**USB를 활용한 AVML 작업하기**

- 목차 -

1. 기존의 설치 방식과 차이점?
2. 환경 구성
3. 테스트
4. 기존의 설치 방식과 차이점?

-기존의 방식-

# Install MUSL

sudo apt-get install musl-dev musl-tools musl

# Install Rust via rustup

curl https://sh.rustup.rs -sSf | sh -s -- -y

# Add the MUSL target for Rust

rustup target add x86\_64-unknown-linux-musl

# Build

cargo build --release --target x86\_64-unknown-linux-musl

# Build without upload functionality

cargo build --release --target x86\_64-unknown-linux-musl --no-default-features

(위의 글은 github에 설명되어 있는 설치 방식이다 확인해보면 2번째 줄에 curl 명령어를 사용하는 것을 볼 수 있는데 이는 **실제 호스트에 다운 및 설치**하며 이 설치한 것을 사용하기 위해서는 환경변수를 지정해야 합니다. 이에 따라 우리는 다음과 같이 다운 및 설치를 진행해야 합니다.)

-usb에 avml 사용 환경을 생성하기 위한 수정된 방식-

1단계: avml을 git clone하기

2단계: rustup을 rust사이트에서 대상 호스트 환경에 맞게 다운 및 설치하기

3단계: musl을 다운 받기(만약 2단계에서 gnu가 아닌 musl인 것을 확인했을 시 이 단계는 건너 뛰어도 됩니다)

4단계: 2단계에서 다운 받은 rust 설치파일 내 cargo를 사용하여 avml을 build 한다.

5단계: build avml을 사용해본다.

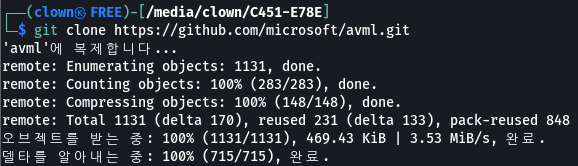
- 1단계 - avml clone하기 -

1. 장착한 usb의 경로 확인하기



(위의 예시에서는 /dev/sdb1 이 USB 드라이브의 파티션을 나타내며, /media/clown/C451-E78E가 해당 파티션을 마운트한 디렉토리입니다.)

1. avml 복사하기

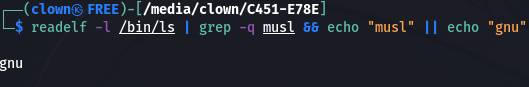


이렇게 복사를 하고 나면 다음 단계로 넘어갑니다.

- 2단계 - 필수 설치파일[rustup] 준비하기 -

1. 터미널에서 gnu인지 musl인지 확인합니다.

(커맨드: readelf -l /bin/ls | grep -q musl && echo "musl" || echo "gnu"\n)



(gnu인 것을 확인했습니다)

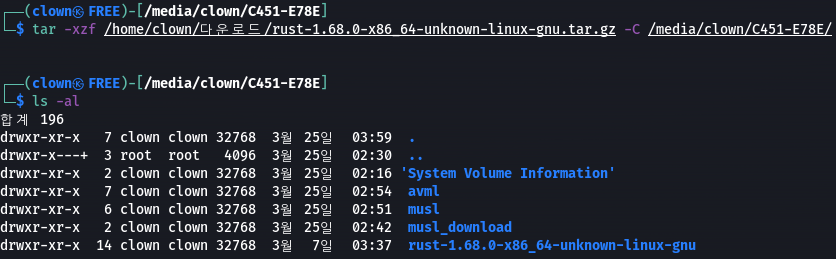
2. 아래 링크로 가서 다운받습니다.

<https://forge.rust-lang.org/infra/other-installation-methods.html>

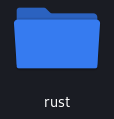
아래로 쭉 내리다면 “본인의 환경에 부합하는 파일”을 다운 받습니다 (현재 테스트하는 환경에서는 위의 명령에서 “gnu”라고 나타나서 아래와 같이 선택해서 다운 받습니다)



1. 다운 받은 파일을 압축 해제 합니다.

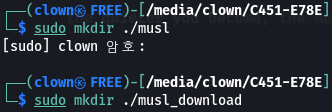


1. 작업의 편의성을 위하여 만들어진 디렉토리인 rust-1.68.0-x86\_64-unknown-linux-gnu를 간단하게 rust로 명명합니다.



- 3단계 - 필수 설치 파일 [musl] 준비하기 -

1. USB 드라이브 내에 musl 라이브러리를 저장할 디렉토리(musl)를 생성하고 다운로드한 musl 압축 파일을 저장할 디렉토리(musl\_download)를 생성합니다.

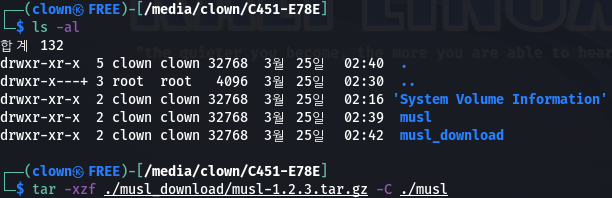


1. Musl 압축 파일을 다운로드합니다. Musl 공식 홈페이지에서 최신 버전의 링크를 확인할 수 있습니다. 이 예에서는 1.2.3버전을 사용하겠습니다.

(커맨드: wget -P ./musl\_download https://musl.libc.org/releases/musl-1.2.3.tar.gz)



1. 다운로드한 압축 파일을 앞서 생성한 ‘musl’ 디렉토리로 이동하고 압축을 해제합니다.



1. 압축 해제된 musl 디렉토리로 이동합니다.



1. Musl 라이브러리를 빌드하고 설치합니다. 이때 ‘—prefix’ 옵션을 사용하여 라이브러리가 설치될 USB 드라이브 경로를 지정합니다.

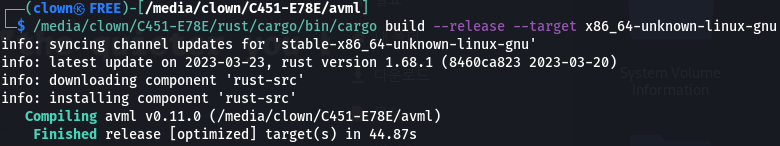
(커맨드: ./configure --prefix=/media/clown/C451-E78E/musl)







- 4단계 – (2단계에서 다운 받은 rust 필수 설치 파일안에 cargo 디렉토리 있음) cargo로 avml build 진행하기



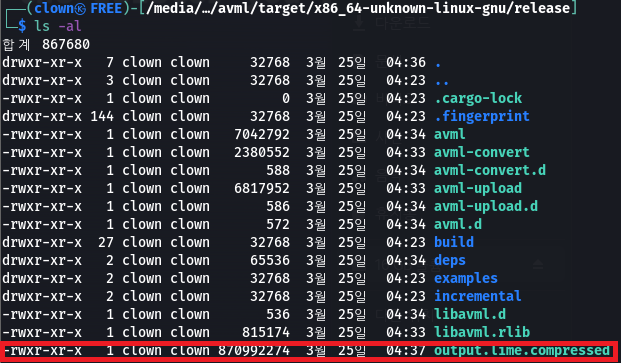
1단계에서 git clone 한 avml 디렉토리로 이동해서 cargo build –release –target x86\_64-unknown-linux-gnu를 입력한다. {위 사진은 해당 전반적으로 진행되는 대량의 로그를 짧게 함축했으면 성공했으면 하단에 보이는 메시지와 같이 Finished release [optimized] target(s) in ~ 로 나타난다.

(만약에 빌드하는 도중에 에러가 발생하면 다시 위의 사진에서 보이는 명령을 똑같이 입력해서 계속 진행하도록 한다)

- 5단계 – build한 avml 정상적으로 동작하는지 테스트 해보기, sudo ./avml –compress output.lime.compressed



기다리면 아래와 같이 파일 등장합니다.



결과물이 생성된 것을 볼 수 있습니다.

-결론-

Avml을 사용하기 위하여

리눅스 환경에서는 rust 설치 파일로



Musl과



gnu

위 두가지를 미리 준비해 둬야하는 것은 아닌가? 라는 생각이 든다.

대부분의 경우 기본적으로 설치된 glibc를 사용하면 문제가 없지만, 일부 경우 musl libc를 사용하는 경우가 있기때문이다.