

# debuginfod

## and its interaction with the Yocto Project

Dylan Garza

June 3, 2022

# debuginfod

- What is debuginfod? debuginfod is a daemon turns a machine that holds debug artifacts into file server for easier debugging.
- Why use it? Debug info is too big to be stored locally on the target devices. Different versions of binaries exist, storing and searching for debug artifacts manually is too slow.

# Installing debuginfod

There are two ways to install on a Ubuntu machine:

- Upgrade to Ubuntu 22.04
- append the impish package to `/etc/apt/sources.list`

# Setting up and using debuginfod

Starting the debuginfod server:

- To start the file server, simply run `debuginfod -F`. Different arguments will search for different debug artifacts (`-F -R -U`)
- Providing a path to a directory will point debuginfod where to scan for the debug artifacts.
- To set a rescan time, provide `-t [time in seconds]`

# Setting up and using debuginfod (cont.)

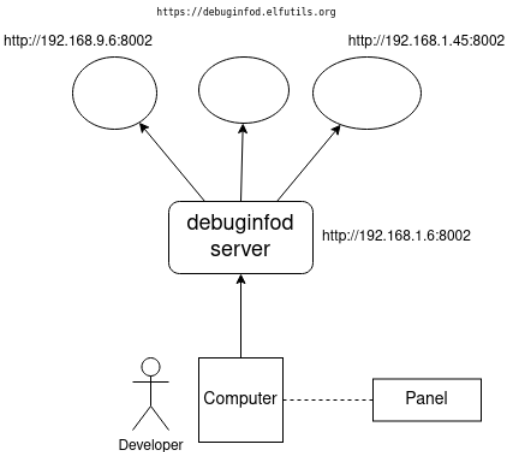
On the host machine:

- set environment variable `DEBUGINFOD_URLS` to the ip address of the file server, prefixed by `http://` and suffixed by the port number defaulted to `:8002`, which can be changed by `-p [port num]`

```
export DEBUGINFOD_URLS="http://ip_addr:8002"
```

# Federating debuginfod servers

debuginfod servers can act as a host machine and query other servers  
federated debuginfod servers



## Federating debuginfo servers (cont.)

```
2 3 4 5 6 |> [dmu-2]
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from nvim...

This GDB supports auto-downloading debuginfo from the following URLs:
https://debuginfo.archlinux.org
Enable debuginfo for this session? (y or [n]) y
Debuginfo has been enabled.
To make this setting permanent, add 'set debuginfo enabled on' to .gdbinit.
Downloading 14.73 MB separate debug info for /home/dylandy/nvim
Reading symbols from /home/dylandy/.cache/debuginfo_client/2e9f4968123c62457f5ad3f6c68c518c328c49/debuginfo...
(gdb) list
Downloading 0.06 MB source file /usr/src/debug/nvim-0.7.0/src/nvim/main.c
200 set_lang_var(); // set setlang and vctype
201
202 init_signs();
203 ui_comp_syn_init();
204 }
205
206 #ifdef MAKE_LIB
207 int nvim_main(int argc, char **argv); // silence -Wmissing-prototypes
208 int nvim_main(int argc, char **argv)
209 #elif defined WIN32
210
x Error of failed request: BadName (named color or font does not exist)
Major opcode of failed request: 140 (RANDR)
Minor opcode of failed request: 16 (RRCreateMode)
Serial number of failed request: 35
Current serial number in output stream: 35
dylandy@super-desktop
os Artix Linux
host x86_64
kernel 5.17.6-artix1-1
uptime 4h 56m
pkgs 703
memory 2127M / 32077M

dylandy@super-desktop ~
└─ export DEBUIFINFOD_URLS="https://192.168.1.2:8002"
└─ dylandy@super-desktop ~
└─ rm -rf .cache/debuginfo_client
└─ dylandy@super-desktop ~

[Pv4:192.168.1.6] dylandy | RAM: 1.7 GB | CPU: 0% | Speed: 2.8 GHz | M2022-06-03 09:53:30
rdb: Can't open display ''
dylandy@super-computer
os Artix Linux
host 17295N-G.AASBUI 0.1
kernel 5.17.6-artix1-1
uptime 15d 22h 15m
pkgs 895
memory 562M / 15689M

Couldn't get a file descriptor referring to the console.
-bash: [: : integer expression expected
[dylandy@super-computer ~] export DEBUIFINFOD_URLS="https://debuginfo.elfutils.org"
[dylandy@super-computer ~] ls ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default qlen 1000
    link/ether c4:23:60:ed:a1:11 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/24 brd 192.168.1.255 scope global dynamic noprefixroute wlan0
        valid_lft 86214sec preferred_lft 86214sec
    inet6 fe80::8b6d:fbfe:2023:e2f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
[dylandy@super-computer ~] ls []

[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): rescantime 300
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): fdcache fids 144
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): fdcache mbs 16908
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): fdcache prefch 66
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): fdcache tpdtr /tmp
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): fdcache tpdtr minx 25
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): groom time 86400
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): prefch fids 0
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): prefch mbs 0
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): forwarded tli limit 8
[Fri 03 Jun 2022 01:52:08 PM GMT] (16168/16168): upstream debuginfo servers: https://debuginfo.d.archlinux.org
[Fri 03 Jun 2022 01:52:09 PM GMT] (16168/16170): grooming database
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): database record counts:
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): file d/e 922
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): file s 35476
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): archive d/e 0
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): archive sref 0
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): archive sref 0
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): buildids 615
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): filenames 310324
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): files scanned (#) 300010
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): files scanned (mb) 82493
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): index db size (mb) 80
[Fri 03 Jun 2022 01:52:10 PM GMT] (16168/16170): groomed database in 0.75601s

[0] > gdbx
"super-desktop" 09:53 03-Jun-22
[0] > bashx
"dylandy@super-compute" 09:55 03-Jun-22
```

# Demo



# debuginfod with the Yocto Project

For versions after 3.4(honister) only 2 changes are needed for debuginfod to work on a build. Append the following the `local.conf`:

1

```
PACKAGE_CONFIG_pn-elfutils-native="debuginfod libdebuginfod
```

2

```
DISTROFEATURES += "debuginfod"
```

Yocto Project has built in script for debuginfod to eliminate compatability issues.

# debuginfod-find

Command apart of elfutils, it performs debuginfod server querying without gdb. Can query for debuginfo, executables, and sources.

```
debuginfod-find [debuginfo/executable/source] [path/to/file or Build-ID] [path/to/source]
```

If the path to the source is not known, but debug info is available, use the `list` command under `gdb` to provide source path.

# Demo