



La belleza de compilar

Gentoo y amigos

Sets de instrucciones de CPU

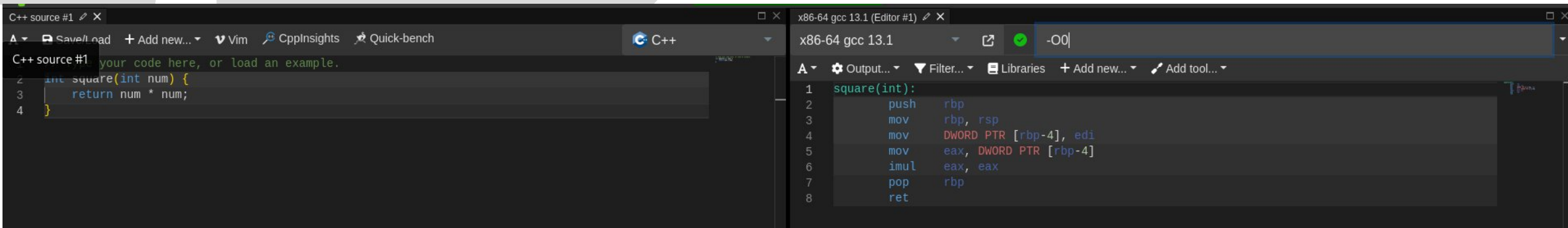
- X86_64: sse4.1, avx, fma, etc
- ARM: v6,v7, v8...
 - neon
 - criptográficas para aes, sha...
- PowerPC: vsx

En el compilador

- march
 - native
 - znver1,2,3
 - haswell
 - skylake
- mtune
- Instrucciones específicas
 - mfma
 - m3dnow
 - mavx
 - msse
- Niveles de optimización:
 - O0 para debugear
 - O1 optimizaciones básicas
 - O2 optimizaciones recomendadas
 - O3 optimizaciones agresivas
 - Os optimización para reducir el tamaño

Ejemplos niveles de optimización:

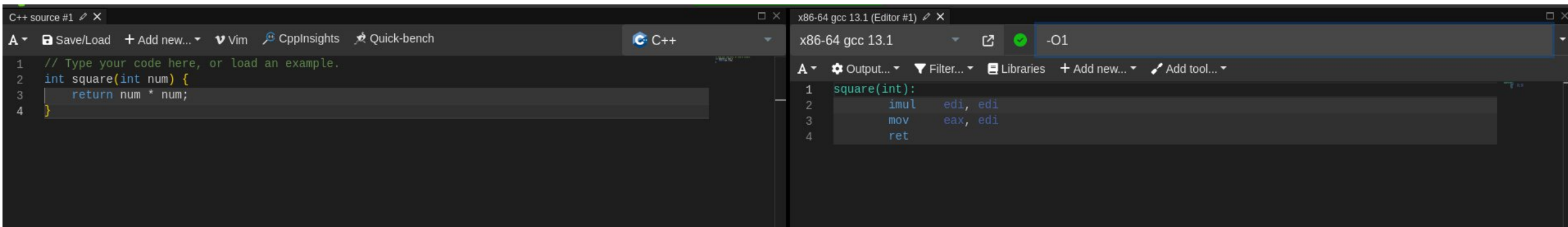
-O0 vs -O1



This screenshot shows the VS Code editor with a C++ source file and its assembly output for the -O0 optimization level. The source code is a simple square function. The assembly output shows a function prologue with stack frame setup, followed by a multiplication operation using the `imul` instruction.

```
C++ source #1 X
A Save/Load + Add new... Vim CppInsights Quick-bench C++
C++ source #1: your code here, or load an example.
1 // Type your code here, or load an example.
2 int square(int num) {
3     return num * num;
4 }

x86-64 gcc 13.1 (Editor #1) X
x86-64 gcc 13.1 -O0
A Output... Filter... Libraries + Add new... Add tool...
1 square(int):
2     push    rbp
3     mov     rbp, rsp
4     mov     DWORD PTR [rbp-4], edi
5     mov     eax, DWORD PTR [rbp-4]
6     imul    eax, eax
7     pop     rbp
8     ret
```



This screenshot shows the VS Code editor with the same C++ source code and its assembly output for the -O1 optimization level. The assembly output is significantly more compact than the -O0 version, as it has optimized away the stack frame setup and the memory access, using a single `imul edi, edi` instruction to perform the multiplication.

```
C++ source #1 X
A Save/Load + Add new... Vim CppInsights Quick-bench C++
1 // Type your code here, or load an example.
2 int square(int num) {
3     return num * num;
4 }

x86-64 gcc 13.1 (Editor #1) X
x86-64 gcc 13.1 -O1
A Output... Filter... Libraries + Add new... Add tool...
1 square(int):
2     imul    edi, edi
3     mov     eax, edi
4     ret
```

Ejemplos uso de juegos de instrucciones:

-march=x86-64 vs x86-64-v4

```
// Then perform the matrix-matrix multiplication
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++) {
        double sum = 0.0;
        for (int k = 0; k < n; k++) {
            sum += B[i][k]*A[k][j];
        }
        C[i][j] = sum;
    }
}

// Compute now the Frobenius norm
double Fsum = 0.0;
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++) {
        Fsum += C[i][j]*C[i][j];
    }
}
```

```
205     mov     rax, QWORD PTR [rsp+32]
206     mov     rdi, QWORD PTR [rbx+r11*8]
207     xor     esi, esi
208     mov     r10, QWORD PTR [rax+r11*8]
209     .L26:
210     xor     eax, eax
211     pxor    xmm1, xmm1
212     .L22:
213     mov     rdx, QWORD PTR [r15+rax*8]
214     movsd   xmm0, QWORD PTR [rdx+rsi]
215     mulsd   xmm0, QWORD PTR [rdi+rax*8]
216     mov     rdx, rax
217     add     rax, 1
218     addsd   xmm1, xmm0
219     cmp     r14, rdx
```

```
// Then perform the matrix-matrix multiplication
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++) {
        double sum = 0.0;
        for (int k = 0; k < n; k++) {
            sum += B[i][k]*A[k][j];
        }
        C[i][j] = sum;
    }
}

// Compute now the Frobenius norm
double Fsum = 0.0;
for (int i = 0; i < n; i++){
    for (int j = 0; j < n; j++) {
        Fsum += C[i][j]*C[i][j];
    }
}
```

```
205     mov     rdi, QWORD PTR [rbx+r11*8]
206     xor     esi, esi
207     mov     r10, QWORD PTR [rax+r11*8]
208     .L26:
209     xor     eax, eax
210     vxorpd  xmm0, xmm0, xmm0
211     .L22:
212     mov     rdx, QWORD PTR [r15+rax*8]
213     vmovsd  xmm2, QWORD PTR [rdi+rax*8]
214     vfmadd231sd  xmm0, xmm2, QWORD PTR [rdx+rsi]
215     mov     rdx, rax
216     add     rax, 1
217     cmp     r14, rdx
218     jne     .L22
219     vmovsd  QWORD PTR [r10+rsi], xmm0
```

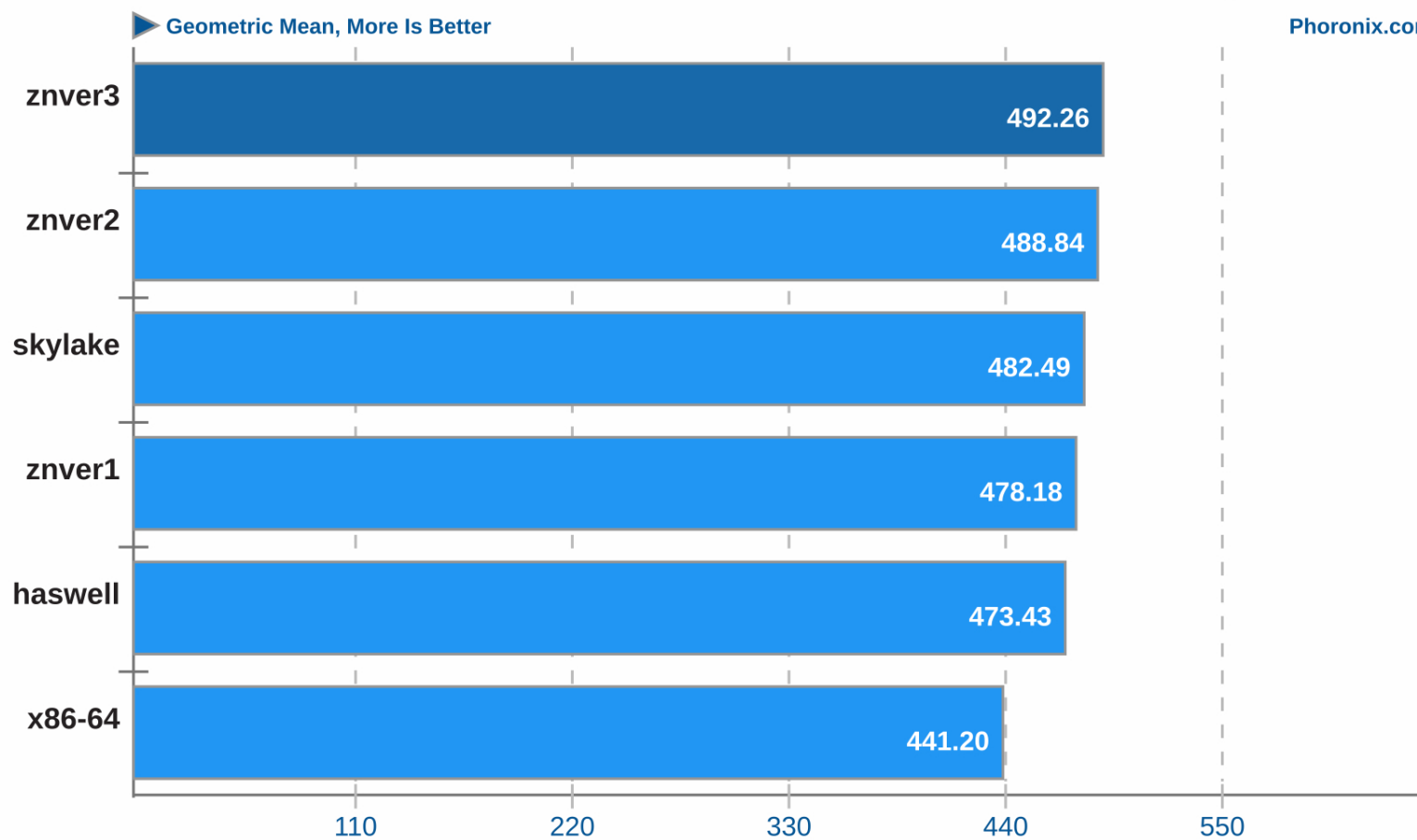
Benchmarks

Geometric Mean Of All Test Results

Result Composite



Phoronix.com

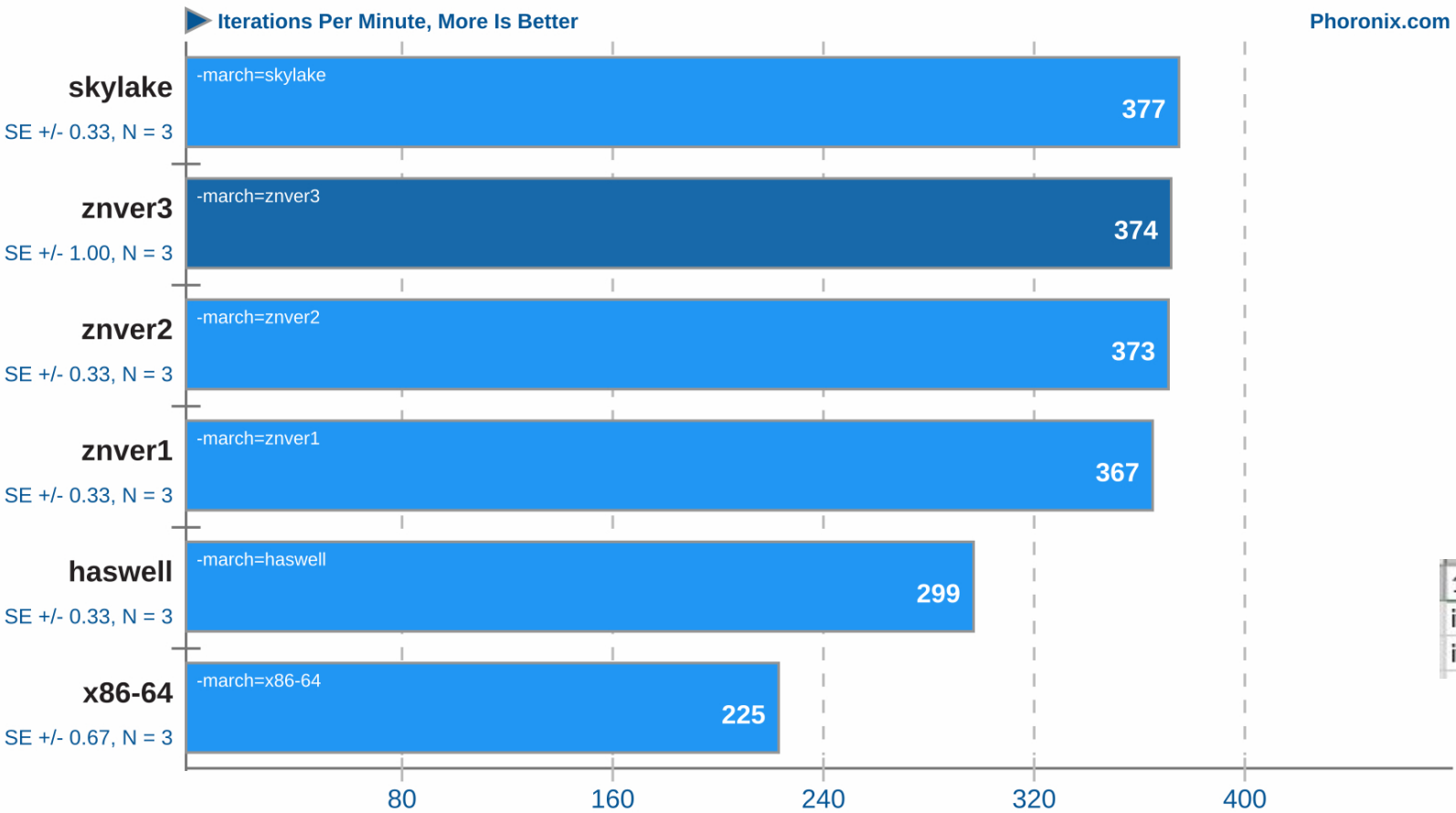


GraphicsMagick 1.3.33

Operation: Sharpen



Phoronix.com



100 iteraciones	gentoo	debian	arch
iteraciones/s	6,983	4,137	4,138
iteraciones/cpu	0,975	0,402	0,41

1. (CC) gcc options: -fopenmp -O3 -pthread -ljbig -lwebp -lwebpmux -ltiff -lfreetype -ljpeg -lXext -lSM -lICE -lX11 -llzma -lbz2 -lxml2 -lz -lm -lpthread

Fedora 39 Planning For RPM 4.19 - Adds x86-64 v2/v3/v4 Feature Levels

Written by [Michael Larabel](#) in [Fedora](#) on 31 March 2023 at 06:27 AM EDT. [17 Comments](#)



With the Fedora 39 release later this year the developers are planning on moving to RPM 4.19 as the newest version of their packaging format.

Most exciting with RPM 4.19 is that it adds the x86-64 micro-architecture feature level of x86-64 v2/v3/v4 as new architectures for handling of RPM packages. Hopefully having this built-in support for those micro-architecture feature levels will lead to more widespread use on Fedora and other RPM-based Linux distributions.

Arch Linux Developers Discuss Idea Of Providing An x86-64-v3 Port

Written by [Michael Larabel](#) in [Arch Linux](#) on 16 March 2021 at 08:30 AM EDT. [83 Comments](#)



While recently Arch Linux developers and stakeholders were discussing [the possibility of raising the x86-64 base requirements](#) for this Linux distribution to the "x86-64-v2" micro architecture feature level that roughly correlates to Intel Nehalem and newer, now the discussion has shifted to keeping the same x86-64 base level while potentially offering "x86-64-v3" port for those with newer Intel/AMD CPUs.

Rather than raising the Arch Linux x86 64-bit requirements to the "x86-64-v2" level that would yield issues for those trying to run this distribution on the oldest of original x86-64 AMD/Intel processors, the proposal has morphed into providing an x86-64-v3 port that would be maintained concurrently to base x86-64. With this, users running Arch on vintage PCs wouldn't lose out on updates while those on more recent hardware would be able to tap into more optimized packages by default.

Aggressive compiler flags

Clear Linux OS uses aggressive [compiler flags](#) to optimize software built for it. The flags it implements are:

[mtune and march](#)

Options used to tune generated code with optimized instructions for performance and compatibility.

Clear Linux OS defines its minimum hardware requirements to be x86-64 (Intel Core 2 2010) or later. This enables compiler optimizations that are available on the Haswell generation processors or newer.

Clear Linux OS sets **march=westmere** and **mtune=haswell**.

[Note](#)

Clear Linux OS doesn't require Advanced Encryption Standard (AES) for the x86-64 microarchitecture code name Nehalem (released in 2008). Refer to the [AES](#) page for more details.

[O3](#)

The largest preset of compiler options optimizations for performance.

View the "Optimize Options" section of the GCC man page for additional details.

[LTO](#)



Nada nuevo

- Gentoo 2000
- CRUX 2002
- Funtoo 2008
- KISS 2019
- Venom 2019

Opciones

- Tu distribución favorita con su gestor de compilaciones
 - Debian y apt-src, apt-build
 - Arch y ABS
 - FreeBSD y sus ports
 - NetBSD y pkg_src
 - Seguramente otros que no conozco
- Metadistribuciones o source-based



Algún motivo más?

- Personalización
- Aprender
- Es divertido :)



Personalización

- Añadir/quita funcionalidades del código
- Añadir funcionalidades extraoficiales/parches
- Arreglar bugs

Añadir/quita funcionalidades del código

LOCAL USE FLAGS

static suexec suexec-caps suexec-syslog

GLOBAL USE FLAGS

debug doc gdbm ldap selinux split-usr ssl systemd threads

APACHE2_MODULES (USE EXPAND)

access_compat actions alias asis auth_basic auth_digest auth_form authn_anon
authn_core authn_dbd authn_dbm authn_file authn_socache authz_core authz_dbd
authz_dbm authz_groupfile authz_host authz_owner authz_user autoindex brotli
cache cache_disk cache_socache cern_meta cgi cgid charset_lite dav dav_fs
dav_lock dbd deflate dir dumpio env expires ext_filter file_cache filter
headers http2 ident imagemap include info lbmethod_bybusyness
lbmethod_byrequests lbmethod_bytraffic lbmethod_heartbeat log_config
log_forensic logio lua macro md mime mime_magic negotiation proxy proxy_ajp
proxy_balancer proxy_connect proxy_fcgi proxy_ftp proxy_hcheck proxy_html
proxy_http proxy_http2 proxy_scgi proxy_uwsgi proxy_wstunnel ratelimit
remoteip reqtimeout rewrite session session_cookie session_crypto
session_dbd setenvif slotmem_shm socache_memcache socache_shmcb speling
status substitute unique_id unixd userdir usertrack version vhost_alias
watchdog xml2enc

APACHE2_MPMS (USE EXPAND)

event prefork worker

LUA_SINGLE_TARGET (USE EXPAND)

lua5-1 lua5-3 lua5-4

```
.config - Linux/x86_64 6.1.10-gentoo Kernel Configuration

General setup
[ ] Compile also drivers which will not load
[ ] Compile the kernel with warnings as errors
(-x86_64) Local version - append to kernel release
[ ] Automatically append version information to the version string
() Build ID Salt
Kernel compression mode (ZSTD) --->
() Default init path
((none)) Default hostname
-* System V IPC
[*] POSIX Message Queues
[*] General notification queue
[*] Enable process_vm_readv/writev syscalls
[ ] uselib syscall (for libc5 and earlier)
[*] Auditing support
IRQ subsystem --->
Timers subsystem --->
BPF subsystem --->
Preemption Model (Preemptible Kernel (Low-Latency Desktop)) --->
[*] Preemption behaviour defined on boot
[ ] Core Scheduling for SMT
CPU/Task time and stats accounting --->
[*] CPU isolation
RCU Subsystem --->
<*> Kernel .config support
[*] Enable access to .config through /proc/config.gz
< > Enable kernel headers through /sys/kernel/kheaders.tar.xz
(18) Kernel log buffer size (16 => 64KB, 17 => 128KB)
(12) CPU kernel log buffer size contribution (13 => 8 KB, 17 => 128KB)

F1Help F2SymInfo F3Help 2 F4ShowAll F5Back F6Save F7Load F8SymSearch F9Exit
```

USEs

```
# These settings were set by the catalyst build script that auto-
# matically
# built this stage.
# Please consult /usr/share/portage/config/make.conf.example for
# a more
# detailed example.
COMMON_FLAGS="-O3 -march=native -pipe"
CFLAGS="${COMMON_FLAGS}"
CXXFLAGS="${COMMON_FLAGS}"
FCFLAGS="${COMMON_FLAGS}"
FFLAGS="${COMMON_FLAGS}"

RUSTFLAGS="-C target-cpu=native -C opt-level=3"

MAKEOPTS="-j16"
# NOTE: This stage was built with the bindist Use flag enabled
PORTDIR="/var/db/repos/gentoo"
DISTDIR="/var/cache/distfiles"
PKGDIR="/var/cache/binpkgs"

# This sets the language of build output to English.
# Please keep this setting intact when reporting bugs.
LC_MESSAGES=C
USE="eloguind -systemd -consolekit -webkit bluetooth pgo wayland
lto graphite pulseaudio screencast pipewire vulkan vaapi vdpau"
L10N="es es-ES"
LINGUAS="es es-ES"
CPU_FLAGS_X86="aes avx avx2 f16c fma3 mmx mmxext pclmul popcnt r
drand sha sse sse2 sse3 sse4_1 sse4_2 sse4a ssse3"

VIDEO_CARDS="amdgpu radeonsi"

GRUB_PLATFORMS="efi-64"
```

```
# required by app-emulation/virt-manager-3.2.0::gentoo[gtk]
# required by virt-manager (argument)
net-misc/spice-gtk usbredir
# required by app-emulation/libvirt-7.3.0::gentoo[virt-network]
# required by app-emulation/libvirt-glib-4.0.0::gentoo
# required by app-emulation/virt-manager-3.2.0::gentoo
# required by virt-manager (argument)
net-dns/dnsmasq script
app-emulation/libvirt libssh lxc
```

Exprimiendo que todo es libre

```
325 local tools="cargo"
326 use clippy && tools+=",clippy"
327 use miri && tools+=",miri"
328 use profiler && tools+=",rust-demangler"
329 use rustfmt && tools+=",rustfmt"
330 use rust-analyzer && tools+=",rust-analyzer","analysis"
331 use rust-src && tools+=",src"
332
333 local rust_stage0_root
334 if use system-bootstrap; then
335     local printsysroot
336     printsysroot="$(rustc --print sysroot || die "Can't determine rust's sysroot")"
337     rust_stage0_root="${printsysroot}"
338 else
339     rust_stage0_root="${WORKDIR}/rust-stage0"
340 fi
341 # in case of prefix it will be already prefixed, as --print sysroot returns full path
342 [[ -d ${rust_stage0_root} ]] || die "${rust_stage0_root} is not a directory"
343
344 rust_target="$(rust_abi)"
345
346 local cm_btype="$(usex debug DEBUG RELEASE)"
347 cat <<- _EOF_ > "${S}/config.toml"
348     changelog-seen = 2
349     [llvm]
350     download-ci-llvm = false
351     optimize = ${toml_usex !debug}
352     release-debuginfo = ${toml_usex debug}
353     assertions = ${toml_usex debug}
354     ninja = true
355     "rust-analyzer" = "rust-analyzer"
```


- 27.2.2 systemd
- 28 System logger
 - 28.1 OpenRC
 - 28.2 systemd
- 29 Optional: Cron daemon
 - 29.1 OpenRC
 - 29.1.1 crone
 - 29.1.2 Alternative: dcron
 - 29.1.3 Alternative: fcron
 - 29.1.4 Alternative: bcron
 - 29.2 systemd
- 30 Optional: File indexing
- 31 Optional: Remote shell access
 - 31.1 OpenRC
 - 31.2 systemd
- 32 Time synchronization
 - 32.1 OpenRC
 - 32.2 systemd
- 33 Filesystem tools
- 34 Networking tools
 - 34.1 Installing a DHCP client
 - 34.2 Optional: Installing a PPPoE client
 - 34.3 Optional: Install wireless networking tools
- 35 Selecting a boot loader
- 36 Default: GRUB
 - 36.1 Emerge
 - 36.2 Install
 - 36.3 Configure
- 37 Alternative 1: LILO
 - 37.1 Emerge
 - 37.2 Configure
 - 37.3 Install
- 38 Alternative 2: efibootmgr
- 39 Alternative 3: Syslinux
- 40 Rebooting the system
- 41 User administration
 - 41.1 Adding a user for daily use
- 42 Disk cleanup
 - 42.1 Removing tarballs
- 43 Where to go from here
 - 43.1 Additional documentation
 - 43.2 Gentoo online
 - 43.2.1 Forums and IRC
 - 43.2.2 Mailing lists
 - 43.2.3 Bugs
 - 43.2.4 Development guide
 - 43.3 Closing thoughts

Aprendizaje

Aprender no es una opción

Instalar Gentoo te enseña a no reinstalar

1. Preparar medio de instalación (sea cual sea)
2. Formatear
3. Descomprimir el stage3
4. Configurar el sistema base desde un chroot
 1. Configurar herramientas extra
5. Instalar el kernel
6. Instalar el bootloader

Todo es modular (siempre lo fue)

- Librería base de C
 - glibc
 - musl
 - uclib
- Init
 - SystemD
 - OpenRC
 - s6
 - runit
- Servicios de red
 - NetworkManager
 - Netifric
- Desktop Environment
- Shell
- Demonio Cron
- Logs del sistema
- SSL
- Compilador

Fuentes

- Benchmarks:
 - <https://www.phoronix.com/review/amd-znver3-gcc11>
- Noticias:
 - <https://www.phoronix.com/news/Fedora-39-RPM-4.19>
 - <https://www.phoronix.com/news/Arch-Linux-x86-64-v3-Port-RFC>
- Documentación:
 - <https://clearlinux.github.io/clear-linux-documentation/guides/clear/performance.html>
 - <https://gcc.gnu.org/onlinedocs/gcc/x86-Options.html>
 - <https://wiki.gentoo.org/wiki/Handbook:AMD64>