

---

# Gentoo CFLAGS

Which one to choose?

Onkobu Tanaake

2025-01-03T10:00:00

The following article is only about `march` and three scenarios: a standalone computer, a build server and a client consuming the build server packages. For the first case the choice is simple: `march=native`. The only disadvantage is the loss of the system in case hardware has to be switched that is incompatible with the architecture setting. If a replacement CPU does not support opcodes the binaries contain the system will not boot. For the curious there is a plethora of options. Examine the output of:

```
# march=native detailliert erläutern, was auch für den
# Vergleich zweier CPUs ganz praktisch ist:
#
gcc -v -E -x c /dev/null -o /dev/null -march=native 2>&1 \
| grep /ccl \
| grep mtune
```

A build server (or build node being used by a `distcc` setup) creates packages for other computers and potentially different CPUs. I own a set of AMD CPUs that differ a bit. Next to this there'd be Intel CPUs. Both can be satisfied by using the generic `x86-64-v2` to `v4`. Take the lowest common version returned by this script:

```
#!/bin/sh -eu

flags=$(cat /proc/cpuinfo | grep flags | head -n 1 | cut -d: -f2)

supports_v2='awk "/cx16/&&/lahf/&&/popcnt/&&/sse4_1/&&/sse
supports_v3='awk "/avx/&&/avx2/&&/bmi1/&&/bmi2/&&/f16c/&&/fma/&&/a
supports_v4='awk "/avx512f/&&/avx512bw/&&/avx512cd/&&/avx512dq/&&/

echo "$flags" | eval $supports_v2 || exit 2 && echo "CPU supports
echo "$flags" | eval $supports_v3 || exit 3 && echo "CPU supports
echo "$flags" | eval $supports_v4 || exit 4 && echo "CPU supports
```

Since I run the build environment itself as a `CHROOT` set apart from the true server this can be even different from the host's `CFLAGS`. Whereas the server itself uses `march=native` my `CHROOT` environment specifies `march=x86-64-v2`.

I even set apart each client from the build server's packages. If a package benefits from more specific `CFLAGS` the client skips the server's package and rolls its own.

Therefore each client uses a more specific architecture setting. And since my main work horses that are energy saving laptops with mainboards and CPUs that cannot be replaced with a free choice my first choice are Safe CFLAGS[1].

Strangely I could run all machines with `athlon64` which broke recently. Harfbuzz with its central role for Firefox, LightDM, Qt and Libreoffice in combination with GCC14 and an exotic bug[2] broke an entire client setup. This particular client was a Piledriver AMD CPU that triggers the bug in gcc.

Finally have a look at `hartwork/resolve-march-native`[3]. Like the first script above this unrolls `-march=native` filtering only the most important CFLAGS. It is shipped with the package `app-misc/resolve-march-native`[4].