**Estado del Arte: Linux From Scratch**

**Introducción**

El presente estado del arte se centra en el análisis de herramientas y avances relacionados con el desarrollo del proyecto Linux From Scratch (LFS) y de herramientas similares dentro del ámbito de los sistemas operativos. El objetivo es comprender las tendencias actuales y los desarrollos más recientes en este campo, abarcando el periodo de los últimos diez años.

El proyecto Linux From Scratch busca proporcionar un vasto alcance del funcionamiento interno de un sistema operativo Linux. Permite a los usuarios compilar y configurar un sistema operativo completo desde el código fuente, lo que resulta en un mayor control y personalización. Este enfoque también fomenta un aprendizaje práctico de los conceptos fundamentales de los sistemas operativos.

**Marco Teórico**

**Teorías Fundamentales sobre Sistemas Operativos**

Los sistemas operativos (SO) son el software principal que gestiona los recursos del hardware y proporciona servicios a los programas de usuario. Existen varias teorías que abordan los conceptos fundamentales de los sistemas operativos, desde su estructura hasta la gestión de procesos y memoria. Una de las principales teorías que sustentan el desarrollo de LFS es la teoría del control de recursos, que establece que un sistema operativo debe administrar eficientemente los recursos del sistema (CPU, memoria, dispositivos de entrada/salida, etc.) para asegurar que el software y los usuarios puedan operar de manera eficiente.

Otro concepto clave en el marco de los sistemas operativos es la teoría de la abstracción, que sugiere que los sistemas operativos proporcionan una capa de abstracción entre el hardware y el software para facilitar su interacción. Esto es central en LFS, ya que los usuarios configuran y ajustan estas interacciones al construir el sistema desde cero, asegurando que el sistema final esté optimizado para sus necesidades.

**Enfoque de Construcción desde Cero**

El enfoque de construcción desde cero es uno de los pilares fundamentales de LFS. A diferencia de las distribuciones de Linux convencionales, que proporcionan un sistema preconfigurado y listo para usar, LFS permite a los usuarios aprender y comprender cómo se construye un sistema operativo, desde el núcleo hasta las aplicaciones más básicas. Este enfoque está alineado con la teoría del aprendizaje experiencial, que sugiere que el verdadero aprendizaje ocurre cuando los individuos están activamente involucrados en la creación y resolución de problemas.

Construir un sistema desde el código fuente también se conecta con la teoría de la personalización del software, que plantea que los usuarios deben ser capaces de adaptar el sistema operativo a sus necesidades específicas. Al permitir la compilación de paquetes de software individuales y la configuración del sistema según los requisitos del usuario, LFS otorga un control total sobre el entorno, lo cual es esencial para aquellos que buscan un sistema operativo altamente personalizado.

**Teoría del Control y la Gestión de Dependencias**

Uno de los desafíos más importantes al construir un sistema operativo es la correcta gestión de dependencias. Los paquetes de software suelen depender de bibliotecas o componentes específicos para funcionar correctamente. La teoría de la gestión de dependencias sugiere que un sistema operativo debe gestionar estas interdependencias de manera efectiva para garantizar la estabilidad y funcionalidad del sistema. En el contexto de LFS, la construcción manual de paquetes y la configuración de estas dependencias es una parte esencial del proceso. Los usuarios deben tener un entendimiento de las relaciones entre los paquetes y cómo estas interacciones afectan el rendimiento general del sistema.

**Systemd: El Sistema de Inicialización Moderno**

El uso de Systemd en LFS 12.2 refleja una tendencia importante en el mundo de Linux hacia sistemas de inicialización más modernos y complejos. Systemd se encarga de la gestión de servicios y procesos en un sistema Linux, y reemplaza sistemas de inicialización tradicionales como SysVinit. Desde su adopción en distribuciones como Debian y Ubuntu, Systemd se ha convertido en un estándar. Systemd aborda la gestión de servicios, la supervisión de procesos y la configuración del arranque de una manera que permite un control más preciso sobre el inicio y funcionamiento del sistema. En LFS, la integración de Systemd no solo proporciona una plataforma robusta para gestionar los servicios, sino que también introduce un concepto más moderno y flexible de gestión de procesos y servicios.

**Distribuciones de Linux y la Filosofía del Software Libre**

La filosofía detrás de Linux y el software libre también es una parte fundamental del LFS. El concepto de software libre, definido por Richard Stallman y la Free Software Foundation, subraya la importancia de la libertad del usuario para ejecutar, estudiar, modificar y distribuir software. Este principio guía el proyecto LFS, que permite a los usuarios construir un sistema operativo completamente libre y personalizable, brindando una comprensión profunda del funcionamiento de cada componente del sistema.

**Desarrollo**

Linux From Scratch (LFS) es un proyecto que ofrece una metodología detallada para construir un sistema operativo Linux personalizado desde cero, utilizando exclusivamente código fuente. Esta iniciativa no solo fomenta un entendimiento profundo del sistema operativo, sino que también permite a los usuarios crear sistemas altamente optimizados y adaptados a necesidades específicas.

El proyecto se compone de varios subproyectos que expanden su funcionalidad, entre los cuales destacan Beyond Linux From Scratch (BLFS), Automated Linux From Scratch (ALFS), Multilib Linux From Scratch (MLFS) y Gaming Linux From Scratch (GLFS). BLFS, por ejemplo, permite extender un sistema básico de LFS incorporando herramientas y aplicaciones adicionales, mientras que ALFS introduce herramientas para automatizar el proceso de construcción, facilitando así la gestión de sistemas más complejos. Por otro lado, MLFS se centra en compatibilidad multiarquitectura, permitiendo ejecutar aplicaciones de 32 bits en sistemas de 64 bits, y GLFS está orientado a proporcionar un entorno optimizado para juegos.

Aunque LFS se centra en la construcción manual y detallada de un sistema Linux, existen otras herramientas y distribuciones que buscan objetivos similares a través de diferentes enfoques. Gentoo, por ejemplo, es una distribución basada en código fuente que combina un alto grado de personalización con herramientas automatizadas para simplificar la instalación y mantenimiento. Funtoo, derivado de Gentoo, ofrece mejoras adicionales en rendimiento y gestión de paquetes. T2 SDE y Exherbo, por su parte, son plataformas que permiten construir sistemas Linux desde cero, ofreciendo flexibilidad para adaptarlos a casos de uso específicos. Finalmente, Source Mage introduce un enfoque basado en "hechizos" para gestionar paquetes y construir sistemas personalizados, mientras mantiene la simplicidad como una de sus prioridades.

Estos proyectos además de destacar por su capacidad de personalización también fomentan comunidades activas que comparten conocimientos y experiencias. Genera discusiones fructíferas entre usuarios que comparten sus proyectos basados en LFS o en otras herramientas similares, lo que resalta la relevancia de estas iniciativas en el ámbito de la educación y el desarrollo profesional.

**Proyectos Actuales: Desarrollo de Habilidades Técnicas**

Linux es un sistema operativo robusto y versátil que ofrece un campo amplio para la experimentación y el aprendizaje. A lo largo de los años, muchos proyectos prácticos han permitido a los usuarios profundizar en diversos aspectos de la informática, desde la administración del sistema hasta el desarrollo de hardware. A continuación, se presentan algunos proyectos de Linux que no solo son educativos, sino que también son útiles para mejorar habilidades técnicas. Si bien es cierto, no se encontró información sobre personas que han desarrollado estos experimentos puntuales, pero de todos modos forman parte de un conjunto de ideas interesantes para lograr el propósito educativo.

* **Herramienta de Monitoreo del Sistema**: Este proyecto se centra en crear y configurar un sistema propio de monitorización en Linux. Es ideal para iniciarse en el análisis del rendimiento del sistema, ya que en lugar de depender de herramientas como htop, se busca desarrollar soluciones personalizadas para supervisar el estado del sistema. Esto implica aprender cómo interactúan los componentes del sistema operativo y cómo desarrollar y configurar servicios específicos en Linux. Además, el proyecto puede ampliarse para incluir monitoreo de redes, lo que ofrece la oportunidad de profundizar en el conocimiento sobre redes y sus componentes.
* **Espejo Inteligente**: Combinar una Raspberry Pi y Linux para crear un espejo interactivo. Se basa en el uso de un espejo bidireccional junto con una Raspberry Pi para mostrar información como el clima, la hora y la fecha, entre otros datos. A través de la instalación y personalización de los repositorios de MagicMirror desde GitHub, se pueden realizar ajustes que adapten el proyecto a necesidades específicas.
* **Teléfono Pi**: Este proyecto combina Linux y módulos GSM para construir un teléfono básico con una Raspberry Pi. Permite enviar y recibir mensajes de texto, además de realizar llamadas, similar a un teléfono inteligente sencillo. Se puede integrar una pantalla táctil y personalizar la interfaz para mostrar información útil, como la hora y la fecha, mientras se adquiere experiencia en el uso de módulos de comunicación y en la implementación de software de telefonía.

Es importante destacar que en la web hemos encontrado información sobre una persona que empezó el proceso de construcción de Linux From Scratch en una Raspberry Pi. A pesar de que LFS generalmente se asocia con arquitecturas de PC, es posible adaptarlo a la Raspberry Pi con algunos ajustes en la configuración y herramientas compatibles.

Asimismo, el LFS ha sido probado con diferentes versiones del kernel de Linux. Por ejemplo:

LFS 12.2-systemd 🡪 Linux 6.10.5

LFS 12.1-systemd 🡪 Linux 6.7.4

LFS 11.1-systemd 🡪 Linux 5.16.19

LFS 11.0 🡪 Linux 5.13.12

LFS 10.0 🡪 Linux 5.8.3

LFS 9.1 🡪 Linux 5.5.3

LFS 9.0-systemd 🡪 Linux 5.2.8

LFS 9.0 🡪 Linux 5.2.8

En el manual se mencionan distintos sistemas operativos host para comenzar la construcción del LFS como: Debian, OpenMandriva, Fedora, or openSUSE. En nuestro caso, hemos utilizado Rocky Linux 9.4 como host. De la misma manera, también conocemos allegados que lo han hecho utilizando Ubuntu.

**Referencias Bibliográficas**

* <https://linuxfromscratch.org/lfs/>
* <https://0xc0decafe.com/linux-from-scratch-is-it-worth-it>
* <https://www.muylinux.com/2023/09/04/linux-from-scratch-12/>
* <https://informatecdigital.com/linux/linux-from-scratch-linux-desde-cero-para-principiantes/>
* <https://keepcoding.io/blog/que-es-systemd/>
* <https://www.liquidweb.com/blog/what-is-systemctl-an-in-depth-overview/>
* <https://natnat1.medium.com/sysvinit-vs-systemd-88395a76e869>
* <https://www.gentoo.org/get-started/about/>
* <https://sourcemage.org/About>
* <https://www.geeksforgeeks.org/linux-project-ideas-for-beginners/>
* <https://forums.raspberrypi.com/viewtopic.php?t=284787>