منتشر کرده است. برخی از مهم‌ترین آثار او عبارتند از:

۱. کلیسای جامع و بازار: (The Cathedral and the Bazaar)	این کتاب یکی از تأثیرگذارترین آثار در دنیای نرم‌افزارهای متن‌باز است. ریموند در این کتاب، مدل‌های توسعه نرم‌افزار را مقایسه می‌کند: مدل «کلیسای جامع» (توسعه متمرکز و بسته) در مقابل مدل «بازار» (توسعه باز و مشارکتی). او استدلال می‌کند که مدل بازار، به دلیل مشارکت گسترده و باز بودن، کارآمدتر و انعطاف‌پذیرتر است.
۲. هنر برنامه‌نویسی یونیکس: (The Art of Unix Programming)	

اشتباهات فاجعه‌بار در یونیکس

نورمن معتقد بود که یکی از مهم‌ترین چالش‌های رابط کاربری یونیکس این است که اشتباهات کاربران می‌توانند پیامدهای فاجعه‌باری داشته باشند. این موضوع برخلاف سیستم‌های کاربرمحور است که به گونه‌ای طراحی می‌شوند که اشتباهات کاربر را کاهش دهند و در صورت بروز خطا، امکان بازگردانی را فراهم کنند. او برای مثال به عدم وجود سیستم تأیید یا هشدار مناسب در یونیکس اشاره کرد. بسیاری از دستورات قدرتمند، بدون تأیید اضافی اجرا می‌شوند و می‌توانند داده‌های مهم را به صورت غیرقابل بازگشت حذف کنند. این موضوع نشان‌دهنده بی‌توجهی به اصول طراحی شناختی است که پیشنهاد می‌کند سیستم‌ها باید بازخورد مناسب ارائه دهند و کاربران را از عواقب تصمیماتشان آگاه کنند.

مشکلات رابط کاربری یونیکس

نورمن در مقاله خود به چندین مشکل کلیدی در رابط کاربری یونیکس اشاره کرد:

۱. **عدم وجود بازخورد مناسب:** یونیکس اغلب بازخورد کافی به کاربران نمی‌دهد. به عنوان مثال، اگر کاربر یک دستور اشتباه وارد کند، سیستم ممکن است بدون هیچ توضیحی از اجرای دستور خودداری کند یا حتی بدتر، دستور را به طور نادرست اجرا کند. این عدم بازخورد باعث می‌شود که کاربران نتوانند به راحتی اشتباهات خود را تشخیص دهند و اصلاح کنند.
۲. **پیچیدگی دستورات:** دستورات یونیکس اغلب کوتاه و مختصر هستند، اما این کوتاهی باعث می‌شود که دستورات برای کاربران عادی غیرقابل درک باشند. به عنوان مثال، دستور `rm -rf` ممکن است برای یک برنامه‌نویس قابل درک باشد، اما برای یک کاربر عادی می‌تواند بسیار گیج‌کننده و خطرناک باشد.
۳. **عدم وجود رابط گرافیکی:** در زمان نوشتن مقاله نورمن، یونیکس فاقد یک رابط گرافیکی کاربرپسند بود. کاربران مجبور بودند تمام دستورات را به صورت متنی وارد کنند، که این امر باعث افزایش احتمال خطا و سردرگمی می‌شد.
۴. **اشتباهات فاجعه‌بار:** نورمن به این موضوع اشاره کرد که کاربران یونیکس به راحتی می‌توانند اشتباهات فاجعه‌باری مرتکب شوند، مانند حذف تصادفی فایل‌های مهم یا از دست دادن ساعت‌ها کار به دلیل یک دستور اشتباه. این اشتباهات نه تنها باعث ناامیدی کاربران می‌شود، بلکه می‌تواند عواقب جدی نیز داشته باشد.

نیاز به طراحی کاربرمحور

انتقادات دان نورمن از رابط کاربری یونیکس نشان‌دهنده این است که فلسفه طراحی یونیکس، اگرچه از نظر فنی بسیار قدرتمند و انعطاف‌پذیر است، اما به نیازها و انتظارات کاربران عادی توجه کافی نداشته است. نورمن استدلال کرد که طراحی سیستم‌ها باید بر پایه **مدل‌های شناختی کاربران** باشد و به آن‌ها کمک کند که به راحتی با سیستم تعامل داشته باشند. در دنیای امروز، که **تجربه کاربری (User Experience)** به عنوان یکی از عوامل

می کردند، اکنون خود به پشتیبانان آن تبدیل شده اند. برای مثال، **مایکروسافت** که زمانی لینوکس را دشمن خود می دانست، اکنون در توسعه پروژه های متن باز مشارکت می کند و حتی **Azure**، سرویس ابری خود را بر اساس لینوکس اجرا می کند. همچنین، شرکت هایی مانند **گوگل**، **آمازون** و **فیسبوک** به طور گسترده از لینوکس در زیرساخت های خود استفاده می کنند و حتی خود در توسعه آن نقش دارند.

گسترش فلسفه یونیکس در عصر اینترنت

فلسفه یونیکس نه تنها در توسعه سیستم عامل ها، بلکه در شکل گیری و رشد سریع **اینترنت** نیز نقشی اساسی داشته است. در دهه های ۱۹۷۰ و ۱۹۸۰، زمانی که شبکه های کامپیوتری در مراحل اولیه خود بودند، یونیکس با ارائه یک **سیستم عامل پایدار، قابل اعتماد و ماژولار**، به عنوان بستر اصلی بسیاری از این شبکه ها مورد استفاده قرار گرفت. بسیاری از مفاهیم کلیدی که امروزه در اینترنت استفاده می شوند، مانند **پروتکل TCP/IP**، **سرورهای وب و ابزارهای مدیریت شبکه**، ابتدا در محیط یونیکس توسعه یافتند. در واقع، یکی از دلایلی که اینترنت توانست با این سرعت رشد کند، همین زیرساخت قوی و انعطاف پذیری بود که یونیکس فراهم کرد. یکی از ویژگی های کلیدی یونیکس که به رشد اینترنت کمک کرد، **ماهیت چندکاربره و چندوظیفه ای** آن بود. این قابلیت باعث شد که رایانه ها بتوانند همزمان چندین اتصال شبکه ای را مدیریت کرده و منابع خود را بین کاربران مختلف به اشتراک بگذارند. در دوران اولیه اینترنت، اکثر **سرورهای ایمیل**، **پایگاه های داده و سرویس های وب** روی سیستم های یونیکسی اجرا می شدند، زیرا این سیستم عامل به طور ذاتی برای پردازش های همزمان و مدیریت کارآمد شبکه طراحی شده بود.

یکی دیگر از جنبه های مهم یونیکس، **ماژولار بودن** آن است. این ویژگی امکان می دهد که اجزای مختلف سیستم بدون نیاز به تغییر کلی آن، به روزرسانی یا جایگزین شوند. این مفهوم در اینترنت نیز بسیار کاربردی بوده است، زیرا شبکه های کامپیوتری دائماً در حال تغییر و توسعه هستند. برای مثال، وقتی یک پروتکل شبکه جدید مانند **IPv۶** معرفی شد، نیازی به بازنویسی کامل سیستم های یونیکسی نبود؛ بلکه تنها لایه های مرتبط با شبکه به روزرسانی شدند، بدون آنکه سایر اجزای سیستم تحت تأثیر قرار بگیرند.

همچنین، **پروتکل های اساسی اینترنت، مانند TCP/IP**، ابتدا در محیط یونیکس توسعه یافتند. در دهه ۱۹۸۰، دانشگاه کالیفرنیا، برکلی (UCB) مجموعه ای از ابزارهای شبکه را به همراه **پروتکل TCP/IP** در نسخه های **BSD یونیکس** پیاده سازی کرد. این کار باعث شد که یونیکس به **اولین سیستم عاملی** تبدیل شود که به صورت بومی از اینترنت پشتیبانی می کند. بعدها، این پروتکل ها به استاندارد جهانی تبدیل شدند و در سیستم عامل های دیگر نیز مورد استفاده قرار گرفتند، اما نقش یونیکس در این توسعه غیرقابل انکار است.

یکی دیگر از تأثیرات مهم یونیکس بر اینترنت، **نرم افزارهای سروری قدرتمند و متن باز** بود که روی این سیستم عامل توسعه یافتند. برای مثال، **وب سرور Apache**، که امروزه یکی از پرکاربردترین وب سرورها در اینترنت است، روی سیستم های یونیکسی توسعه یافت و از همان ابتدا بر پایه اصول سادگی، انعطاف پذیری و کارایی یونیکس طراحی شد. به همین ترتیب، بسیاری از پایگاه های داده قدرتمند مانند **MySQL و PostgreSQL**

در مورد لینوکس، فلسفه آن به طور عمده بر آزادی، شفافیت، و کنترل کامل کاربر بر سیستم عامل خود متمرکز است. این فلسفه به کاربران و توسعه دهندگان این امکان را می دهد که سیستم عامل را به طور کامل شخصی سازی کنند، ویژگی های آن را تغییر دهند و حتی کد منبع آن را مشاهده و اصلاح کنند. برخی از اصول کلیدی فلسفه لینوکس شامل موارد زیر است:

۱. **آزادی نرم افزاری: (Open Source)** لینوکس یک نرم افزار آزاد است، به این معنا که کاربران می توانند به کد منبع آن دسترسی پیدا کرده و آن را تغییر دهند. این امر به توسعه دهندگان و سازمان ها این امکان را می دهد که نیازهای خاص خود را برآورده کنند.
۲. **ساده سازی و کم حجم بودن:** لینوکس فلسفه ای دارد که به دنبال ساده سازی سیستم ها و کاهش پیچیدگی است. این به معنای داشتن ابزارهایی است که دقیقاً آنچه را که برای انجام کارهای خاص نیاز دارید، فراهم می کنند.
۳. **مدیریت منابع:** در فلسفه لینوکس، کارایی و استفاده بهینه از منابع سیستم بسیار مهم است. سیستم عامل های مبتنی بر لینوکس به طور معمول منابع کمتری مصرف می کنند، که این موضوع آن را برای دستگاه های قدیمی تر و یا سیستم های نیازمند بهینه سازی بالا مناسب می کند.
۴. **جامعه محور بودن:** لینوکس بر پایه مشارکت جامعه ای از توسعه دهندگان، متخصصان و کاربران استوار است. این فلسفه باعث شده که پروژه های لینوکس به طور مستمر بهبود یابند و ابداعات جدیدی ایجاد شود.

این فلسفه ها به طور عمیقی بر توسعه نرم افزار، امنیت، و حتی نحوه تعامل افراد با دنیای دیجیتال تأثیر گذاشته است. آزادی های ارائه شده توسط لینوکس باعث شده که بسیاری از شرکت ها و افراد به جای استفاده از سیستم های تجاری، به استفاده از این سیستم عامل روی بیاورند و آن را برای مقاصد مختلف، از سرورهای اینترنتی گرفته تا دستگاه های شخصی، به کار بگیرند. در نتیجه، لینوکس نه تنها به عنوان یک سیستم عامل، بلکه به عنوان یک جنبش فرهنگی در دنیای نرم افزار شناخته می شود.

فلسفه زندگی و فلسفه فناوری

همان طور که بسیاری از ما در زندگی شخصی خود دارای یک فلسفه هستیم، شرکت ها و سیستم عامل ها نیز فلسفه های خاص خود را دارند. این فلسفه ها ممکن است ساده یا پیچیده باشند، اما در هر صورت، آن ها راهنمای رفتار و تصمیم گیری ها هستند. برای مثال، زمانی که در شرکت IBM کار می کردم، فلسفه این شرکت به طور کامل مستند شده بود و در فرهنگ سازمانی آن ریشه دوانده بود. فلسفه IBM بر پایه رفتار منصفانه، محترمانه و با کرامت با کارکنان، مشتریان و تأمین کنندگان استوار بود. این فلسفه نه تنها نحوه تعاملات تجاری را شکل می داد، بلکه بر محصولات و خدمات شرکت نیز تأثیر می گذاشت.

این فلسفه سازمانی به طور مستقیم بر روابط داخلی و خارجی شرکت تأثیرگذار بود. به عنوان مثال، توجه به منافع مشترک و اهمیت همکاری های طولانی مدت با مشتریان باعث شد که IBM به طور مداوم خدماتی متناسب با نیازهای خاص مشتریان خود ارائه دهد. این رویکرد به ویژه در تولید و توسعه نرم افزارها و سیستم های تکنولوژیک

اصول، هسته‌ی اصلی طرز فکر در دنیای نرم‌افزارهای متن‌باز را تشکیل می‌دهند و راهنمایی برای طراحی و توسعه‌ی سیستم‌های پایدار، کارآمد و انعطاف‌پذیر هستند.

از سوی دیگر، **اریک ریموند** در کتاب معروف خود (*The Art of Unix Programming*) هنر برنامه‌نویسی یونیکس (۱۷ قانون مهم درباره‌ی طراحی و توسعه‌ی نرم‌افزارهای یونیکس را ارائه داده است. او این قوانین را از تجربه‌ی چندین دهه کار با یونیکس و لینوکس استخراج کرده و نشان داده که چگونه این سیستم‌عامل‌ها به مرور زمان تکامل یافته‌اند.

همچنین، **دانشگاه ایالتی اورگان** فلسفه‌ی خاص خود را در مورد لینوکس ارائه داده است. این دیدگاه، لینوکس را از منظر یک **مهندس نرم‌افزار** بررسی می‌کند و نشان می‌دهد که چگونه معماری باز و انعطاف‌پذیر لینوکس باعث شده است که امروزه به یکی از پرکاربردترین سیستم‌عامل‌های جهان تبدیل شود. بسیاری از شرکت‌ها فلسفه‌های خاص خود را دارند که معمولاً تأثیر زیادی بر فرهنگ سازمانی و نحوه تعاملات تجاری آن‌ها می‌گذارد. برای مثال، در IBM، فلسفه‌ای که طی سال‌ها رشد کرده و به خوبی مستند شده بود، مبنای تمامی رفتارها و تصمیمات این شرکت بود. این فلسفه بر مبنای منصفانه بودن، احترام به کرامت انسانی و ایجاد تعاملات محترمانه با تمامی ذینفعان قرار داشت.

در دنیای سیستم‌عامل‌ها نیز هر کدام فلسفه خاص خود را دارند. فلسفه ویندوز مشابه سیستم‌عامل VMS است که کاربران را از پیچیدگی‌های فنی دور نگه می‌دارد تا تجربه‌ای ساده و بدون دردسر داشته باشند. این تمایل به محافظت از کاربران از پیچیدگی‌ها در طراحی رابط کاربری ویندوز و دیگر سیستم‌عامل‌ها مشهود است. برخلاف ویندوز و macOS که به کاربر تجربه‌ای کاربرپسندتر و ساده‌تر می‌دهند، لینوکس بیشتر به آزادی و قابلیت‌های پیشرفته برای توسعه‌دهندگان و کاربران حرفه‌ای تأکید دارد. فلسفه لینوکس بر این استوار است که کاربر باید کنترل کامل سیستم را در دست داشته باشد و قادر به تغییر و سفارشی‌سازی آن باشد. از این رو، سیستم‌عامل‌هایی مانند ویندوز و macOS نسبت به لینوکس محدودیت‌های بیشتری دارند و اغلب بسیاری از تنظیمات پیچیده و دستورات پیشرفته را پنهان می‌کنند.

اصول کلیدی فلسفه یونیکس و لینوکس

بیشتر اصول فلسفه یونیکس و لینوکس حول سادگی، کارایی و انعطاف‌پذیری می‌چرخد. این اصول، موجب می‌شود که کاربران تجربه‌ای قدرتمند و شخصی‌سازی شده از سیستم‌عامل خود داشته باشند. در ادامه، برخی از این اصول را با جزئیات بیشتری توضیح می‌دهیم:

لینوکس در برابر مدل‌های سنتی توسعه نرم‌افزار: نگاهی به تفاوت‌های اساسی

امروزه، نرم‌افزار متن‌باز، به‌ویژه لینوکس، در مرکز توجه رسانه‌ها و دنیای فناوری قرار دارد. بسیاری از تحلیل‌گران و متخصصان معتقدند که مدل توسعه‌ی لینوکس و نرم‌افزارهای متن‌باز، رویکرد سنتی ساخت نرم‌افزار را دگرگون کرده است. اما برای درک بهتر این تفاوت، باید نگاهی به مدل سنتی توسعه نرم‌افزار تجاری بیندازیم.

مدل سنتی توسعه‌ی نرم‌افزار: کنترل، سلسله‌مراتب و سیاست‌های سخت‌گیرانه

شرکت‌های نرم‌افزاری بزرگ معمولاً توسعه‌ی نرم‌افزار را بر اساس مجموعه‌ای از سیاست‌های دقیق و فرایندهای کنترلی سخت‌گیرانه انجام می‌دهند. این مدل دارای ویژگی‌های زیر است:

- **تضمین کیفیت (Quality Assurance - QA):** هر تغییری که در کد اعمال می‌شود، باید از فیلترهای مختلف عبور کند. تیم‌های آزمایش و تضمین کیفیت (QA) مسئول بررسی این تغییرات و شناسایی اشکالات هستند.
- **سیستم‌های کنترل نسخه:** تغییرات نرم‌افزاری باید در سیستم‌های کنترل منبع (مانند Git) در مدل‌های مدرن (ثبت شوند). هیچ توسعه‌دهنده‌ای نمی‌تواند آزادانه بخش‌های مهم کد را تغییر دهد، مگر اینکه دلایل قانع‌کننده‌ای ارائه کند و فرآیند تأیید را طی کند.
- **محدودیت‌های توسعه‌دهندگان:** در شرکت‌های بزرگ، هر توسعه‌دهنده فقط به بخش مشخصی از کد دسترسی دارد و اجازه‌ی تغییر بخش‌های دیگر را ندارد. این کار برای مدیریت بهتر تغییرات و کاهش ریسک‌های احتمالی انجام می‌شود.
- **مستندسازی و تحلیل آماری:** تیم‌های نرم‌افزاری مستندات دقیقی از تمامی تغییرات، مشکلات و راه‌حل‌ها تهیه می‌کنند تا در آینده امکان رفع اشکال یا بهبود عملکرد نرم‌افزار فراهم باشد. همچنین، معیارهای آماری برای تعیین زمان انتشار نسخه‌ی بعدی نرم‌افزار استفاده می‌شود.

این فرآیند در پروژه‌های بزرگ تجاری، مانند سیستم‌عامل‌های مایکروسافت و ویندوز یا macOS، بسیار پیچیده و زمان‌بر است. هر نسخه‌ی جدید از این نرم‌افزارها معمولاً پس از ماه‌ها (و گاهی سال‌ها) تست و بررسی منتشر می‌شود، زیرا شرکت‌ها باید از کیفیت و پایداری محصول خود اطمینان حاصل کنند.

“لینوکس منسوخ شده است”

در سال ۱۹۹۲، اندرو اس. تانباوم، که برای نوشتن سیستم میکرو کرنل Minix شناخته می‌شد، مقاله‌ای در گروه خبری یوزنت تحت عنوان “لینوکس منسوخ است” منتشر کرد. این مقاله به سرعت تبدیل به یکی از موضوعات بحث‌برانگیز در دنیای سیستم‌های عامل شد. تانباوم در این مقاله، انتقادهای جدی از ساختار هسته لینوکس وارد کرد و آن را قدیمی و غیرمنطقی دانست. یکی از مهم‌ترین نقدهای او این بود که هسته لینوکس یکپارچه است و به همین دلیل نمی‌تواند به اندازه کافی انعطاف‌پذیر باشد. به گفته او، این نوع هسته‌ها که در آن همه چیز در داخل یک بخش بزرگ قرار دارد، منجر به مشکلاتی در توسعه و نگهداری سیستم خواهد شد.

نقد دیگری که تانباوم مطرح کرد، عدم قابلیت حمل لینوکس بود. او اشاره کرد که لینوکس به شدت به ویژگی‌های پردازنده اینتل ۳۸۶ وابسته است، که باعث محدود شدن آن به سیستم‌هایی می‌شود که از این پردازنده استفاده می‌کنند. از دیدگاه او، نوشتن سیستم عاملی که به سخت‌افزار خاصی وابسته باشد، اشتباه است و این می‌تواند مانع از گسترش و پیشرفت سیستم عامل در آینده شود. در حالی که او به این انتقاد توجه داشت، تاریخ نشان داد که لینوکس به سرعت به سایر پلتفرم‌ها منتقل شد و در سیستم‌های مختلف کارکرد خوبی داشت.

یکی دیگر از انتقادات تانباوم مربوط به مدیریت کد منبع بود. او معتقد بود که هیچ نظارت و کنترلی بر کد منبع لینوکس وجود ندارد، که این می‌تواند منجر به مشکلاتی در هماهنگی و استانداردسازی سیستم عامل شود. به گفته او، این مشکل در نهایت می‌تواند رشد و پیشرفت لینوکس را کند کرده و آن را در برابر رقبا ضعیف‌تر کند. این نگرانی نیز نادرست از آب درآمد، چرا که مدل توسعه باز لینوکس به شکلی ساختاریافته و با همکاری هزاران توسعه‌دهنده در سراسر دنیا پیشرفت کرد. انتقاد دیگری که توسط تانباوم مطرح شد، مربوط به ویژگی‌هایی مانند سیستم‌های فایل چندرشته‌ای بود که او آن‌ها را “هک عملکرد” می‌دانست و معتقد بود که این ویژگی‌ها هیچ فایده واقعی ندارند. این نظر نیز بعداً رد شد، زیرا لینوکس توانست این ویژگی‌ها را به شکلی بهینه‌سازی کند و آن‌ها را در محیط‌های مختلف استفاده کند.

پیش‌بینی تانباوم مبنی بر اینکه لینوکس در چند سال منسوخ خواهد شد و گنو هرد جایگزین آن خواهد شد، به وضوح نادرست از آب درآمد. لینوکس نه تنها به طور موفقیت‌آمیز به پلتفرم‌های مختلف منتقل شد، بلکه مدل توسعه باز آن به یک نمونه برجسته از توسعه نرم‌افزار تبدیل شد. در مقابل، گنو هرد، که تانباوم آن را مدرن‌تر و مناسب‌تر می‌دانست، هنوز نتوانسته است به سطح ثبات و قابلیت استفاده در سرورهای تولیدی برسد. این موضوع نشان داد که پیش‌بینی‌های او درباره آینده لینوکس اشتباه بوده‌اند. در نهایت، این که تانباوم پردازنده‌های سری X۸۶ اینتل را “عجیب” دانست، خود یک خطای پیش‌بینی دیگر بود. پردازنده‌های این خانواده به یکی از پرفروش‌ترین و پراستفاده‌ترین پردازنده‌ها در جهان تبدیل شدند و شرکت اینتل به یکی از غول‌های فناوری جهان تبدیل شد. این اشتباهات در ارزیابی تانباوم به وضوح نشان می‌دهد که پیش‌بینی‌های تکنولوژیکی می‌توانند تا چه اندازه اشتباه از آب درآیند و در مقابل، لینوکس توانست به پیشرفت‌های شگرفی دست یابد که تانباوم هیچگاه تصورش را نمی‌کرد.

ویدئوهای K۴ و K۸، نیاز به بهینه‌سازی API های مدیریت ویدئو در لینوکس بیش از پیش احساس می‌شود. بهینه‌سازی عملکرد، کاهش تأخیر در پردازش ویدئو و پشتیبانی از استانداردهای جدید، از جمله مواردی هستند که در توسعه نسخه‌های آینده این API ها مورد توجه قرار گرفته‌اند. در نهایت، آینده مدیریت ویدئو در لینوکس به میزان همکاری بین توسعه‌دهندگان متن‌باز، شرکت‌های سخت‌افزاری و جامعه کاربران بستگی دارد. استانداردسازی بیشتر، توسعه درایورهای باکیفیت و افزایش پشتیبانی از قابلیت‌های جدید، می‌تواند تجربه کاربری بهتری را در زمینه پردازش ویدئوی دیجیتال ارائه دهد.

تاریخچه توسعه

تاریخچه ساده شده سیستم عامل های یونیکس مانند لینوکس دارای معماری و مفاهیم مشابهی است) به عنوان بخشی از استاندارد (POSIX اما کد منبع غیر رایگان را با یونیکس یا MINIX اصلی به اشتراک نمی‌گذارد.

توزیع لینوکس و نرم افزار

تفاوت اصلی بین لینوکس و بسیاری دیگر از سیستم عامل های محبوب معاصر این است که هسته لینوکس و سایر اجزای آن نرم افزار رایگان و منبع باز هستند. لینوکس تنها چنین سیستم عاملی نیست، اگرچه بیشترین استفاده را دارد. برخی از مجوزهای نرم‌افزار آزاد و متن‌باز بر اساس اصل کپی‌لفت، نوعی دوسویه است: هر اثری که از یک نرم‌افزار کپی‌لفت مشتق شده باشد، باید خود کپی‌لفت نیز باشد. رایج‌ترین مجوز نرم‌افزار آزاد، مجوز عمومی عمومی گنو (GPL)، نوعی کپی‌لفت است و برای هسته لینوکس و بسیاری از اجزای پروژه گنو استفاده می‌شود. توزیع‌های مبتنی بر لینوکس توسط توسعه‌دهندگان برای قابلیت همکاری با سایر سیستم عامل ها و استانداردهای محاسباتی تعیین شده در نظر گرفته شده‌اند. سیستم‌های لینوکس در صورت امکان از استانداردهای SUS، POSIX، LSB، ISO و ANSI پیروی می‌کنند، اگرچه تا به امروز تنها یک توزیع لینوکس دارای گواهینامه ۱، POSIX، Linux-FT است. پروژه‌های نرم‌افزار رایگان، اگرچه از طریق همکاری توسعه یافته‌اند، اغلب مستقل از یکدیگر تولید می‌شوند. با این حال، این واقعیت که مجوزهای نرم‌افزار به صراحت اجازه توزیع مجدد را می‌دهند، مبنایی را برای پروژه‌های مقیاس بزرگتر فراهم می‌کند که نرم‌افزار تولید شده توسط پروژه‌های مستقل را جمع‌آوری می‌کند و آن را به یکباره در قالب یک توزیع لینوکس در دسترس قرار می‌دهد.

نقش هسته لینوکس در توسعه سیستم عامل

هسته لینوکس که توسط لینوس توروالدز در سال ۱۹۹۱ ایجاد شد، پایه و اساس این سیستم عامل است. این هسته منابع سخت‌افزاری را مدیریت می‌کند، فرآیندها را زمان‌بندی می‌کند و امکانات شبکه را فراهم می‌سازد. هسته لینوکس به دلیل طراحی ماژولار خود، امکان گسترش و سفارشی‌سازی را برای توسعه‌دهندگان فراهم می‌کند.

فصل ۵

سهم بازار و جذب

بسیاری از مطالعات کمی در مورد نرم افزارهای متن باز/رایگان بر موضوعاتی از جمله سهم بازار و قابلیت اطمینان تمرکز دارند، با مطالعات متعددی که به طور خاص لینوکس را بررسی می کنند. بازار لینوکس در حال رشد است و انتظار می رود اندازه بازار سیستم عامل لینوکس تا سال ۲۰۲۷ شاهد رشد ۱۹,۲ درصدی باشد و به ۱۵,۶۴ میلیارد دلار برسد، در مقایسه با ۳,۸۹ میلیارد دلار در سال ۲۰۱۹. تحلیلگران و طرفداران، موفقیت نسبی لینوکس را به امنیت و قابلیت اطمینان آن نسبت می دهند، همچنین هزینه کم و آزادی از قفل شدن به فروشنده از دیگر مزایای آن محسوب می شود. **دسکتاپ و لپ تاپ** بر اساس آمار وب سرور (یعنی بر اساس اعداد ثبت شده از بازدید از وبسایت ها توسط دستگاه های مشتری)، تا ماه مه ۲۰۲۲، تخمینی سهم بازار لینوکس در رایانه های رومیزی حدود ۲,۵٪ است. در مقایسه، مایکروسافت ویندوز سهم بازار حدود ۷۵,۵ درصد را دارد در حالی که macOS حدود ۱۴,۹ درصد را پوشش می دهد. با وجود این سهم بازار نسبتاً کوچک، لینوکس در میان کاربران حرفه ای و توسعه دهندگان محبوبیت خاصی دارد. برخی از محبوب ترین توزیع های لینوکس برای کاربران دسکتاپ شامل اوبونتو، فدورا و لینوکس مینت هستند که تجربه ای پایدار و کاربرپسند ارائه می دهند.

وب سرورها W³Cook

آماري را منتشر می کند که از ۱,۰۰۰,۰۰۰ دامنه برتر الکسا استفاده می کنند، که تا می ۲۰۱۵ تخمین می زند که ۹۶,۵۵٪ از وب سرورها لینوکس، ۱,۷۳٪ ویندوز و ۱,۷۲٪ FreeBSD را اجرا می کنند W³Techs. نیز آماری را منتشر می کند که از ۱۰,۰۰۰,۰۰۰ دامنه برتر الکسا و ۱,۰۰۰,۰۰۰ دامنه برتر Tranco استفاده می کند. این داده ها ماهانه به روز می شوند و تا نوامبر ۲۰۲۰ تخمین زده می شود که لینوکس توسط ۳۹٪ از سرورهای وب استفاده می شود، در مقابل ۲۱,۹٪ توسط مایکروسافت ویندوز استفاده می شود. ۴۰,۱ درصد از انواع دیگر یونیکس استفاده کردند. گزارش سه ماهه اول سال ۲۰۰۷ IDC نشان داد که لینوکس ۱۲,۷ درصد از کل بازار سرور را در آن زمان در اختیار داشت. این تخمین بر اساس تعداد سرورهای لینوکس فروخته شده توسط شرکت های مختلف بود و شامل سخت افزار سرور خریداری شده جداگانه که لینوکس بعداً روی آن نصب شده بود، نمی شد.

مزایای لینوکس

یکی از دلایل محبوبیت لینوکس در سرورها و محیط های ابری، پایداری، امنیت بالا و عدم وابستگی به یک شرکت خاص است. برخلاف ویندوز که نیاز به مجوزهای پولی دارد، لینوکس معمولاً رایگان بوده و تحت مجوزهای متن باز مانند GPL منتشر می شود. همچنین، به دلیل دسترسی کامل به کد منبع، کاربران و شرکت ها می توانند لینوکس را به نیازهای خود سفارشی سازی کنند. علاوه بر این، اکوسیستم نرم افزاری قوی، ابزارهای توسعه فراوان و انعطاف پذیری بالا باعث شده است که لینوکس در بین توسعه دهندگان نرم افزار و متخصصان شبکه جایگاه ویژه ای داشته باشد. **چالش ها و محدودیت ها** با وجود تمام مزایای خود، لینوکس در برخی حوزه ها با چالش هایی مواجه است. یکی از این چالش ها، سهم نسبتاً پایین در بازار دسکتاپ است. بسیاری از کاربران عادی ترجیح می دهند از

حق چاپ، علامت تجاری، و نامگذاری

هسته لینوکس تحت مجوز عمومی عمومی گنو (GPL)، نسخه ۲ مجوز دارد. GPL، ایجاب می‌کند که هر کسی که نرم‌افزار را بر اساس کد منبع تحت این مجوز توزیع می‌کند، باید کد منبع اصلی (و هرگونه تغییر) را تحت همین مجوز در اختیار گیرنده قرار دهد. سایر اجزای کلیدی یک توزیع معمولی لینوکس نیز عمدتاً تحت مجوز GPL هستند، اما ممکن است از مجوزهای دیگری استفاده کنند. بسیاری از کتابخانه‌ها از مجوز عمومی عمومی کمتر گنو (LGPL)، یک نوع مجازتر از GPL، و پیاده‌سازی X.Org سیستم پنجره X از مجوز MIT استفاده می‌کنند.

توروالدز بیان می‌کند که هسته لینوکس از نسخه ۲ GPL به نسخه ۳ منتقل نمی‌شود. او به طور خاص از برخی مقررات در مجوز جدید که استفاده از نرم‌افزار را در مدیریت حقوق دیجیتال منع می‌کند، بیزار است. همچنین گرفتن مجوز از همه دارندگان حق چاپ، که تعداد آنها به هزاران می‌رسد، غیرعملی خواهد بود.

مطالعه سال ۲۰۰۱ روی ۷،۱ Red Hat Linux نشان داد که این توزیع حاوی ۳۰ میلیون خط منبع کد است. با استفاده از مدل هزینه سازنده، این مطالعه تخمین زد که این توزیع به حدود هشت هزار سال زمان توسعه نیاز دارد. بر اساس این مطالعه، اگر همه این نرم‌افزار با ابزارهای اختصاصی متعارف توسعه داده می‌شد، حدود ۱،۵۷ میلیارد دلار برای توسعه آن در سال ۲۰۲۰ در ایالات متحده هزینه می‌شد. بیشتر کد منبع (۷۱٪) به زبان برنامه‌نویسی C نوشته شده بود، اما بسیاری از زبان‌های دیگر از جمله ++C، Lisp، زبان اسمبلی، Perl، Python، و Fortran زبان‌های مختلف پورته اسکریپت استفاده شدند. کمی بیش از نیمی از تمام خطوط کد تحت مجوز GPL بودند. خود هسته لینوکس ۲،۴ میلیون خط کد یا ۸ درصد از کل کد بود.

یکی از ویژگی‌های کلیدی لینوکس انعطاف‌پذیری آن در استفاده در انواع مختلف دستگاه‌ها از رایانه‌های رومیزی تا سرورها، ابررایانه‌ها، و حتی دستگاه‌های کوچک تعبیه‌شده است. توسعه‌دهندگان از سراسر جهان در بهبود و بهینه‌سازی کد هسته لینوکس مشارکت دارند. جامعه توسعه‌دهندگان لینوکس بسیار گسترده و پویاست و هر ساله هزاران وصله (patch) و تغییر به هسته لینوکس اضافه می‌شود. لینوکس نه تنها به عنوان یک سیستم‌عامل برای کاربران نهایی محبوب است، بلکه به عنوان پایه‌ای برای بسیاری از زیرساخت‌های حیاتی نیز مورد استفاده قرار می‌گیرد. بسیاری از سرورهای اینترنتی، دستگاه‌های شبکه، پایگاه‌های داده و خدمات ابری از لینوکس به دلیل امنیت بالا، پایداری و انعطاف‌پذیری آن استفاده می‌کنند. توزیع‌های معروفی مانند Ubuntu، Debian، Fedora، CentOS و Arch Linux هر کدام ویژگی‌های خاصی دارند و نیازهای مختلف کاربران را برآورده می‌کنند. یکی از بزرگ‌ترین مزایای لینوکس در مقایسه با سیستم‌عامل‌های اختصاصی، مدل توسعه متن‌باز آن است. این مدل نه تنها هزینه‌های توسعه را کاهش می‌دهد، بلکه باعث افزایش نوآوری و ایجاد جامعه‌ای فعال از توسعه‌دهندگان و کاربران می‌شود که به رفع اشکالات و بهبود عملکرد کمک می‌کنند. همچنین، امنیت لینوکس نسبت به بسیاری از سیستم‌عامل‌های اختصاصی بالاتر است، زیرا کاربران و توسعه‌دهندگان می‌توانند کد منبع را بررسی کرده و آسیب‌پذیری‌های احتمالی را اصلاح کنند.

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.



Emperor-OS



Predator-OS