

عنوان کتاب: مدیریت و برنامه‌نویسی گرداننده‌ها در لینوکس Linux Driver Programming Management

مؤلف و مترجم: امیر حسین پی‌براه، وحید کاظم پور، حامد جانزاده

ناشر: انتشارات اردلان - شرکت داده پردازی ایران

ویراستار علمی: محمد تشکری

ویراستار متن: سید حسین بنی احمد - مهرداد تاج بخش

طراحی و صفحه‌آرایی: مهران تاج بخش

لیتوگرافی: نصر

چاپ: هاشمی

صحافی: پیام

تیراژ: ۳۵۰۰ نسخه

نویت چاپ: اول - بهار ۱۳۸۵



مرکز انحصاری پخش: انتشارات اردلان

نشانی: تهران، خیابان استاد نجات الهی، خیابان مراغه، شماره ۶، واحد ۱۰۸

تلفن: ۸۸۸۰۶۴۸۰ - ۸۸۹۲۶۲۸۴ - ۸۸۹۲۶۲۸۵

فاکس: ۸۸۸۰۶۴۷۹

پست الکترونیکی: ardalanpublishing@easta.net

وب سایت: www.ardalanpublishing.com

پی‌براه، امیر حسین

مدیریت و برنامه‌نویسی گرداننده‌ها در لینوکس / امیر حسین پی‌براه، وحید کاظم پور، حامد جانزاده — تهران: اردلان، ۱۳۸۴.

۲۶۰ ص.

ISBN 964-92399-8-7

فهرست‌نویسی بر اساس اطلاعات فیپا.

۱. سیستم عامل لینوکس. ۲. برنامه‌نویسی سیستمها. الف. کاظم پور، وحید،

ب. جانزاده، حامد. ج. عنوان.

۰۰۵/۴۴۶

۸۷پ۹۴/س/۷۶/۷۶ QA

۸۴-۳۹۲۹۶م

کتابخانه ملی ایران

کلیه حقوق مادی و معنوی این کتاب متعلق به شرکت داده پردازی و انتشارات اردلان می‌باشد.

هرگونه تکثیر و کپی برداری منوط به اخذ مجوز رسمی از شرکت داده پردازی ایران می‌باشد.

پیش گفتار

معرفی

۱- نقش گرداننده دستگاه

۱-۲-۱ ساختار هسته لینوکس

۱-۲-۱-۱ وظایف اصلی سیستم عامل

۱-۲-۱-۲ ساختار فایل های دستگاه (Device File)

۱-۲-۱-۳ پیمانه های قابل بار شدن (Loadable Modules)

۱-۳-۱ انواع دستگاه ها در لینوکس

۱-۴-۱ نحوه نصب گرداننده دستگاه به صورت پیمانه

۲- برنامه نویسی پیمانه

۲-۱-۲ ورود به دنیای پیمانه ها

۲-۲-۲ کامپایل و بار کردن پیمانه ها

۲-۲-۲-۱ کامپایل کردن پیمانه ها در هسته ۲.۴

۲-۲-۲-۲ کامپایل کردن پیمانه ها در هسته ۲.۶ سیستم عامل

۲-۳ جدول سمبل ها (Symbol Table)

۲-۴ شروع و خاتمه عملیات پیمانه ها

۲-۴-۱ ماکروه های `__init` و `__exit`

۲-۴-۲ کنترل خطا در حین عملیات شروع

۲-۵ ایجاد مجوز

۲-۶ دادن آرگومان به پیمانه ها

۳

۵

۵

۸

۱۲

۱۳

۱۶

۲۴

۲۵

۲۸

۲۸

۳۱

۳۴

۳۷

۳۹

۴۱

۴۳

۴۷

۵۲	۷-۲ پیمانه هایی که در چند فایل نوشته شده اند
۵۴	۸-۲ پیمانه برای هسته های از قبل کامپایل شده
۵۹	۳- گرداننده دستگاه رشته ای
۶۰	۱-۳ طراحی dpichr
۶۱	۲-۳ اعداد minor و major
۶۳	۳-۲ نحوه نمایش شماره دستگاه ها در هسته
۶۴	۳-۲-۲ تخصیص دادن و آزاد کردن شماره دستگاه ها
۶۶	۳-۲-۳ تخصیص دادن اعداد major به صورت پویا
۷۲	۳-۳ برخی از ساختمان داده های مهم
۷۳	۱-۳-۳ ساختار file_operation
۸۲	۲-۳-۳ ساختار inode
۸۵	۳-۳-۳ ساختار inode
۸۶	۴-۳ ثبت کردن دستگاه رشته ای
۸۸	۳-۴-۱ ثبت کردن دستگاه در dpichr
۹۰	۳-۴-۲ روش قدیمی
۹۱	۵-۳ عملکرد های open و release
۹۲	۱-۵-۳ عملکرد open
۹۴	۲-۵-۳ عملکرد release
۹۶	۶-۳ استفاده از حافظه در dpichr
۹۸	۷-۳ عملکرد های read و write

۱۰۴

read عملکرد ۱-۷-۳

۱۰۵

write عملکرد ۲-۷-۳

۱۰۶

writev و readv عملکرد های ۳-۷-۳

۱۰۸

۸- استفاده از گرداننده جدید

۱۱۰

۴- مباحث پیشرفته در برنامه نویسی پیمانه

۱۱۱

۱-۴ روش های رفع خطا

۱۱۱

۱-۴-۱ رفع خطا با استفاده از چاپ کردن

۱۱۶

۱-۴-۲ رفع خطا با استفاده از گزارش گرفتن

۱۲۱

۲-۴ همزمانی

۱۲۲

۱-۲-۴ سمافور و mutex

۱۲۷

۲-۲-۴ مفهوم spinlock

۱۳۳

۳-۴ زمان بندی

۱۳۳

۱-۳-۴ محاسبه زمان

۱۳۶

۲-۳-۴ زمان جاری سیستم

۱۳۸

۳-۳-۴ ایجاد تاخیر

۱۳۸

۱-۳-۳-۴ تاخیر های بزرگ

۱۴۱

۲-۳-۳-۴ تاخیر های کوچک

۱۴۲

۴-۳-۴ مفهوم timer در هسته لینوکس

۱۴۶

۵-۳-۴ مفهوم tasklet

۱۴۸

۶-۳-۴ صف کار (workqueue)

۱۵۱

۴-۴ تخصیص حافظه

۱۵۱	۱-۴-۴ تابع kmalloc
۱۵۲	۲-۴-۴ تابع get_free_page
۱۵۴	۳-۴-۴ تابع vmalloc
۱۵۵	۵- کارکردهای پیشرفته در گرداننده دستگاه رشته ای
۱۵۶	۱-۵ ioctl
۱۵۸	۱-۱-۵ انتخاب دستور ioctl
۱۶۲	۲-۱-۵ مقادیر بازگشتی
۱۶۳	۳-۱-۵ دستورات از پیش تعریف شده
۱۶۵	۴-۱-۵ استفاده از آرگومان های ioctl
۱۶۹	۵-۱-۵ سطوح دسترسی و فعالیت های حفاظت شده
۱۷۱	۶-۱-۵ پیاده سازی فرامین ioctl
۱۷۲	۲-۵ ورودی / خروجی مسدود شدنی
۱۷۳	۱-۲-۵ آشنایی با حالت انتظار (sleeping)
۱۷۵	۲-۲-۵ حالت انتظار به روش ساده
۱۷۹	۳-۲-۵ عملیات مسدود شدنی و مسدود نشدنی
۱۸۲	۴-۲-۵ یک نمونه ورودی / خروجی مسدود شدنی
۱۸۶	۵-۲-۵ حالت انتظار پیشرفته
۱۸۶	۱-۵-۲ چگونه یک فرایند به حالت انتظار می رود
۱۸۹	۲-۵-۲ مسدود کردن دستی
۱۹۴	۳-۵-۲ انتظار های انحصاری
۱۹۶	۴-۵-۲ جزئیات بیدار کردن

۱۹۸

۱۹۹

۲۰۰

۲۰۵

۲۰۶

۲۰۷

۲۰۸

۲۰۹

۲۱۱

۲۱۲

۲۱۶

۲۱۶

۲۱۷

۲۱۹

۲۲۰

۲۲۱

۲۲۴

۲۲۵

۲۲۷

۲۳۰

۲۳۱

۵-۵-۲-۵ داستان قدیمی sleep_on

۶-۲-۵ آزمایش کردن گرداننده dpichrpipe

۳-۵ سیستم فراخوان های pool و select

۱-۳-۵ تعامل با read و write

۱-۳-۵ خواندن داده از دستگاه

۲-۳-۵ نوشتن در دستگاه

۳-۱-۳-۵ تخلیه داده در زمان خروجی

۶- ارتباط با سخت افزار

۱-۶ پورت های ورودی / خروجی و حافظه ورودی / خروجی

۱-۱-۶ ثبات های ورودی / خروجی و حافظه قراردادی

۲-۶ استفاده از پورت های ورودی / خروجی

۱-۲-۶ تخصیص دهی پورت ورودی / خروجی

۲-۲-۶ استفاده از پورت های ورودی / خروجی

۳-۲-۶ دسترسی به پورت ورودی / خروجی از طریق فضای کاربر

۴-۲-۶ اعمال رشته ای

۵-۲-۶ وابستگی به سخت افزار

۳-۶ استفاده از حافظه ورودی / خروجی

۱-۳-۶ تخصیص و نگاشت حافظه ورودی / خروجی

۲-۳-۶ دسترسی به حافظه ورودی / خروجی

۳-۳-۶ پورت ها در حافظه ورودی / خروجی

۴-۳-۶ پیمانه نمونه برای حافظه ورودی و خروجی

۲۳۵	۷- وقفه ها
۲۳۷	۷-۱ آماده سازی پورت موازی
۲۳۷	۷-۲ راه اندازی یک رسیدگی کننده وقفه
۲۴۲	۷-۲-۱ واسطه /proc
۲۴۵	۷-۲-۲ تشخیص شماره IRQ به صورت خودکار
۲۴۷	۷-۲-۲-۱ جستجوی وابسته به هسته
۲۵۰	۷-۲-۲-۲ جستجوی دستی
۲۵۳	۷-۲-۳ رسیدگی کننده های سریع و کند
۲۵۴	۷-۳ پیاده سازی یک رسیدگی کننده وقفه
۲۵۹	۷-۳-۱ آرگومان ها و مقادیر بازگشتی مربوط به رسیدگی کننده
۲۶۱	۷-۳-۲ فعال و غیر فعال کردن وقفه
۲۶۱	۷-۳-۲-۱ غیر فعال نمودن یک وقفه واحد
۲۶۲	۷-۳-۲-۲ غیر فعال نمودن تمامی وقفه ها

پیشگفتار

بی شک اولین نسل کاربران رایانه های رومیزی (PC) در پذیرش برخی تحولات و محصولات اختیاری از خود نداشتند. چرا که تنها سیستم عامل مطرح در جهان -DOS- محصول شرکت مایکرو سافت بود. پس از DOS سیستم عامل Windows هم به همین صورت - یعنی انحصاری - توسعه پیدا کرد و در واقع انحصار بازار سیستم عامل کامپیوترهای شخصی در دست شرکت مایکرو سافت قرار گرفت. با افزایش تعداد کاربران کامپیوترهای شخصی و کاربردهای آن و نیز پیشرفت فن آوری ساخت CPU های قدرتمند، توسعه سیستم عامل Windows به سرعت افزایش یافت. جالب اینکه همه این سال ها جای خالی یک رقیب جدی چندان محسوس نبود. اما با تصمیمات دولت ها مبنی بر استفاده از IT (فن آوری اطلاعات) جهت سهولت بخشیدن به فرآیندهای اداری و مالی و تشدید این تصمیمات توسط شرکت های خصوصی عملکرد انحصاری مایکرو سافت نمایان گردید. شرکت های کامپیوتری مختلفی قصد رقابت با مایکرو سافت را داشتند اما پس از سعی فراوان یا شکست می خوردند یا توسط شرکت مایکرو سافت خریداری می شدند. حتی شکستن انحصار مایکرو سافت با تصویب قوانین در اروپا نیز امکان پذیر نشد. تنها شرکتی که توانست مشتریان خاص خود را بیابد و به عنوان رقیب (اما نه جدی) مطرح بماند شرکت Apple بود که البته سهم این شرکت از بازار به نسبت مایکرو سافت بسیار اندک است. تا اینکه نهضت Open Source دارای یک سیستم عامل شد. هسته این سیستم عامل را یک دانشجوی فنلاندی بنام Linus Torvalds (لینوس توروالدز) در سال ۱۹۹۰ ایجاد کرد و طبق مجوز GPL -از مجوز های Open Source- بر روی اینترنت قرار داد. Linus نام این هسته را Linux گذاشت. شرکت ها و افراد زیادی به سرعت ابزارهای گنو و هسته لینوکس را کنار هم گذاشتند و توزیع های GNU / Linux را ایجاد کردند. با توجه به آزادی هایی که مجوز GPL در اختیار کاربران قرار می دهد، کم کم توجه برنامه نویسان، شرکتها و دولتها به گنو / لینوکس و نرم افزارهای آزاد/متن باز بیشتر و بیشتر شد. تا جایی که شرکت معظم

Novell که از رقبای سرسخت Microsoft در سیستم عامل های سرویس دهنده کوچک و متوسط می باشد، توسعه سیستم عامل اصلی خود یعنی Netware را در نسخه 6.5 متوقف کرده و با خرید یک شرکت آلمانی توزیع کننده لینوکس بنام SuSE به قیمت ۲۵۰ میلیون دلار، برنامه های بازار خود را بر اصول و اساس GNU / Linux تنظیم کرد. در همین راستا شرکت های بزرگ نظیر HP نیز گامهای قابل توجهی برداشتند. شرکت HP نخستین Laptop های لینوکسی را به بازار عرضه کرد و راه انداز اکثر محصولات چاپگر و اسکنر خود را تحت لینوکس بصورت متن باز در اختیار توسعه دهندگان قرار داد. شرکت IBM نیز سرمایه گذاری هنگفتی بر روی لینوکس جهت پشتیبانی از Mainframe ها و سرورهای خود به مبلغ ۴۰۰ میلیون دلار انجام داده است. شرکت های جدیدی نیز با فعالیت خدمات و توسعه گنو / لینوکس ایجاد گردیده و کسب درآمد نموده اند. به عنوان مثال شرکت Redhat از معروف ترین توزیع کننده های گنو / لینوکس در دنیا در سال ۲۰۰۴ از پشتیبانی لینوکس مبلغ ۱۰ میلیون دلار درآمد داشته است. در کشور ما ایران نیز دولت تصمیم به استفاده از این سیستم عامل در صنعت IT گرفته و این سیستم عامل را به عنوان گزینه ای دیگر در کنار بقیه سیستم عامل ها به عموم کاربران پیشنهاد می نماید. استفاده از فناوری اطلاعات به زودی بصورت یک نیاز روزمره مطرح خواهد بود و آن وقت وابستگی در این صنعت به یک شرکت خارجی مثل این خواهد بود که ما در نیروگاه های برق به یک کشور خارجی وابسته باشیم. از قابلیت های گنو / لینوکس می توان به موارد زیر اشاره نمود:

پشتیبانی کامل از زبان فارسی، عدم وابستگی به بستر سخت افزاری، عدم تاثیر نصب و حذف برنامه های جانبی در سرعت سیستم، حداکثر قابلیت سفارشی شدن توسط کاربران و برنامه نویسان، عدم آسیب پذیری در مقابل ویروس، کرمها و برنامه های جاسوسی، سازگاری با اسناد تولید شده در سیستم عامل و برنامه های شرکت مایکروسافت، پشتیبانی فنی و آموزشی در داخل کشور، امکان نصب در کنار سیستم عامل قبلی (مثلا ویندوز)، پشتیبانی از انواع سخت افزارها و هزاران قابلیت منحصر به فرد دیگر.