6.18. GCC-13.2.0 - 2차 설치

GCC 패키지에는 C 컴파일러와 C++ 컴파일러가 포함된 GNU 컴파일러 컬렉션이 포함되어 있습니다.

- **빌드 시간:** 4.4 SBU
- **디스크 공간:** 4.8 GB

6.18.1. GCC 설치

GCC의 첫 번째 빌드에서와 마찬가지로 GMP, MPFR 및 MPC 패키지가 필요합니다. 타르볼의 압축을 풀고 디렉터리 이름을 변경합니다.

```
tar -xf ../mpfr-4.2.1.tar.xz
mv -v mpfr-4.2.1 mpfr
tar -xf ../gmp-6.3.0.tar.xz
mv -v gmp-6.3.0 gmp
tar -xf ../mpc-1.3.1.tar.gz
mv -v mpc-1.3.1 mpc
```

x86_64에서 빌드하는 경우 64비트 라이브러리의 기본 디렉터리 이름을 “lib”로 변경합니다.

```
case $(uname -m) in
  x86_64)
    sed -e '/m64=/s/lib64/lib/' \
        -i.orig gcc/config/i386/t-linux64
  ;;
esac
```

POSIX 스레드를 지원하는 라이브러리를 빌드할 수 있도록 libgcc 및 libstdc++ 헤더의 빌드 규칙을 재정의합니다.

```
sed '/thread_header =/s/@.*@/gthr-posix.h/' \
    -i libgcc/Makefile.in libstdc++-v3/include/Makefile.in
```

분리된 빌드 디렉터리를 생성합니다.

```
mkdir -v build
cd       build
```

GCC 빌드를 시작하기 전에 최적화 플래그를 재정의하는 환경 변수[1)]를 설정 해제해야 합니다.

이제 컴파일을 위해 GCC를 준비합니다.

```
../configure                                        \
    --build=$(../config.guess)                      \
    --host=$LFS_TGT                                 \
    --target=$LFS_TGT                               \
    LDFLAGS_FOR_TARGET=-L$PWD/$LFS_TGT/libgcc       \
    --prefix=/usr                                   \
    --with-build-sysroot=$LFS                       \
    --enable-default-pie                            \
    --enable-default-ssp                            \
    --disable-nls                                   \
    --disable-multilib                              \
    --disable-libatomic                             \
    --disable-libgomp                               \
    --disable-libquadmath                           \
    --disable-libsanitizer                          \
    --disable-libssp                                \
    --disable-libvtv                                \
    --enable-languages=c,c++
```

설정 옵션 설명

- *–with-build-sysroot=$LFS*
  일반적으로 *–host*를 사용하면 크로스 컴파일러가 GCC 빌드에 사용되며, 해당 컴파일러는 $LFS에서 헤더와 라이브러리를 찾아야 한다는 것을 알고 있습니다. 그러나 GCC 빌드 시스템은 이 위치를 인식하지 못하는 다른 도구도 사용합니다. 이 스위치는 이러한 도구가 호스트가 아닌 $LFS에서 필요한 파일을 찾도록 하기 위해 필요합니다.
- *–target=$LFS_TGT*
  GCC를 크로스 컴파일하고 있으므로 이전에 컴파일한 GCC 바이너리로 대상 라이브러리(`libgcc` 및 `libstdc++`)를 빌드할 수 없습니다(해당 바이너리는 호스트에서 실행되지 않음). GCC 빌드 시스템은 기본적으로 호스트의 C 및 C++ 컴파일러를 사용하려고 시도합니다. 다른 버전의 GCC로 GCC 대상 라이브러리를 빌드하는 것은 지원되지 않으므로 호스트의 컴파일러를 사용하면 빌드가 실패할 수 있습니다. 이 옵션은 GCC 1차에 빌드한 라이브러리를 사용 하도록 합니다.
- *LDFLAGS_FOR_TARGET=...*
  libstdc++가 GCC 패스 1에서 빌드된 정적 버전 대신 이 패스에서 빌드 중인 libgcc 공유 라이브러리를 사용하도록 합니다. 이는 C++ 예외 처리를 지원하는 데 필요합니다.
- *–disable-libsanitizer*
  GCC 새니타이저 런타임 라이브러리를 비활성화합니다. 임시 설치에는 필요하지 않습니다. 이 스위치는 libcrypt가 설치되지 안은 대상 시스템에 대한 GCC를 빌드하는 데 필요합니다. GCC - 1차에서는 –disable-libstdcxx로 이것을 암시했지만 이제는 명시적으로 설정에 전달해야 합니다.

패키지를 컴파일합니다.

```
make
```

패키지를 설치합니다.

```
make DESTDIR=$LFS install
```

마지막으로 유용한 심볼릭 링크를 생성합니다. 많은 프로그램과 스크립트는 **gcc** 대신 **cc**를 실행하는데, 이는 프로그램을 범용적으로 유지하고, GNU C 컴파일러가 항상 설치되어 있지 않은 모든 종류의 UNIX 시스템에서 사용할 수 있도록 합니다. 시스템 관리자가 사용할 C 컴파일러를 **cc**를 통해서 원하는대로 결정할 수 있습니다.

```
ln -sv gcc $LFS/usr/bin/cc
```

이 패키지에 대한 자세한 내용은 섹션 8.28.2, "GCC의 내용"에 나와 있습니다.

[1]
CFLAGS, CXXFLAGS, LDFLAGS 등

From:
https://gamu.kr/dokuwiki/ - **전자 수첩**

Permanent link:
https://gamu.kr/dokuwiki/linuxfromscratch/12.1/057-gcc-13.2.0_-_pass_2?rev=1717832697

Last update: **2024/06/08 07:44**