

Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024



First-Party kernel.org Build Environments

Guillaume Tucker
gtucker@gtucker.io

Reviewed-by: Nathan Chancellor
nathan@kernel.org



LINUX
PLUMBERS
CONFERENCE Vienna, Austria / Sept. 18-20, 2024

Take me to your Compiler

Simple case with standard distro toolchain*:

```
make defconfig  
make
```

*Terms and conditions apply for:

- cross-compiling
- Rust
- Clang
- eBPF kselftest
- documentation
- new features in linux-next
- ...

“ There are times where you may need access to different versions of LLVM for building the kernel instead of the one available through your distribution, such as reproducing an issue only visible with specific versions or gaining access to a feature only available in a newer version. ”

- **Slim LLVM toolchains for building the Linux kernel**



State of the Art

“ These compilers are only functional for kernel builds, they cannot be used to build userspace code. ”

kernel.org toolchain tarballs

GCC

+ cross-compilers

LLVM

+ Rust

Comprehensive

all architectures

recent versions

tailored for kernel builds

Index of /pub/tools/crosstool/files/bin/x86_64/14.2.0/

../			
sha256sums.asc	05-Aug-2024 13:05	6427	
x86_64-gcc-14.2.0-nolibc-aarch64-linux.tar.gz	05-Aug-2024 12:37	70M	
x86_64-gcc-14.2.0-nolibc-aarch64-linux.tar.sign	05-Aug-2024 12:37	566	
x86_64-gcc-14.2.0-nolibc-aarch64-linux.tar.xz	05-Aug-2024 12:37	42M	
x86_64-gcc-14.2.0-nolibc-alpha-linux.tar.gz	05-Aug-2024 12:38	46M	
x86_64-gcc-14.2.0-nolibc-alpha-linux.tar.sign	05-Aug-2024 12:38	566	
x86_64-gcc-14.2.0-nolibc-alpha-linux.tar.xz	05-Aug-2024 12:38	29M	
x86_64-gcc-14.2.0-nolibc-arc-linux.tar.gz	05-Aug-2024 12:39	48M	
x86_64-gcc-14.2.0-nolibc-arc-linux.tar.sign	05-Aug-2024 12:39	566	
x86_64-gcc-14.2.0-nolibc-arc-linux.tar.xz	05-Aug-2024 12:39	30M	
x86_64-gcc-14.2.0-nolibc-arm-linux-gnueabi.tar.gz	05-Aug-2024 12:40	52M	
x86_64-gcc-14.2.0-nolibc-arm-linux-gnueabi.tar...>	05-Aug-2024 12:40	566	
x86_64-gcc-14.2.0-nolibc-arm-linux-gnueabi.tar.xz	05-Aug-2024 12:40	33M	
x86_64-gcc-14.2.0-nolibc-csky-linux.tar.gz	05-Aug-2024 12:40	49M	

<https://mirrors.edge.kernel.org/pub/tools/>



LINUX
PLUMBERS
CONFERENCE

Vienna, Austria / Sept. 18-20, 2024

State of the Art

Linaro tuxmake containers

```
$ tuxmake --runtime=docker --target=x86_64 --toolchain=korg-clang-18
```

Available as a service via [TuxSuite](#)

Default containers provided by Linaro

Alternative containers with kernel.org toolchains:

<https://www.linaro.org/blog/tuxmake-building-linux-with-kernel-org-toolchains/>

“ TuxMake provides Docker container images to build Linux kernels across various architectures and toolchain combinations. ”

- stylesen 'Tuxmake: Building Linux with kernel.org LLVM toolchains'



LINUX
PLUMBERS
CONFERENCE Vienna, Austria / Sept. 18-20, 2024



tuxmake

What's the issue again?

Tarballs are nice but not super convenient
download, extract, install, adjust \$PATH, remove, upgrade by hand

Not everything is included in tarballs

```
bash bc bison flex git libelf-dev libssl-dev make
```

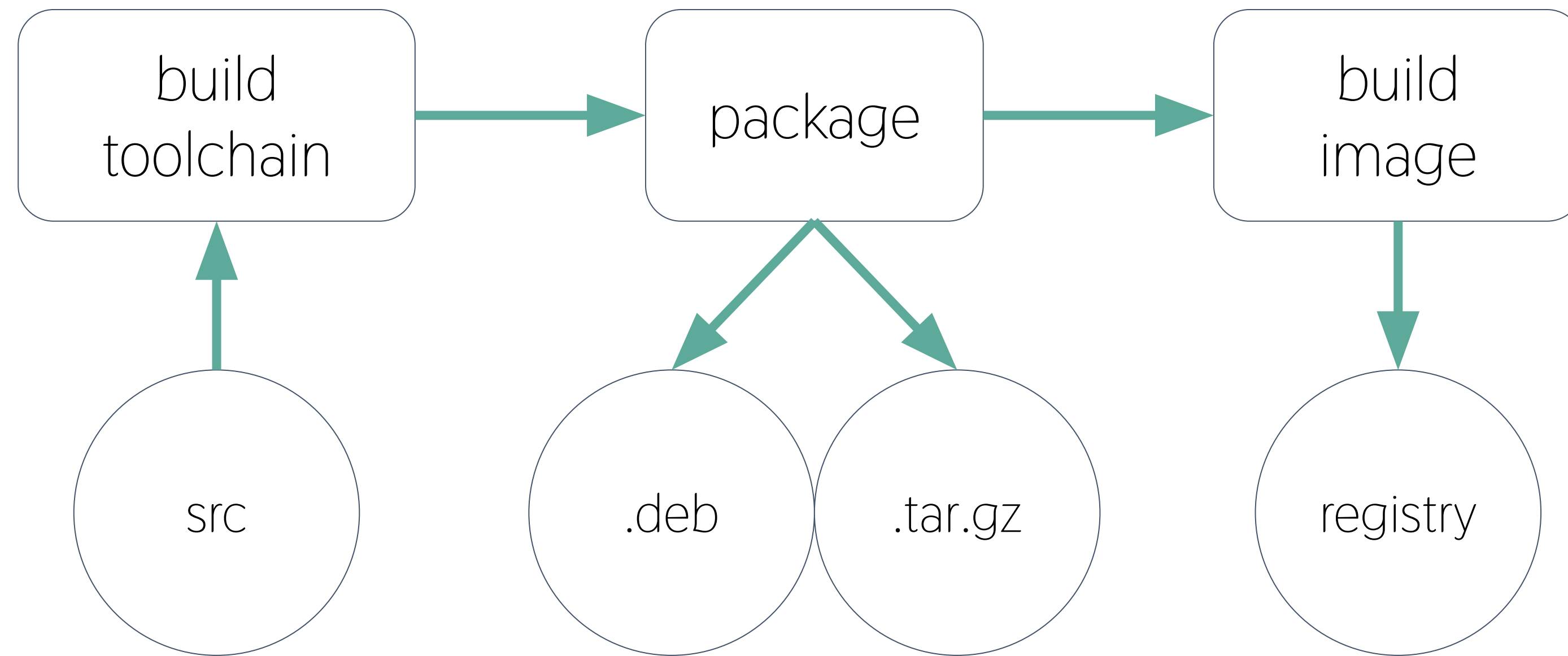
Tarballs are uploaded by designated individuals (i.e. Arnd)
discoverability, traceability, ability for developers to contribute

Everybody* uses containers, mostly Docker, especially for automation

*except those who don't, obviously



Toolchain Pipeline in the Sky



```
docker build tools/containers/gcc/14.2/x86 -t registry.kernel.org/gcc:14.2-x86
```

 *nothing groundbreaking so far in 2024 (!)*





Down to Earth



I really don't want some kind of top-level CI for the base kernel project.



- Linus Torvalds, [kci-gitlab: Introducing GitLab-CI Pipeline for Kernel Testing](#)

 Vendor lock-in alert: GitLab, GitHub, GKE, Azure, AWS 

OCI standard makes images vendor neutral

Harbor is a popular open-source registry

Note: images may also be saved as files for plain HTTP downloads (meh)

```
docker build tools/containers/gcc/14.2/x86 -t gcc:14.2-x86
docker save gcc:14.2-x86 | gzip > gcc-14.2-x86.gz
# upload to web server, then to "pull":
curl https://images.kernel.org/gcc-14.2-x86.gz | gunzip | podman load
```

↔ Git hooks on git.kernel.org to trigger builds?

↔ Other vendor-neutral solutions needed?



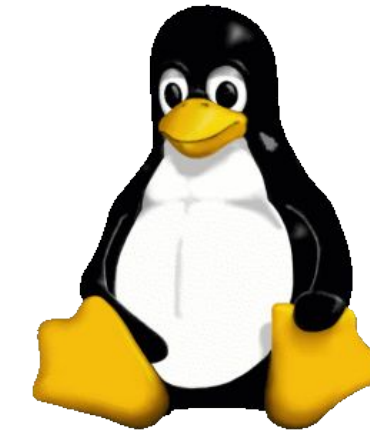
Down to Earth

Business as Usual

Keep the current tarballs
no change to existing workflows

Add packages with meta-data
.deb, .rpm, Yocto

Add Containerfiles in upstream kernel tree (e.g. tools/containers)
entirely optional, backwards-compatible



Kbuild

Building upon first-party container images:

```
make CONTAINER=gcc:14.2-x86
```

Does this seem useful?

Raise your hand if you think so

Can it be implemented?

Kbuild is complex but working PoC (next slide)

 *Thanks Nathan for suggesting the Kbuild integration!*



```
diff --git a/Makefile b/Makefile
index c6f549f6a4ae..e2a55162238d 100644
--- a/Makefile
+++ b/Makefile
@@ -1,3 +1,13 @@
+ifneq ($(CONTAINER),)
+PHONY := __all
+__all:
+%:
+    @echo RUNNING IN CONTAINER
+    @docker run -v $(PWD):/src -w /src \
+    $(CONTAINER) $(MAKE) \
+    $(subst CONTAINER=$(CONTAINER),,$(MAKEFLAGS)) \
+    $(GNUMAKEFLAGS) $(MAKECMDGOALS)
+else
+    # SPDX-License-Identifier: GPL-2.0
+    VERSION = 6
+    PATCHLEVEL = 7
@@ -2051,3 +2061,4 @@ FORCE:
+    # Declare the contents of the PHONY variable as phony. We keep
+    # that
+    # information in a variable so we can use it in if_changed and
+    # friends.
+    .PHONY: $(PHONY)
+endif # DOCKER
```

<https://gitlab.com/gtucker/linux/-/commits/linux-6.7-make-container>



**LINUX
PLUMBERS
CONFERENCE** Vienna, Austria / Sept. 18-20, 2024

```
$ make CONTAINER=gtucker/gcc-12:x86 defconfig
RUNNING IN CONTAINER
HOSTCC  scripts/basic/fixdep
HOSTCC  scripts/kconfig/conf.o
HOSTCC  scripts/kconfig/confdata.o
HOSTCC  scripts/kconfig/expr.o
LEX      scripts/kconfig/lexer.lex.c
YACC     scripts/kconfig/parser.tab.[ch]
HOSTCC  scripts/kconfig/lexer.lex.o
HOSTCC  scripts/kconfig/menu.o
HOSTCC  scripts/kconfig/parser.tab.o
HOSTCC  scripts/kconfig/preprocess.o
HOSTCC  scripts/kconfig/symbol.o
HOSTCC  scripts/kconfig/util.o
HOSTLD  scripts/kconfig/conf
*** Default configuration is based on 'x86_64_defconfig'
#
# configuration written to .config
#
```

Makemake

Dwarf planet in the Solar System

Makemake is a dwarf planet and the largest of what is known as the classical population of Kuiper belt objects, with a diameter approximately that of Saturn's moon Iapetus, or 60% that of Pluto. It has one known satellite. Its extremely low average temperature, about 40 K, means its surface is covered with methane, ethane, and possibly nitrogen ices.

[Wikipedia](#)



Vielen Dank



**LINUX
PLUMBERS
CONFERENCE** Vienna, Austria / Sept. 18-20, 2024