

0. 소개

이 문서는 Ubuntu 22.04 환경에서 FreeSWITCH를 **전체 소스 빌드 방식**으로 설치하는 방법을 설명한다.

Ubuntu 22.04는 FreeSWITCH의 공식 apt 저장소에서 지원되지 않기 때문에, 시스템은 apt 기반 의존성과 소스 컴파일 의존성을 함께 준비해야 한다.

본 가이드는 재현 가능한 설치를 보장하기 위해 배경 설명, 설계 이유, 그리고 단계별 설치 절차를 제공한다.

1. 시스템 준비

FreeSWITCH를 빌드하기 전에 운영체제를 최신 상태로 유지하고 필요한 빌드 도구가 설치되어 있는지 확인해야 한다. 이 도구들은 autotools, CMake 기반 프로젝트, 그리고 코덱 라이브러리를 컴파일하는 데 사용된다.

1.1 시스템 업데이트

```
sudo apt update && sudo apt upgrade -y
```

1.2 핵심 빌드 도구 설치

```
sudo apt install -y git wget curl unzip build-essential g++ automake  
autoconf libtool cmake pkg-config python3-dev yasm nasm
```

이 도구들은 이후 소스에서 빌드되는 의존성들을 컴파일하는 데 필수적이다.

2. apt를 통한 표준 의존성 설치

FreeSWITCH는 다수의 멀티미디어 및 네트워크 라이브러리에 의존한다.

Ubuntu에서 제공하는 일부 라이브러리는 그대로 사용 가능하므로 apt를 통해 설치한다.

```
sudo apt install -y libjpeg-dev libncurses5-dev libssl-dev libpcre3-dev  
libspeex-dev libspeexdsp-dev libsqlite3-dev libcurl4-openssl-dev libldns-  
dev libedit-dev libtiff-dev liblua5.2-dev libopus-dev libsndfile1-dev  
libavformat-dev libavresample-dev libavcodec-dev libavutil-dev libswscale-  
dev libswresample-dev libasound2-dev libunwind-dev libevent-dev
```

이 패키지들은 FreeSWITCH가 기대하는 기본 멀티미디어 스택을 구성한다.

3. 소스에서 빌드해야 하는 의존성

일부 Ubuntu 패키지(특히 sofia-sip, spandsp, 저수준 코덱 라이브러리)는 버전이 오래되었거나 FreeSWITCH와 호환되지 않는다.

따라서 패치되었거나 최신 버전을 직접 컴파일해야 한다.

3.1 libks

```
git clone https://github.com/signalwire/libks.git
cd libks
cmake .
make -j$(nproc)
sudo make install
sudo ldconfig
```

3.2 sofia-sip (패치된 버전 필수)

```
git clone https://github.com/freeswitch/sofia-sip.git
cd sofia-sip
./bootstrap.sh
./configure
make -j$(nproc)
sudo make install
sudo ldconfig
```

3.3 spandsp

```
git clone https://github.com/freeswitch/spandsp.git
cd spandsp
./bootstrap.sh
./configure
make -j$(nproc)
sudo make install
sudo ldconfig
```

3.4 libtiff

```
wget https://download.osgeo.org/libtiff/tiff-4.5.1.tar.gz
tar -xvf tiff-4.5.1.tar.gz
cd tiff-4.5.1
./configure
make -j$(nproc)
sudo make install
sudo ldconfig
```

3.5 libyuv

```
git clone https://chromium.googlesource.com/libyuv/libyuv  
cd libyuv  
cmake .  
make -j$(nproc)  
sudo make install  
sudo ldconfig
```

3.6 libvpx

```
git clone https://chromium.googlesource.com/webm/libvpx  
cd libvpx  
.configure  
make -j$(nproc)  
sudo make install  
sudo ldconfig
```

이 과정은 코덱 처리 및 SIP 스택 동작의 일관성을 보장한다.

4. FreeSWITCH 빌드

4.1 저장소 클론

```
git clone https://github.com/signalwire/freeswitch.git  
cd freeswitch  
git checkout v1.10.9
```

4.2 Bootstrap 및 Configure

```
./bootstrap.sh -j  
./configure --enable-core-pgsql-support
```

4.3 빌드

```
make -j$(nproc)
```

4.4 설치

```
sudo make install  
sudo make cd-sounds-install cd-moh-install
```

이 과정에서 핵심 텔레포니 모듈, 코덱, XML 처리 엔진이 함께 컴파일된다.

5. systemd 서비스 설정

자동 실행을 위해 systemd 서비스를 생성한다.

```
sudo cp debian/freeswitch-systemd.freeswitch.service  
/etc/systemd/system/freeswitch.service  
sudo systemctl daemon-reload  
sudo systemctl enable freeswitch  
sudo systemctl start freeswitch
```

이를 통해 FreeSWITCH는 시스템 부팅 시 자동으로 실행되고, 재시작이 가능해진다.

6. 설치 검증 절차

서비스 상태 확인

```
systemctl status freeswitch
```

CLI 접속

```
fs_cli
```

테스트 오디오 재생

```
fs_cli -x "playback local_stream://moh"
```

로그 확인

```
tail -f /usr/local/freeswitch/log/freeswitch.log
```

이 단계들은 설치가 정상적으로 완료되었고 미디어 서브시스템이 올바르게 동작하는지 확인하는 데 사용된다.

7. 디렉터리 구조

FreeSWITCH는 기본적으로 `/usr/local/freeswitch/` 경로에 설치된다.

- `bin/` – 실행 파일 (`freeswitch`, `fs_cli`)
- `conf/` – XML 설정 파일 (SIP 프로파일, 다이얼플랜)
- `log/` – 실행 로그
- `sounds/` – 시스템 안내 음성 및 MOH
- `/var/lib/freeswitch/` – 데이터베이스 및 런타임 상태

이 구조를 이해하면 디버깅 및 커스터마이징이 수월해진다.

8. 사용자 정의 오디오 처리

FreeSWITCH 호환 형식으로 오디오를 변환한다.

```
ffmpeg -i input.mp3 -ac 1 -ar 8000 output.wav
```

다음 경로에 저장한다.

```
/usr/local/freeswitch/sounds/custom/
```

다이얼플랜에서의 사용 예:

```
<action application="playback" data="custom/output.wav"/>
```

9. 문제 해결

빌드 실패 시

누락된 라이브러리를 확인하기 위해 `configure`를 다시 실행한다.

```
./configure
```

SIP 등록 실패

패치된 `sofia-sip`이 설치되어 있는지 확인한다.

오디오가 나오지 않을 때

코덱 협상 및 NAT 설정을 점검한다.

MOH 누락

```
sudo make cd-moh-install
```

위 단계들은 설치 및 운영 중 자주 발생하는 문제를 해결하는 데 도움이 된다.