

SECCIÓN 1: ESTADO DEL ARTE

1.1 Documento Completo

El documento tiene entre 6-8 páginas

Incluye bibliografía en formato APA o IEEE

Cubre investigaciones de los últimos 10 años

1.2 Contenido del Estado del Arte

Revisión de papers académicos sobre LFS

Comparación con alternativas (Gentoo, Buildroot, Yocto, Alpine)

Identificación de problemas y metodologías

Incluye proyectos académicos y empresariales recientes

Discusión de tendencias actuales

Análisis de beneficios del sistema a medida

1.3 Calidad Académica

Texto claro y bien estructurado

Referencias bibliográficas correctamente citadas

Análisis académico profundo (no solo documentación técnica)

SECCIÓN 2: INSTALACIÓN DEL LFS

2.1 Versión de LFS utilizada



2.2 Versión de Rocky Linux (Host)

Ubicación	us-central1-b
Imagen de origen del disco de arranque	rocky-linux-10-optimized-gcp-v20251017

2.3 Sistema Booteable

El sistema arranca de manera independiente

Se puede iniciar sesión correctamente

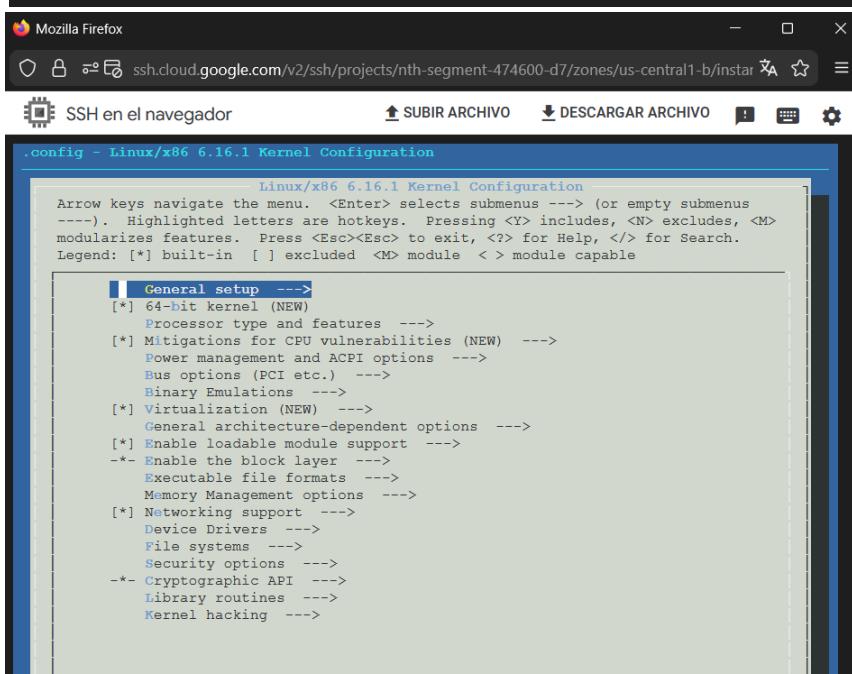
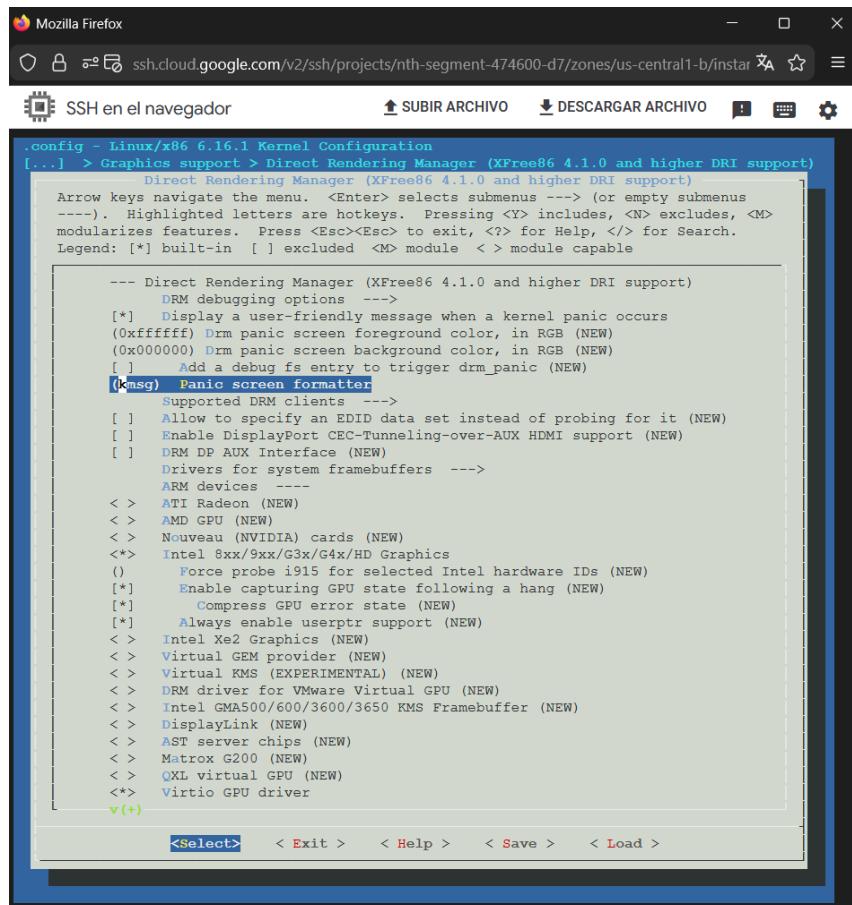
Sistema funcional sin depender del host

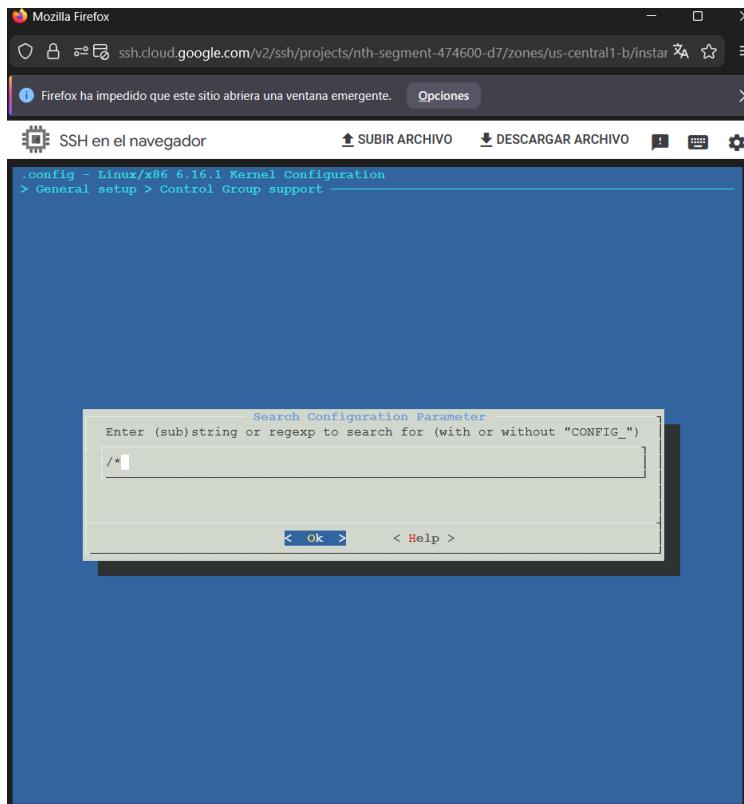
2.4 Documentación Técnica

Comandos principales documentados (En el registro de actividad)

Evidencia de compilación de paquetes

Configuración del kernel documentada





La configuración del kernel 6.12 para el proyecto Linux From Scratch 12.4 se centra en asegurar compatibilidad con virtualización, cumplir los requisitos mínimos de systemd y mantener un diseño monolítico y simple para el arranque. Esto implica habilitar funciones esenciales como IPC System V, cgroups y soporte completo para arquitectura x86_64, además de incluir drivers VirtIO para un rendimiento óptimo en entornos virtualizados. También se activan controladores críticos para discos (SCSI, virtio_blk), red (E1000, virtio_net) y el sistema de archivos ext4 integrado directamente en el kernel para evitar fallos de montaje durante el boot.

Asimismo, se habilitan opciones fundamentales para que systemd funcione como PID 1, como devtmpfs, inotify, signalfd, timerfd y epoll. Se incluyen los pseudo-filesystems /proc y sysfs, necesarios para exponer información del sistema al espacio de usuario. Finalmente, se evita compilar información de depuración para reducir el tamaño del kernel y acelerar la compilación, priorizando una configuración limpia y funcional.

Configuración de GRUB documentada

```
[paredesivan003@rocky-host-lfs ~]$ sudo su -
Last login: Tue Dec  2 22:12:38 UTC 2025 on pts/1
[root@rocky-host-lfs ~]# grub2-mkconfig -o /boot/grub2/grub.cfg
Generating grub configuration file ...
Found Linux From Scratch (12.4-systemd) on /dev/sdb2
Adding boot menu entry for UEFI Firmware Settings ...
done
[root@rocky-host-lfs ~]#
```

```

SSH en el navegador
inmod video_cirrus
}

serial --speed=115200
terminal_output serial console
if [ $feature_timeout_style = xy ] ; then
    set timeout_style=menu
    set timeout=30
# Fallback normal timeout code in case the timeout_style feature is
# unavailable.
else
    set timeout=30
fi
## END /etc/grub.d/00_header ###

## BEGIN /etc/grub.d/01_users ##
if [ -f ${prefix}/user.cfg ] ; then
    source ${prefix}/user.cfg
    if [ "$GRUB2_PASSWORD" ] ; then
        set superusers=root
        export superusers
        password_pbkdf2 root ${GRUB2_PASSWORD}
    fi
fi
## END /etc/grub.d/01_users ###

## BEGIN /etc/grub.d/08_fallback_counting ##
insmod increment
# Check if boot_counter exists and boot_success=0 to activate this behaviour.
if [ -n "${boot_counter}" -a "${boot_success}" = "0" ] ; then
    # If counter has ended, choose to boot rollback deployment,
    # i.e. default on corosync-based systems.
    if [ "${boot_counter}" = "0" -o "${boot_counter}" = "-1" ] ; then
        set default=
    else
        boot_counter=$((boot_counter-1))
    fi
-- INSERT --

```

71,17 29%

La configuración de GRUB consistió en la instalación del gestor de arranque directamente en el MBR del disco secundario (/dev/sdb) para mantener aislado el sistema anfitrión y permitir que el disco virtual arranque de forma independiente en cualquier máquina. El archivo grub.cfg se escribe manualmente, definiendo un menú simple con un único kernel y parámetros esenciales: la ruta al vmlinuz compilado, la partición raíz correcta (/dev/sdb2) y el montaje inicial en modo lectura para que systemd pueda realizar verificaciones de integridad. A diferencia de las distribuciones comerciales, donde la configuración se genera automáticamente, en LFS este proceso manual permite comprender con precisión cómo GRUB localiza y carga el kernel, reforzando el aprendizaje sobre el proceso de arranque.

Resumen de problemas y soluciones incluido (En el registro de actividad)

2.5 Capturas y Evidencias

Capturas del proceso de compilación.

```
(lts chroot) root:/sources$ tar -xvf man-pages-6.15.tar.xz
man-pages-6.15/.checkpatch.conf
man-pages-6.15/.gitignore
man-pages-6.15/.shellcheckrc
man-pages-6.15/AUTHORS
man-pages-6.15/CONTRIBUTING
man-pages-6.15/CONTRIBUTING.d/bugs
man-pages-6.15/CONTRIBUTING.d/external_pages
man-pages-6.15/CONTRIBUTING.d/git
man-pages-6.15/CONTRIBUTING.d/int
man-pages-6.15/CONTRIBUTING.d/mail
man-pages-6.15/CONTRIBUTING.d/patches/base
man-pages-6.15/CONTRIBUTING.d/patches/description
man-pages-6.15/CONTRIBUTING.d/patches/diff
man-pages-6.15/CONTRIBUTING.d/patches/patches
man-pages-6.15/CONTRIBUTING.d/patches/range-diff
man-pages-6.15/CONTRIBUTING.d/patches/sendmail
man-pages-6.15/CONTRIBUTING.d/patches/subject
man-pages-6.15/CONTRIBUTING.d/style/c
man-pages-6.15/CONTRIBUTING.d/style/man
man-pages-6.15/CPPLINT.cfg
man-pages-6.15/Changes
man-pages-6.15/Changes.old
man-pages-6.15/GNOMakefile
man-pages-6.15/INSTALL
man-pages-6.15/LICENSES/BSD-2-Clause.txt
man-pages-6.15/LICENSES/BSD-3-Clause.txt
man-pages-6.15/LICENSES/BSD-4-Clause-UC.txt
man-pages-6.15/LICENSES/CDDL-1.0-or-later.txt
man-pages-6.15/LICENSES/GPL-2.0-or-later.txt
man-pages-6.15/LICENSES/GPL-2.0-only.txt
man-pages-6.15/LICENSES/GPL-3.0-or-later.txt
man-pages-6.15/LICENSES/IGPL-3.0-linking-exception.txt
man-pages-6.15/LICENSES/IGPL-3.0-or-later.txt
man-pages-6.15/LICENSES/linux-man-pages-1-para.txt
man-pages-6.15/LICENSES/linux-man-pages-copyleft-2-para.txt
man-pages-6.15/LICENSES/linux-man-pages-copyleft-var.txt
man-pages-6.15/LICENSES/linux-man-pages-copyleft.txt
man-pages-6.15/README
man-pages-6.15/RELEASE
man-pages-6.15/SPONSORS
man-pages-6.15/etc/checkpatch/checkpatch.conf
man-pages-6.15/etc/clang-tidy/config.yaml
man-pages-6.15/etc/cppcheck/cppcheck.suppress
man-pages-6.15/etc/cpplint/cpplint.cfg
man-pages-6.15/etc/shellcheck/shellcheckrc
man-pages-6.15/1
man-pages-6.15/man/man1/diffman-git.1
man-pages-6.15/man/man1/getent.1
man-pages-6.15/man/man1/iconv.1
man-pages-6.15/man/man1/intro.1
man-pages-6.15/man/man1/ldd.1
man-pages-6.15/man/man1/locale.1
man-pages-6.15/man/man1/localedef.1
man-pages-6.15/man/man1/mansect.1
man-pages-6.15/man/man1/nemusage.1
man-pages-6.15/man/man1/nemusagestat.1
man-pages-6.15/man/man1/ntrace.1
man-pages-6.15/man/man1/pifman.1
man-pages-6.15/man/man1/pold.1
man-pages-6.15/man/man1/sorcman.1
man-pages-6.15/man/man1/sprof.1
```





```
(lfs chroot) root:/sources/glibc-2.42/build# ls
Makefile           configparms      glibid          libc.so.phdr   link-defines.h.d    rtld-offsets.h.d    sysvipc
Versions.all       conform          hesiod         libc.symlist   link-static-libc.out  rtld-sizes.h     tcb-offsets.h
Versions.def       cpu-features-offsets.h  iconv          libc_nondynamic_debug.map  link-static-libc.test-result  rtld-sizes.h.d  tcb-offsets.h.d
Versions.mk        cpio-features-offsets.h.d  libcryptodata  libc_nonshared.a  linkto[...]  runtime-linker.h  turnos
Versions.tmp       cstdlib          inet           libc.pic.a    lint-makefiles.out  runtime-linker.stamp  testroot.pristine
Versions.v         csu              intl           libc.pic.opts  lint-makefiles.test-result  setup           testroot.root
Versions.v.i       ctype             io             libc.pic.os    locale           shlib-versions.v    testrun.sh
abi-versions.h    debug             jmp_buf_macros.h  libc.pic.os.clean  locale-defines.h    shlib-versions.v.i  tests.sum
argp               debugglibc.sh   jmp_buf_macros.h.d  libdl.map        locale-defines.h.d  signalstack-offsets.h    time
assert            descr-const.h   jmp_buf_asp.h    libbio          localedata      signalstack-offsets.h.d  time64-compat.mk
bits               descr-const.h.d  jmp_buf_asp.h.d  libdl-macros.h  login           shlib-versions.v    time64-compat.mk.i
c++-types-check.cut dirent            libdl-compat-choose.h  libdm.map        login           signalstack-offsets.h  time64-compat.mk.i
c++-types-check.test-result dl-tunable-list.h  libdlbrokenLocale.map  libhwvec.map    math             socket           tcb-offsets.h
cgetsets          dl-tunable-list.stamp  libfnl.map      libhwvec.mk    mathvec          socket           tcb-offsets.h
check-installed-headers-c.out dummy.c          libc-abis.h   libnl.map       misc            stamp.o          tcb-offsets.h
check-installed-headers-c-test-result dummy.o          libc-abis.stamp  libnss_compat.map  nis             stamp.o          tcb-offsets.h
check-installed-headers-cxx.out dummy.o.dt      libc-modules.h  libnss_db.map   nptl            stamp.os          tcb-offsets.h
check-installed-headers-cxx-test-result dummy.o.dt      libc-modules.stamp  libnss_dns.map  stdio-common    stamp.os          tcb-offsets.h
check-local-headers.out elf              libc-pic.os    libnss_hesiod.map  nszd           stdio           versions.stmp
check-local-headers.test-result features-offsets.h  libc-dynsym   libnss_hesiod.map  nszd           stdio           wcsmbcs
check-wrapper-headers.out first-versions.h  libc.map       libpthread.read.map  nszd           stdio           wctype
check-wrapper-headers.test-result format.lsd      libc.so       libresolv.map  pc              sunrpc          wctype
cmath              gcc-macros.h   libc.so.6      librbrt.map    posix           support          wctype
config.h          gcc-macros.h.in  libc.so.dyn   libthread_db.map  resource        sysd-rules
config.log         gmon             libc.so.dynsym  libutil.map    rt              sysd-sorted
config.make       gnu              libc.so.imeprel  link-defines.h  rtld-offsets.h  sysd-syscalls
config.status     lfs              libc.so.imeprel  link-defines.h  rtld-offsets.h  sysd-versions
(lfs chroot) root:/sources/glibc-2.42/build#
```

Captura del sistema arrancado

```
SSH en el navegador
GNU/Linux, Linux 6.16.1-lfs-12.4-systemd (on /dev/sdb2)
UEFI Firmware Settings
```

```
Starting Virtual Console Setup...
[ OK ] Adding 563196k swap on /dev/sdb3. Priority:1 extents:1 across:563196k SS
[ OK ] Activated swap '/dev/sdb3'.
[ OK ] Reached target Swap.
[ OK ] Finished Enable Persistent Storage in systemd-networkd.
[ OK ] Finished File System Check on /dev/sd4.
[ OK ] Finished Virtual Console Setup.
[ OK ] Mounting /home...
[ OK ] Mounting Temporary Directory /tmp...
[ OK ] Reached target Containers.
[ OK ] Mounted Temporary Directory /tmp.
[ OK ] EXT4-fs (sd4): mounted filesystem atf6f80f6-4ed1-4e5c-9337-02306b4fc679 r/w with ordered data mode. Quota mode: none.
[ OK ] Mounted /home.
[ OK ] Reached target Local File Systems.
[ OK ] Listening on System Extension Image Management.
[ OK ] Starting Create System Files and Directories...
[ OK ] Finished Create System Files and Directories.
[ OK ] Starting Record System Boot/shutdown in UTMP...
[ OK ] Finished Record System Boot/shutdown in UTMP.
[ OK ] Started Target System Initialization.
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timer Units.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Listening on OpenSSH Server Socket (tmux-ssh-generator, AF_UNIX Local).
[ OK ] Listening on Hostname Service Socket.
[ OK ] Reached target Socket Units.
[ OK ] Reached target Basic System.
[ OK ] Starting D-Bus System Message Bus...
[ OK ] Started OpenSSH Daemon.
[ OK ] Starting User Login Management...
[ OK ] Starting Permit User Sessions...
[ OK ] Finished Permit User Sessions.
[ OK ] Started Getty on ttyl.
[ OK ] Started Serial Getty on ttys0.
[ OK ] Reached target Login Prompts.
[ OK ] Started D-Bus System Message Bus.
[ OK ] Starting Record Runlevel Change in UTMP...
[ OK ] Started User Login Management.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
[ OK ] Starting Record Runlevel Change in UTMP...
[ OK ] Finished Record Runlevel Change in UTMP...
[ OK ] Started Hostname Service.
```

lfs login: []

```
[ OK ] Finished Virtual Console setup.
      Mounting Temporary Directory /tmp...
[ OK ] Reached target Containers.
[ OK ] Mounted Temporary Directory /tmp.
[ OK ] Reached target Local File Systems.
[ OK ] Listening on System Extension Image Management.
      Starting Create System Files and Directories...
[ OK ] Finished Create System Files and Directories.
      Starting Record System Boot/Shutdown in UTMP...
[ OK ] Finished Record System Boot/Shutdown in UTMP.
[ OK ] Reached target System Initialization.
[ OK ] Started Daily Cleanup of Temporary Directories.
[ OK ] Reached target Timer Units.
[ OK ] Listening on D-Bus System Message Bus Socket.
[ OK ] Listening on Hostname Service Socket.
[ OK ] Reached target Socket Units.
[ OK ] Reached target Basic System.
      Starting D-Bus System Message Bus...
[ OK ] Started Getty on ttyl.
[ OK ] Started Serial Getty on ttys0.
[ OK ] Reached target Login Prompts.
      Starting User Login Management...
[ OK ] Started User Login Management.
[ OK ] Started D-Bus System Message Bus.
[ OK ] Reached target Multi-User System.
[ OK ] Reached target Graphical Interface.
      Starting Record Runlevel Change in UTMP...
      Starting Hostname Service...
[ OK ] Finished Record Runlevel Change in UTMP.
[ OK ] Started Hostname Service.

lfs login: root
Contraseña:
Sin correo.
-bash-5.3# pw
-bash: pw: orden no encontrada
-bash-5.3# pwd
/root
-bash-5.3# cd /
-bash-5.3# pwd
/
-bash-5.3# swapoff -a
-bash-5.3# ls -l /dev/sdb3
brw-rw---- 1 root disk 8, 19 dic  2 22:22 /dev/sdb3
-bash-5.3# mkswap /dev/sdb3
```

Logs relevantes incluidos (En el registro de actividad)

SECCIÓN 3: SYSTEMD (OBLIGATORIO)

3.1 Implementación de Systemd

El sistema usa systemd como init (Si)

Systemd aparece como PID 1 (Si)

3.2 Verificación y Evidencias (Todas estas verificaciones se hacen con el script check_server.sh, revisar la guia de pruebas)

Incluye salida del comando systemctl

Incluye salida de pstree mostrando systemd como PID 1

Captura de systemctl status

3.3 Servicio Implementado

Nombre del servicio demostrado (SSH, nginx, etc.)

3.4 Evidencia del Servicio

Al menos un servicio levantado con systemd

Evidencia de que el servicio está activo

Comandos de habilitación/inicio documentados (En la guia de pruebas)

3.5 Documentación de Systemd

Logs o capturas de configuración de unidades

Documentación de archivos .service creados/modificados (En la guia de pruebas)

```
root@lfs:~# pstree -p | head -20
systemd(1)---agetty(271)
|   +--agetty(272)
|   +--dbus-daemon(266)
|   +--sshd(267)---sshd-session(296)---sshd-session(314)---bash(315)---head(780)
|   |   +--pstree(779)
|   +--systemd(303)---(sd-pam)(306)
|   +--systemd-journal(134)
|   +--systemd-logind(268)
|   +--systemd-network(207)
|   +--systemd-nsresou(164)---systemd-nsresou(712)
|   |   +--systemd-nsresou(713)
|   |   +--systemd-nsresou(716)
|   |   +--systemd-nsresou(717)
|   |   +--systemd-nsresou(718)
|   +--systemd-oomd(139)
|   +--systemd-resolve(165)
|   +--systemd-timesyn(193)---{systemd-timesyn}(196)
|   +--systemd-udevd(194)
root@lfs:~# journalctl
dic 02 21:39:29 lfs kernel: Linux version 6.16.1 (root@rocky-host-lfs) (gcc (GCC) 15.2.0, GNU ld (GNU Binutils) 2.45) #1 SMP PREEMPT_DYNAMIC Tue Dec 2 20:52:29 UTC 2025
dic 02 21:39:29 lfs kernel: Command line: BOOT_IMAGE=/boot/vmlinuz-6.16.1-lfs-12.4-systemd root=/dev/sdb2 ro net.ifnames console=ttyS0
dic 02 21:39:29 lfs kernel: [Firmware Bug]: TSC doesn't count with P0 frequency!
dic 02 21:39:29 lfs kernel: BIOS-provided physical RAM map:
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000000000-0x000000000000ffff] reserved
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000000000-0x0000000000054fff] usable
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000055000-0x000000000005ffff] reserved
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000060000-0x0000000000097fff] usable
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000098000-0x000000000009ffff] reserved
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000000100000-0x000000000bd2e6fff] usable
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bd2e7000-0x000000000bd2f0fff] ACPI data
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bd2f1000-0x000000000bf8ecfff] usable
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bf8ed000-0x000000000bfbcfcfff] reserved
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bfbd6d000-0x000000000bfbd7eff] ACPI data
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bfbd7f000-0x000000000bfbdfefff] ACPI NVS
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bfbff000-0x000000000bfffdffff] usable
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x000000000bffe0000-0x000000000bfffffff] reserved
dic 02 21:39:29 lfs kernel: BIOS-e820: [mem 0x0000000010000000-0x00000000083fffffff] usable
```

```
root@lfs:~# /srv/http-lfs/start_server.sh
===== Iniciando Servidor HTTP TP LFS =====

Recargando systemd ...
Iniciando servicio lfs-http ...

===== Estado del Servicio =====
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pre
set: enabled)
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 1s ago
     Invocation: b8f8943084054b599ebfd1f910586f57
      Main PID: 394 (python3)
        Tasks: 1 (limit: 38449)
       Memory: 8.9M (peak: 8.9M)
         CPU: 81ms
        CGroup: /system.slice/lfs-http.service
                  └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

===== Información de Conexión =====
* Accede desde:
  Local: http://localhost:8081/
  Red: http://<tu-ip>:8081/

* Comandos útiles:
  Ver logs: journalctl -u lfs-http -f
  Ver estado: systemctl status lfs-http
  Parar: systemctl stop lfs-http
root@lfs:~#
```

```
C:\Program Files\WindowsAp < + | > - □ ×

Ver estado: systemctl status lfs-http
Parar: systemctl stop lfs-http
root@lfs:/# /srv/http-lfs/check_server.sh
== Verificación Servidor HTTP TP LFS ==
Fecha: jue 11 dic 2025 23:09:28 -03

1. 🐀 PID 1 del sistema:
systemd
✓ Systemd es el PID 1

2. ⚙ Estado del servicio lfs-http:
✓ ACTIVO
Main PID: 394 (python3)
    Tasks: 1 (limit: 38449)
    Memory: 8.9M (peak: 8.9M)
    CPU: 85ms
    CGroup: /system.slice/lfs-http.service
        └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

3. 🌐 Puerto 8081:
✓ EN ESCUCHA
LISTEN 0      0          0.0.0.0:8081      0.0.0.0:*      users:(("python3",
pid=394,fd=3))

4. ✎ Prueba de conexión:
✓ Servidor responde (via Python)

5. 📄 Logs recientes:
dic 11 17:05:12 lfs systemd[1]: lfs-http.service: Deactivated successfully.
dic 11 17:05:12 lfs systemd[1]: Stopped Servidor HTTP TP LFS.
-- Boot d2e970741c4e4d31a532748f46b09d37 --
dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

6. 🌐 Información de acceso:
    URL: http://:8081/
    URL local: http://localhost:8081/
root@lfs:/#
```

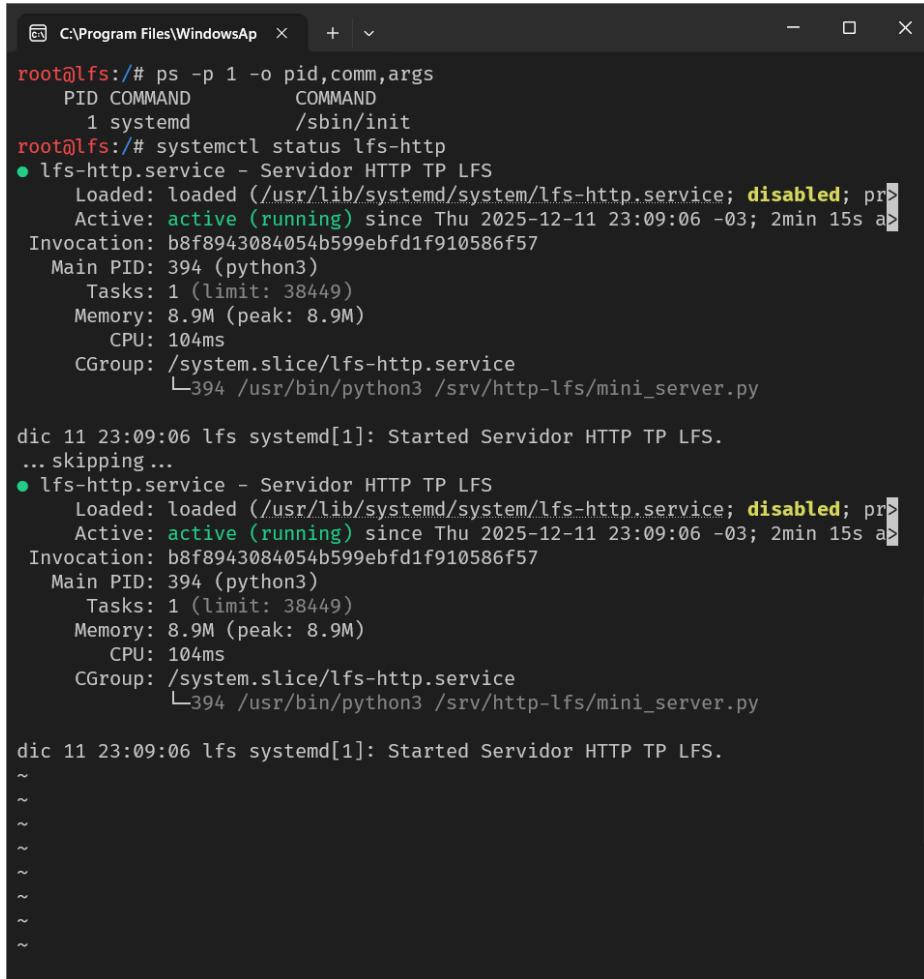
```
2. ⚙ Estado del servicio lfs-http:
✓ ACTIVO
Main PID: 394 (python3)
    Tasks: 1 (limit: 38449)
    Memory: 8.9M (peak: 8.9M)
    CPU: 85ms
    CGroup: /system.slice/lfs-http.service
        └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

3. 🌐 Puerto 8081:
✓ EN ESCUCHA
LISTEN 0      0          0.0.0.0:8081      0.0.0.0:*      users:(("python3",
pid=394,fd=3))

4. ✎ Prueba de conexión:
✓ Servidor responde (via Python)

5. 📄 Logs recientes:
dic 11 17:05:12 lfs systemd[1]: lfs-http.service: Deactivated successfully.
dic 11 17:05:12 lfs systemd[1]: Stopped Servidor HTTP TP LFS.
-- Boot d2e970741c4e4d31a532748f46b09d37 --
dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.

6. 🌐 Información de acceso:
    URL: http://:8081/
    URL local: http://localhost:8081/
root@lfs:/# python3 -c "import urllib.request; print(urllib.request.urlopen(
'http://localhost:8081/').read().decode()[:200])"
<html>
<head><title>LFS Systemd TP</title></head>
<body>
<h1>✓ LFS con Systemd funcionando!</h1>
<p><strong>PID 1:</strong> systemd</p>
```

A screenshot of a terminal window titled 'C:\Program Files\WindowsAp'. The window contains several lines of command-line output from a root user on a system named 'lfs'. The commands run include 'ps -p 1 -o pid,comm,args', 'systemctl status lfs-http', and 'lsof -i :80'. The output shows the PID, command, and resource usage for the 'systemd' process and the 'lfs-http' service, which is listed as active and running. The terminal also displays log entries from 'systemd[1]' starting the 'lfs' service at 23:09:06 on December 11, 2025.

```
root@lfs:/# ps -p 1 -o pid,comm,args
PID COMMAND      COMMAND
 1 systemd      /sbin/init
root@lfs:/# systemctl status lfs-http
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pr>
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 2min 15s a>
     Invocation: b8f8943084054b599ebfd1f910586f57
   Main PID: 394 (python3)
     Tasks: 1 (limit: 38449)
    Memory: 8.9M (peak: 8.9M)
       CPU: 104ms
      CGroup: /system.slice/lfs-http.service
              └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.
... skipping...
● lfs-http.service - Servidor HTTP TP LFS
   Loaded: loaded (/usr/lib/systemd/system/lfs-http.service; disabled; pr>
   Active: active (running) since Thu 2025-12-11 23:09:06 -03; 2min 15s a>
     Invocation: b8f8943084054b599ebfd1f910586f57
   Main PID: 394 (python3)
     Tasks: 1 (limit: 38449)
    Memory: 8.9M (peak: 8.9M)
       CPU: 104ms
      CGroup: /system.slice/lfs-http.service
              └─394 /usr/bin/python3 /srv/http-lfs/mini_server.py

dic 11 23:09:06 lfs systemd[1]: Started Servidor HTTP TP LFS.
~
~
~
~
~
~
~
```

Reflexión sobre beneficios de systemd vs métodos antiguos.

Construir Linux From Scratch utilizando systemd ofrece una formación mucho más alineada con las prácticas modernas y estándares de la industria, ya que este init es el estándar en la mayoría de las distribuciones actuales y permite comprender cómo funcionan realmente los sistemas usados en entornos profesionales. Su capacidad de paralelizar el arranque y activar servicios bajo demanda mejora la eficiencia, mientras que su gestión declarativa de dependencias reduce errores y enseña a trabajar con *unit files*. El control de procesos con cgroups aporta trazabilidad y control preciso de procesos, journald unifica y potencia el análisis de logs, y herramientas como *hostnamectl* o *timedatectl* estandarizan configuraciones antes dispersas. Esto convierte el proyecto LFS en un entrenamiento moderno para administrar sistemas complejos y de alto rendimiento.

SECCIÓN 4: REGISTRO DE PROCESO / DIARIO

4.1 Formato y Ubicación

Archivo incluido en repositorio GitLab

shell: [Red343/flsh](#)

Documentación LFS: [Red343/Documentaci-n-LFS: LFS](#)

Formato adecuado (Markdown o PDF)

4.2 Contenido del Diario

- Fecha y hora registradas en cada sesión
- Tareas realizadas documentadas
- Comandos principales ejecutados
- Problemas encontrados descritos
- Soluciones aplicadas documentadas
- Participantes de cada sesión identificados

4.3 Evidencias Visuales

Capturas de pantalla de errores
Fragmentos de logs relevantes
Evidencias de avances progresivos

4.4 Control de Versiones

Commits con mensajes claros y consistentes
Se identifica quién realizó qué en cada commit
Narración en primera persona plural

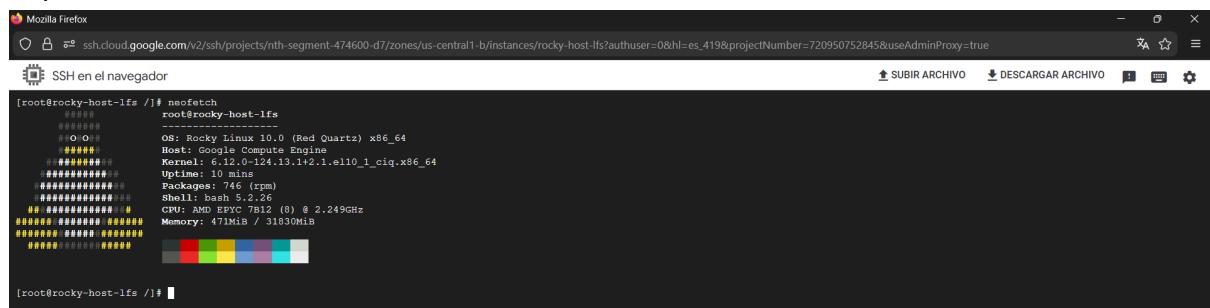
4.5 Calidad del Proceso

Muestra el proceso real de construcción
Incluye reflexión sobre errores y aprendizajes
Evidencia de trabajo grupal colaborativo

SECCIÓN 5: HOST Y REQUISITOS TÉCNICOS

5.1 Informe Técnico del Host

Captura de neofetch



Captura de lscpu

SSH-in-browser

```
[root@rocky-host-lfs ~]# lscpu
Architecture:          x86_64
CPU op-mode(s):        32-bit, 64-bit
Address sizes:         46 bits physical, 48 bits virtual
Byte Order:            Little Endian
CPU(s):                8
On-line CPU(s) list:  0-7
Vendor ID:             GenuineIntel
BIOS Vendor ID:       Google
Model name:            Intel(R) Xeon(R) CPU @ 2.20GHz
BIOS Model name:      CPU @ 2.0GHz
BIOS CPU family:      1
CPU Family:            6
Model:                 79
Thread(s) per core:   2
Core(s) per socket:   4
Socket(s):             1
Stepping:              0
BogoMIPS:              4399.99
Flags:                 fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov pat pse36 clflush mmx fxsr sse sse2 ss ht syscall nx pdpe1gb rdtscp lm constant_tsc rep_good nopl xtpr nonstop_tsc cpuid tsc_known_freq pni pclmulqdq ssse3 fma cx16 pcid sse4_1 ss e4_2 x2apic movbe popcnt aes xsave avx f16c rdrand hypervisor lahf lm abm 3dnowprefetch pti ssbd ibrs ibpb stibp fsgsbase tsc_adjust bmi1 hle avx2 smep bmi2 erms in vpcid rtm rdseed adx smap xsaveopt arat md_clear arch_capabilities

Virtualization features:
Hypervisor vendor:    KVM
Virtualization type:  full

Caches (sum of all):
L1d:                  128 KiB (4 instances)
L1i:                  128 KiB (4 instances)
L2:                   1 MiB (4 instances)
L3:                   55 MiB (1 instance)

NUMA:
NUMA node(s):          1
NUMA node0 CPU(s):     0-7

Vulnerabilities:
Gather data sampling: Not affected
Indirect target selection: Mitigation; Aligned branch/return thunks
Itlb multithit:        Not affected
Ilfif:                 Mitigation; PTE Inversion
Mdts:                  Mitigation; Clear CPU buffers; SMT Host state unknown
Meltdown:              Mitigation; PTI
Mmio stale data:       Vulnerable; Clear CPU buffers attempted, no microcode; SMT Host state unknown
Req file data sampling: Not affected
Retbleed:               Mitigation; IBRS
Spec rstack overflow:  Not affected
Spec store bypass:     Mitigation; Speculative Store Bypass disabled via prctl
Spectre v1:              Mitigation; usercopy/swaps barriers and _user pointer sanitization
Spectre v2:              Mitigation; IBRS; IBPB conditional; STIBP conditional; RSB filling; PBRSB-eIBRS Not affected; BHI SW loop, KVM SW loop
Srbdts:                 Not affected
Tsaa:                  Not affected
Txm sync abort:        Mitigation; Clear CPU buffers; SMT Host state unknown
Vmcspec:                Not affected

[root@rocky-host-lfs ~]#
```

Captura de free -h

SSH-in-browser

UPLOAD FILE DOWNLOAD FILE

```
[root@rocky-host-lfs ~]# free -h
total        used         free        shared      buff/cache   available
Mem:      31Gi       939Mi      29Gi       8.6Mi       904Mi      30Gi
Swap:      0B          0B         0B

[root@rocky-host-lfs ~]#
```

Captura de df -h

SSH-in-browser

UPLOAD FILE DOWNLOAD FILE

```
[root@rocky-host-lfs ~]# df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda2       20G   17G   3.5G  83% /
devtmpfs        16G     0   16G   0% /dev
tmpfs          16G     0   16G   0% /dev/shm
efivarsfs       56K   24K   27K  48% /sys/firmware/efi/efivars
tmpfs          6.3G   8.6M   6.3G  1% /run
tmpfs          1.0M     0   1.0M  0% /run/credentials/systemd-journald.service
/dev/sdal       200M   8.6M   192M  5% /boot/efi
tmpfs          1.0M     0   1.0M  0% /run/credentials/getty@tty1.service
tmpfs          1.0M     0   1.0M  0% /run/credentials/serial-getty@ttyS0.service
tmpfs          3.2G   4.0K   3.2G  1% /run/user/1003

[root@rocky-host-lfs ~]#
```

5.6 Cumplimiento de Requisitos

Cumple requisitos mínimos de CPU (4+ núcleos)

Cumple requisitos mínimos de RAM (8+ GB)

Cumple requisitos mínimos de almacenamiento (20+ GB)

Espacio suficiente asignado a /mnt/lfs (10+ GB)

```
[root@rocky-host-lfs ~]# lsblk
NAME   MAJ:MIN RM  SIZE RO TYPE MOUNTPOINTS
sda      8:0    0   20G  0 disk 
└─sda1   8:1    0  200M  0 part /boot/efi
└─sda2   8:2    0 19.8G  0 part /
sdb      8:16   0   50G  0 disk 
└─sdb1   8:17   0     2M  0 part 
└─sdb2   8:18   0   30G  0 part /mnt/lfs
└─sdb3   8:19   0  550M  0 part 
└─sdb4   8:20   0 19.3G  0 part
```

5.7 Justificación y Evidencias

Justificación clara del cumplimiento de requisitos

Evidencia de funcionamiento correcto del entorno

Documentación de tiempos de compilación

Sin errores por falta de recursos

SECCIÓN 6: OPCIONES INNOVADORAS

Opción 1 - Contenedores

LFS empaquetado en imagen de contenedor

Dockerfile o Podmanfile incluido

Contenedor ejecutable y funcional

Explicación de limitaciones documentada

Reflexión sobre virtualización vs contenerización

#Las pruebas del Docker se encuentran en el archivo de pruebas dentro del repositorio

Github#

Opción 2 - Servicio en Red

Servicio funcional (SSH, lighttpd, nginx, etc.)

Servicio accesible desde la red

Configuración documentada

Pruebas de conectividad incluidas

Capturas de servicio activo

#Las pruebas del Servicio en Red se encuentran en el archivo de pruebas dentro del

repositorio Github#