Разлики при управлението на процеси и настройките на средата при Android и Linux

1. **Какво е Линукс?**

Linux е операционна система с отворен код (ОС). Операционната система е софтуерът, който директно управлява хардуера и ресурсите на системата, като процесор, памет и съхранение. ОС се намира между приложения и хардуер и прави връзките между целия софтуер и физическите ресурси, които вършат работата.

Linux е проектиран да бъде подобен на UNIX, но еволюира да работи на голямо разнообразие от хардуер от телефони до суперкомпютри. Всяка базирана на Linux операционна система включва ядрото на Linux – което управлява хардуерните ресурси – и набор от софтуерни пакети, които съставляват останалата част от операционната система.

Операционната система включва някои общи основни компоненти, като инструментите на GNU, наред с други. Тези инструменти дават на потребителя начин да управлява ресурсите, предоставени от ядрото, да инсталира допълнителен софтуер, да конфигурира настройките за производителност и сигурност и др. Всички тези инструменти, събрани заедно, съставляват функционалната операционна система. Тъй като Linux е операционна система с отворен код, комбинациите от софтуер могат да варират в различните дистрибуции на Linux.

Какво включва Linux ?

* Ядро

Основният компонент на ОС. Без него ОС не може да работи. Ядрото управлява ресурсите на системата и комуникира с хардуера. Той отговаря за управлението на паметта, процесите и файловете.

* Потребителско пространство на системата

Административният слой за задачи на системно ниво като конфигуриране и инсталиране на софтуер. Това включва обвивката или командния ред, даемони, процеси, които се изпълняват във фонов режим, и средата на работния плот.

* Приложения

Тип софтуер, който позволява да се изпълняват специфични задачи. Приложенията включват всичко - от различни видове инструменти и езици за програмиране до многопотребителски бизнес пакети. Повечето дистрибуции на Linux предлагат централна база данни за търсене и изтегляне на допълнителни приложения.

Linux е безплатна операционна система с отворен код, издадена под GNU General Public License (GPL). Всеки може да изпълнява, изучава, променя и разпространява изходния код или дори да продава копия на своя модифициран код, стига да го прави под същия лиценз.

Linux се е превърнал в най-големия софтуерен проект с отворен код в света. Професионални програмисти и любители от цял свят допринасят за развитието на ядрото на Linux, като добавят функции, намират и коригират грешки и пропуски в сигурността и предоставят нови идеи – като същевременно споделят приноса си обратно към общността.

**2. Какво е Андроид?**

Андроид е операционна система за мобилни устройства – смартфони и таблети. Той е безплатен софтуер с отворен код, наричан Android Open Source Project (AOSP). Лицензиран е под лиценза на Apache. Базиран е на модифицирана версия на ядрото на Linux и друг софтуер с отворен код.

* Архитектура на Андроид

Архитектурата на Андроид се състои от различни компоненти, които формират стек, състоящ се от приложения, които образуват горния слой на стека и ядрото на Линукс, което образува най-ниския слой.

* Приложения

В Андроид има вградени основни приложения - SMS програма, електронна поща, календар, карти, браузър, контакти и други функции. За всички приложения е използван езика за програмиране Java.

* Приложна рамка

Разработчиците могат да достъпят същите приложения на рамката, които използват основните приложения. Архитектурата на приложението цели опростяването на повторното използване на компоненти - т.е. възможностите на всяко приложение могат да бъдат публикувани и след това да бъдат използвани от всяко друго приложение, съобразявайки се с ограниченията за сигурност, наложени от рамката. Този механизъм позволява компонентите да бъдат заменени от потребителя.

* Библиотеки

Чрез Android application framework разработчиците могат да бъдат достъпят C/C++ библиотеките, които са включени в Андроид и се използват от различни компоненти на системата.

* Време за изпълнение

Приложенията за Андроид работят в свой собствен процес и имат собствен образец на виртуалната машина Dalvik. Dalvik е създаден така, че едно устройство да може ефективно да управлява множество виртуални машини. Виртуалната машина е базирана на регистър и изпълнява класове, компилирани на език за програмиране Java, който е трансформиран във формат .dex. За основна функционалност, като нишки и управление на паметта на ниско ниво, Dalvik VM разчита на ядрото на Linux.

* Linux ядрото

Линукс ядрото действа като слой между хардуера и останалата част от софтуерния стек.

Операционната система Андроид, разчита на Линукс за системни услуги като управление на процеси, управление на паметта, сигурност, мрежов стек и модел на драйвери. [1]

**3. Разликите между Линукс и Андроид**

Linux е предназначен за компютри на всички компании, докато Android е специално проектиран за мобилни устройства.

Ядрото, използвано в Linux, е Monolithi, докато типът ядро в Android е базиран на Linux.

Linux има предпочитания лиценз на GNU GPLv2 (ядро), докато Android използва предпочитания лиценз на Apache 2.0 и GNU GPLv2.

Linux се пише главно с помощта на C и асемблер, докато Android се пише с помощта на множество езици, включително C, C++, Java и други езици.

Linux се използва в персонални компютри със сложни задачи. Неговите целеви типове системи са вградени системи, мобилни устройства, персонални компютри, сървъри, мейнфрейм компютри и суперкомпютри.

Типовете целеви системи на Android са смартфони и таблетни компютри. Това е най-използваната операционна система като цяло.

Операционната система Android има своя собствена C библиотека, наречена Bionic, докато Linux системите използват библиотека GNU C.

Системите Android използват флаш памет вместо твърди дискове, докато стандартните Linux системи използват магнитни устройства.

Системите Android имат свой собствен мениджър на захранването, докато Linux системите използват APM и ACPI за управление на захранването.

**4. Управление на процесите**

Основните принципи за управление на процесите при операционната система Андроид се изразяват в това, че поддържа процеса на приложение възможно най-дълго време и унищожава процеса, само когато няма достатъчно памет. Това се прави с цел възстановяване на паметта за по-важни или нови процеси. Унищожаването на процеса се определя от нивото на приоритет.

Принципите на управление на процесите, контролира живота на процеса.

Жизненият цикъл улеснява потребителите при превключване между различни задачи и подпомага стартирането на множество приложения.

При Линукс възстановяването на паметта се осъществява, чрез премахване на процес, който има най-нисък приоритет. Нивата в йерархията са пет:

1. Активен процес - той е с критичен приоритет и е важен за дейността, която потребителят извършва в момента.

2.Видим процес - той е с висок приоритет и би могъл да повлияе на това, което потребителят вижда на екрана, въпреки, че не е с критична приоритетност.

3. Стартиран сервизен процес - той е с висок приоритет и изпълнява конкретна услуга.

4. Фонов процес - той е с нисък приоритет и има активност, която не е видима за потребителя.

5. Празен процес - той е с нисък приоритет, съществува за целите на кеширането и не съдържа активни компоненти на приложението. [2]

**5. Настройките на средата**

1. **Linux**

Различни конфигурационни файлове на Linux се използват за конфигуриране на параметрите и първоначалните настройки на различни части от процесите на операционната система. Някои от тези файлове дефинират мрежов достъп и протоколи, докато други инструктират операционната система как да достигне до сървърите за имена в интернет. Квалифициран администратор може да манипулира основната работа на Linux система, като редактира тези конфигурационни файлове.

Различните дистрибуции на Linux не само използват различни конфигурационни файлове на Linux, но и конвенциите за именуване между администраторите на Linux не са идентични. Понякога конфигурационните файлове се упоменават само с името на файла, като „dhclient“ или „nsswitch“. Но те също могат да бъдат посочени с разширението на файла след това, със или без точка, като „dhclient.conf“ или „nsswitch conf“.

Конфигурационният файл на мрежовите скриптове е мрежов конфигурационен файл в Linux, който контролира мрежов интерфейс. Конфигурационните параметри за дадено устройство или цифров интерфейс се съхраняват и редактират в конфигурационен файл, уникален за всеки интерфейс в директорията на мрежовите скриптове.

Има два начина да редактирате конфигурациите на мрежовия интерфейс в Linux:

GUI среда

Командна линия

И REHL (и CentOS), и Ubuntu съдържат графични помощни програми за редактиране на тези конфигурационни файлове. Тези графични помощни програми ще бъдат достъпни по различен начин в зависимост от това кой графичен интерфейс използвате.

Обикновено обаче тези настройки винаги могат да бъдат намерени в секцията за мрежи в системните предпочитания или секцията за помощни програми. Опциите за работа в мрежа ще изброят всеки интерфейс, наличен в системата. Редактирането на тези мрежови интерфейси ще има достъп до повечето от настройките, споменати по-горе.

1. Android

CFG файлът е общ файл с предпочитания, който съхранява настройки и информация за конфигурацията. Използва се от различни програми, което означава, че различните CFG файлове могат да съхраняват данни в различни формати. CFG файловете обикновено не трябва да се отварят ръчно, но могат да бъдат записани в текстов формат, който може да се види в текстов редактор.

Потребителският интерфейс на Android по подразбиране се основава главно на директна манипулация, като се използват сензорни входове, които до голяма степен съответстват на действия в реалния свят, като плъзгане, докосване, прищипване и обратно прищипване за манипулиране на обекти на екрана, заедно с виртуална клавиатура. Игровите контролери и физическите клавиатури в пълен размер се поддържат чрез Bluetooth или USB. Отговорът на въвеждането на потребителя е проектиран да бъде незабавен и осигурява флуиден сензорен интерфейс, като често се използват вибрационните възможности на устройството, за да се осигури хаптична обратна връзка на потребителя. Вътрешен хардуер, като акселерометри, жироскопи и сензори за близост, се използват от някои приложения, за да отговорят на допълнителни действия на потребителя, например коригиране на екрана от портретен към пейзаж в зависимост от това как е ориентирано устройството или позволяване на потребителя да управлява превозно средство в състезателна игра чрез завъртане на устройството, симулиращо управление на волан.

Устройствата с Android се зареждат от началния екран, основния "хъб" за навигация и информация на устройства с Android, аналогично на работния плот, който се намира на персонални компютри. Началните екрани на Android обикновено са съставени от икони на приложения и джаджи; иконите на приложения стартират свързаното приложение, докато джаджи показват на живо, автоматично актуализирано съдържание, като прогноза за времето, входяща поща на потребителя или новинарски тикер директно на началния екран. Началният екран може да се състои от няколко страници, между които потребителят може да плъзга напред-назад. Приложенията на трети страни, налични в Google Play и други магазини за приложения, могат значително да преначертаят началния екран и дори да имитират външния вид на други операционни системи, като Windows Phone. Повечето производители персонализират външния вид и функциите на своите Android устройства, за да се разграничат от своите конкуренти.

Приложенията („приложения“), които разширяват функционалността на устройствата (и трябва да са 64-битови), се пишат с помощта на комплекта за разработка на софтуер за Android (SDK) и често език за програмиране Kotlin, който заменя Java като предпочитан език на Google за Android разработка на приложения през май 2019 г. и първоначално беше обявена през май 2017 г. Java все още се поддържа (първоначално единствената опция за програми за потребителско пространство и често се смесва с Kotlin), както и C++. Java или други JVM езици, като Kotlin, могат да се комбинират с C/C++, заедно с избор от среди за изпълнение, които не са по подразбиране, които позволяват по-добра поддръжка на C++. Езикът за програмиране Go също се поддържа, макар и с ограничен набор от интерфейси за програмиране на приложения (API).

SDK включва изчерпателен набор от инструменти за разработка, включително дебъгер, софтуерни библиотеки, емулатор на слушалка, базиран на QEMU, документация, примерен код и уроци. Първоначално поддържаната от Google интегрирана среда за разработка (IDE) беше Eclipse с помощта на плъгин Android Development Tools (ADT); през декември 2014 г. Google пусна Android Studio, базиран на IntelliJ IDEA, като основна IDE за разработка на приложения за Android. Налични са и други инструменти за разработка, включително собствен комплект за разработка (NDK) за приложения или разширения на C или C++, Google App Inventor, визуална среда за начинаещи програмисти и различни междуплатформени рамки за мобилни уеб приложения. През януари 2014 г. Google представи рамка, базирана на Apache Cordova, за пренасяне на уеб приложения на Chrome HTML 5 към Android, обвити в собствена обвивка на приложения. Освен това Firebase беше придобита от Google през 2014 г., която предоставя полезни инструменти за разработчиците на приложения и уеб.

**6. Перспективи за развитието на Линукс и Андроид**

В тази част от документа ще бъдат представени недостатъците на операционните системи Линукс и Андроид, чието решаване би довело до подобрение на системите.

Недостатъци на Линукс:

* Липсата на стандартно издание затруднява работата на новите потребители, тъй като те изпитват трудност при избирането на едно от всички издания(дистрибуции). Отсъствието на стандартно издание значи, че общността е разделена, което усложнява поддръжката на съответното издание, което потребителят е избрал.
* Приложения, които не са налични за операционната система Линукс. Този проблем се обяснява с малкия пазарен дял на Линукс, спрямо Window и Mac.
* Липсата на собствен софтуер кара потребителите на Линукс да използват софтуер на трети страни, които не предоставят толкова добра функционалност. Пример за това са продуктите за обработка и редакция на видео.
* Проблем със съвместимостта с хардуерни устройства, в резултат на това, че

всички драйвери са вградени в самото ядро на Линукс, а ядрото не може да има драйвери за всеки хардуер. Това създава проблем с принтерите и WiFi картите, тъй като картите може да не са съвместими с Линукс и да се нуждаят от допълнителна конфигурация. [3]

Недостатъци на Андроид:

* Приложенията остават активни на фонов режим, което води до по-бързо изтощаване на клетъчна батерия и захранването на телефона.
* Трудност при разработването на приложения на операционна система Андроид, поради наличието на много размери на дисплея. Това затруднява дизайнерите и разработчиците да направят адаптирано приложение за всички устройства.
* Смартфоните с по-ниски спецификации работят бавно, поради използването на множество приложения, което води до по-бързо прегряване.
* Лоша защита от вируси, което може да доведе до проблеми с извличане на лична информация. [4]

Това са част от недостатъците на двете операционни системи, излагането на проблемите в даден продукт могат да подобрят неговото бъдещо разработване с цел улеснение на потребителите и постигане на повече удобство и сигурност при работа.

Библиография:

[1] Developers, A., 2011. What is android?. *Dosegljivo: http://www. academia. edu/download/30551848/andoid--tech. pdf*.

[2] Alian, M., Guang, S. and Teng, B., 2011. Android operating system.

[3] <https://kerneltips.com/top-10-advantages-and-disadvantages-of-linux>

[4] <https://appsgeyser.com/blog/disadvantages-of-android-operating-system/>

[5] <https://www.educba.com/linux-vs-android/>

[6] <https://www.geeksforgeeks.org/difference-between-linux-and-android/>

[7] <http://www.differencebetween.net/technology/software-technology/difference-between-android-and-linux/>