



# LFS Systemd - Servicio HTTP de Prueba

Este servicio demuestra que **systemd funciona como PID 1** en un sistema LFS (Linux From Scratch).



## Características

- Servidor HTTP simple en Python
  - Gestionado por systemd
  - Muestra información del sistema (PID 1, uptime, etc.)
  - Perfecto para demostraciones del TP de LFS
- 



## Estructura de Archivos

/srv/http-lfs/

```
|— mini_server.py  # Servidor HTTP principal
|— start_server.sh # Script de inicio manual
|— stop_server.sh  # Script de parada manual
```

/usr/lib/systemd/system/

```
└─ lfs-http.service # Definición del servicio systemd
```

---



## Instalación y Configuración

### 1. Crear directorios y scripts

```
# Crear directorio principal
mkdir -p /srv/http-lfs
cd /srv/http-lfs
```

### 2. Crear el servidor HTTP (mini\_server.py)

```
cat > /srv/http-lfs/mini_server.py << 'EOF'
#!/usr/bin/env python3
import http.server
import socketserver
import os
import socket
import time
```

PORT = 8081

```
class LFSHandler(http.server.SimpleHTTPRequestHandler):
    def do_GET(self):
        if self.path == '/':
            # Leer PID 1
            try:
                with open('/proc/1/comm', 'r') as f:
                    pid1 = f.read().strip()
            except:
                pid1 = "Desconocido"

            # Obtener uptime
            try:
                with open('/proc/uptime', 'r') as f:
                    uptime_sec = float(f.read().split()[0])
                    horas = int(uptime_sec // 3600)
                    minutos = int((uptime_sec % 3600) // 60)
                    uptime = f"{horas}h {minutos}m"
            except:
                uptime = "Desconocido"

            # Generar HTML
            html = f"""<!DOCTYPE html>
<html>
<head>
<title>LFS Systemd TP</title>
<style>
body {{ font-family: Arial, sans-serif; margin: 40px; }}
.container {{ max-width: 800px; margin: 0 auto; }}
.header {{ background: #4CAF50; color: white; padding: 20px; border-radius: 5px; }}
.info {{ background: #f5f5f5; padding: 15px; margin: 20px 0; border-left: 4px solid
#2196F3; }}
.status {{ color: #4CAF50; font-weight: bold; }}
</style>
</head>
<body>
<div class="container">
<div class="header">
<h1>✅ LFS con Systemd - TP</h1>
<p>Servicio HTTP de demostración</p>
</div>

<div class="info">
<h2>Información del Sistema</h2>
<p><strong>Hostname:</strong> {socket.gethostname()}</p>
<p><strong>PID 1 (Init):</strong> <span class="status">{pid1}</span></p>
<p><strong>Uptime:</strong> {uptime}</p>

```

```

        <p><strong>Fecha/Hora:</strong> {time.strftime('%Y-%m-%d %H:%M:%S')}</p>
        <p><strong>Puerto:</strong> {PORT}</p>
    </div>

```

```

    <p>Este servicio demuestra que systemd está funcionando como PID 1 en el sistema
    LFS.</p>

```

```

    <div class="info">
        <h3>Comandos útiles:</h3>
        <pre>systemctl status lfs-http
journalctl -u lfs-http -f
curl http://localhost:{PORT}</pre>
    </div>
</div>
</body>
</html>""

```

```

        self.send_response(200)
        self.send_header('Content-type', 'text/html')
        self.end_headers()
        self.wfile.write(html.encode())
    else:
        # Para otras rutas, servir archivos estáticos
        super().do_GET()

def log_message(self, format, *args):
    # Log simplificado
    print(f"[{time.strftime('%H:%M:%S')}] {self.address_string()} - {format % args}")

def main():
    # Cambiar al directorio del script
    os.chdir(os.path.dirname(os.path.abspath(__file__)))

    # Crear página de índice por defecto
    with open('index.html', 'w') as f:
        f.write("<h1>Redirigiendo...</h1><p>Usa la ruta raíz (/) para ver la información del
sistema.</p>")

    # Iniciar servidor
    with socketserver.TCPServer(("", PORT), LFSHandler) as httpd:
        print(f"[{time.strftime('%Y-%m-%d %H:%M:%S')}] Servidor iniciado en puerto {PORT}")
        print(f"[{time.strftime('%Y-%m-%d %H:%M:%S')}] Accede en: http://localhost:{PORT}/")
        httpd.serve_forever()

if __name__ == '__main__':
    main()
EOF

```

```
chmod +x /srv/http-lfs/mini_server.py
```

### 3. Crear servicio systemd (lfs-http.service)

```
cat > /usr/lib/systemd/system/lfs-http.service << 'EOF'
[Unit]
Description=Servidor HTTP TP LFS - Demostración Systemd
Documentation=https://www.linuxfromscratch.org
After=network.target
Wants=network.target

[Service]
Type=simple
User=root
WorkingDirectory=/srv/http-lfs
ExecStart=/usr/bin/python3 /srv/http-lfs/mini_server.py
Restart=on-failure
RestartSec=5
StandardOutput=journal
StandardError=journal

# Configuración de seguridad básica
NoNewPrivileges=true
PrivateTmp=true
ProtectSystem=strict
ReadWritePaths=/srv/http-lfs

# Variables de entorno
Environment="PYTHONUNBUFFERED=1"

[Install]
WantedBy=multi-user.target
EOF
```

### 4. Crear scripts de control manual

#### Script de inicio (**start\_server.sh**):

```
cat > /srv/http-lfs/start_server.sh << 'EOF'
#!/bin/bash
# Script para iniciar el servidor HTTP de prueba

echo "=== Iniciando Servidor HTTP TP LFS ==="
echo ""

# Verificar que Python esté instalado
if ! command -v python3 &> /dev/null; then
    echo "❌ ERROR: Python3 no está instalado"
```

```

    exit 1
fi

# Verificar que el script del servidor existe
if [ ! -f "/srv/http-lfs/mini_server.py" ]; then
    echo "❌ ERROR: No se encuentra mini_server.py"
    exit 1
fi

# Recargar systemd
echo "Recargando systemd..."
systemctl daemon-reload

# Iniciar servicio
echo "Iniciando servicio lfs-http..."
systemctl start lfs-http

# Esperar un momento
sleep 1

# Verificar estado
echo ""
echo "=== Estado del Servicio ==="
systemctl status lfs-http --no-pager

# Mostrar información de conexión
echo ""
echo "=== Información de Conexión ==="
IP_ADDRESS=$(hostname -I 2>/dev/null | awk '{print $1}' || echo "localhost")
echo "📶 Accede desde:"
echo "  Local: http://localhost:8081/"
echo "  Red:   http://${IP_ADDRESS:-<tu-ip>}:8081/"
echo ""
echo "📋 Comandos útiles:"
echo "  Ver logs:    journalctl -u lfs-http -f"
echo "  Ver estado:  systemctl status lfs-http"
echo "  Parar:      systemctl stop lfs-http"
EOF

chmod +x /srv/http-lfs/start_server.sh

```

### Script de parada (**stop\_server.sh**):

```

cat > /srv/http-lfs/stop_server.sh << 'EOF'
#!/bin/bash
# Script para detener el servidor HTTP de prueba

echo "=== Deteniendo Servidor HTTP TP LFS ==="

```

```

echo ""

# Detener servicio
echo "Deteniendo servicio lfs-http..."
systemctl stop lfs-http

# Deshabilitar inicio automático (opcional)
read -p "¿Deshabilitar inicio automático? (s/N): " -n 1 -r
echo ""
if [[ $REPLY =~ ^[Ss]$ ]]; then
    systemctl disable lfs-http
    echo "Servicio deshabilitado para inicio automático."
fi

# Verificar estado
echo ""
echo "=== Estado Final ==="
if systemctl is-active lfs-http && /dev/null; then
    echo "❌ El servicio sigue activo"
    systemctl status lfs-http --no-pager
else
    echo "✅ Servicio detenido correctamente"
fi

# Mostrar puertos en uso
echo ""
echo "=== Puertos en Uso (8081) ==="
if ss -tlnp | grep -q :8081; then
    echo "⚠️ El puerto 8081 todavía está en uso:"
    ss -tlnp | grep :8081
    echo ""
    echo "Para liberar el puerto manualmente:"
    echo " sudo kill \$(lsof -t -i:8081)"
else
    echo "✅ Puerto 8081 liberado"
fi
EOF

```

```

chmod +x /srv/http-lfs/stop_server.sh

```

### Script de verificación (**check\_server.sh**):

```

cat > /srv/http-lfs/check_server.sh << 'EOF'
#!/bin/bash
# Script para verificar el estado del servidor

echo "=== Verificación Servidor HTTP TP LFS ==="
echo "Fecha: $(date)"

```

```
echo ""
```

#### # 1. Verificar PID 1

```
echo "1. 🖱️ PID 1 del sistema:"
```

```
PID1=$(ps -p 1 -o comm= 2>/dev/null || cat /proc/1/comm 2>/dev/null)
```

```
echo " $PID1"
```

```
if [[ "$PID1" == "systemd" ]]; then
```

```
    echo " ✅ Systemd es el PID 1"
```

```
else
```

```
    echo " ⚠️ PID 1 es: $PID1"
```

```
fi
```

#### # 2. Verificar servicio systemd

```
echo ""
```

```
echo "2. ⚙️ Estado del servicio lfs-http:"
```

```
if systemctl is-active lfs-http &> /dev/null; then
```

```
    echo " ✅ ACTIVO"
```

```
    systemctl status lfs-http --no-pager | grep -A5 "Main PID"
```

```
else
```

```
    echo " ❌ INACTIVO"
```

```
fi
```

#### # 3. Verificar puerto

```
echo ""
```

```
echo "3. 🖱️ Puerto 8081:"
```

```
if ss -tlnp | grep -q :8081; then
```

```
    echo " ✅ EN ESCUCHA"
```

```
    ss -tlnp | grep :8081
```

```
else
```

```
    echo " ❌ NO en escucha"
```

```
fi
```

#### # 4. Probar conexión

```
echo ""
```

```
echo "4. 🧪 Prueba de conexión:"
```

```
if command -v curl &> /dev/null; then
```

```
    if timeout 2 curl -s http://localhost:8081/ &> /dev/null; then
```

```
        echo " ✅ Servidor responde"
```

```
        TITLE=$(timeout 2 curl -s http://localhost:8081/ | grep -o '<title>[^<]*</title>' | head -1)
```

```
        echo " Título: $TITLE"
```

```
    else
```

```
        echo " ❌ Servidor NO responde"
```

```
    fi
```

```
else
```

```
    # Usar Python como fallback
```

```
    python3 -c "
```

```
import socket
```

```
try:
```

```

s = socket.socket()
s.settimeout(2)
s.connect(('localhost', 8081))
s.send(b'GET / HTTP/1.0\r\n\r\n')
resp = s.recv(200)
s.close()
if b'200 OK' in resp:
    print(' ✅ Servidor responde (via Python)')
else:
    print(' ⚠️ Respuesta inesperada')
except Exception as e:
    print(f' ❌ Error: {e}')
" 2>/dev/null || echo " ⚠️ No se pudo probar la conexión"
fi

# 5. Mostrar logs recientes
echo ""
echo "5. 📋 Logs recientes:"
journalctl -u lfs-http -n 3 --no-pager 2>/dev/null || echo " No hay logs disponibles"

# 6. Información de acceso
echo ""
echo "6. 🌐 Información de acceso:"
IP=$(hostname -I 2>/dev/null | awk '{print $1}' || echo "localhost")
echo " URL: http://$IP:8081/"
echo " URL local: http://localhost:8081/"
EOF

chmod +x /srv/http-lfs/check_server.sh

```

---

## Uso del Servicio

### Iniciar el servicio (con systemd, como root):

```
# Habilitar para inicio automático (opcional)
systemctl enable lfs-http
```

```
# Iniciar ahora
systemctl start lfs-http
```

```
# O usar el script (recomendado)
/srv/http-lfs/start_server.sh
```

### Verificar estado:

```
# Ver estado del servicio
systemctl status lfs-http
```



```
# Ver logs en tiempo real
journalctl -u lfs-http -f
```

```
# Usar script de verificación (recomendado)
/srv/http-lfs/check_server.sh
```

### **Acceder al servicio:**

```
# Localmente (sin curl)
python3 -c "import urllib.request;
print(urllib.request.urlopen('http://localhost:8081/').read().decode()[:200])"
```

```
# Con curl si está instalado
curl http://localhost:8081/
```

```
# Desde navegador web
# http://<IP-DE-TU-LFS>:8081/
```

### **Detener el servicio:**

```
# Parar servicio
systemctl stop lfs-http
```

```
# Deshabilitar inicio automático
systemctl disable lfs-http
```

```
# O usar el script (recomendado)
/srv/http-lfs/stop_server.sh
```

---

## **Solución de Problemas**

### **Problema: Puerto ya en uso**

```
# Ver qué usa el puerto 8081
ss -tlnp | grep :8081
```

```
# Liberar puerto
sudo kill $(sudo lsof -t -i:8081)
```

```
# O cambiar puerto en mini_server.py
sed -i 's/PORT = 8081/PORT = 8082/' /srv/http-lfs/mini_server.py
```

### **Problema: Python no instalado**

```
# Verificar Python
python3 --version
```

```
# Si no está, instalar desde LFS/BLFS
# wget https://www.python.org/ftp/python/3.11.8/Python-3.11.8.tar.xz
# tar xvf Python-3.11.8.tar.xz
# cd Python-3.11.8
# ./configure --prefix=/usr --enable-shared
# make && make install
```

## Problema: Permisos denegados

```
# Verificar permisos
ls -la /srv/http-lfs/
ls -la /usr/lib/systemd/system/lfs-http.service

# Corregir permisos
chmod 644 /usr/lib/systemd/system/lfs-http.service
chmod +x /srv/http-lfs/*.py /srv/http-lfs/*.sh
```

---



## Para el Informe del TP

### Evidencia a capturar:

#### PID 1 es systemd:

```
ps -p 1 -o pid,comm,args
```

#### Servicio activo:

```
systemctl status lfs-http
```

#### Página web funcionando:

```
curl -s http://localhost:8081/ | grep -o "<title>.*</title>"
```

#### Logs del servicio:

```
journalctl -u lfs-http --since "5 minutes ago"
```

### Comandos para el informe:

```
# Generar reporte automático
{
echo "=== TP LFS - Systemd y Servicio HTTP ==="
echo "Fecha: $(date)"
echo "Hostname: $(hostname)"
echo "PID 1: $(ps -p 1 -o comm=)"
echo "Estado servicio: $(systemctl is-active lfs-http)"
echo "Puerto: $(ss -tlnp | grep :8081 | awk '{print $4}')"
} > /tmp/tp-evidence.txt
```



## Limpieza (Cuando ya no se necesite)

### Eliminar completamente el servicio:

# 1. Detener y deshabilitar

```
systemctl stop lfs-http  
systemctl disable lfs-http
```

# 2. Eliminar archivos de servicio

```
rm -f /usr/lib/systemd/system/lfs-http.service  
systemctl daemon-reload
```

# 3. Eliminar scripts (opcional)

```
rm -rf /srv/http-lfs/
```

# 4. Verificar que no queden procesos

```
kill -f "python.*mini_server"  
ss -tlnp | grep :8081
```

---



## Notas Finales

- **Puerto predeterminado:** 8081
  - **Requisitos:** Python3 instalado
  - **Uso de memoria:** ~2-5 MB
  - **Logs:** Se almacenan en journalctl
  - **Persistencia:** Si se habilita, inicia automáticamente al arrancar
- 







## Enlaces Útiles

- [LFS Official Documentation](#)
  - [Systemd Service Documentation](#)
  - [Python HTTP Server](#)
- 



### Este servicio cumple con los requisitos del TP:

1.  Systemd como PID 1
2.  Servicio activo gestionado por systemd
3.  Accesible vía HTTP para demostración
4.  Fácil de iniciar/detener/verificar