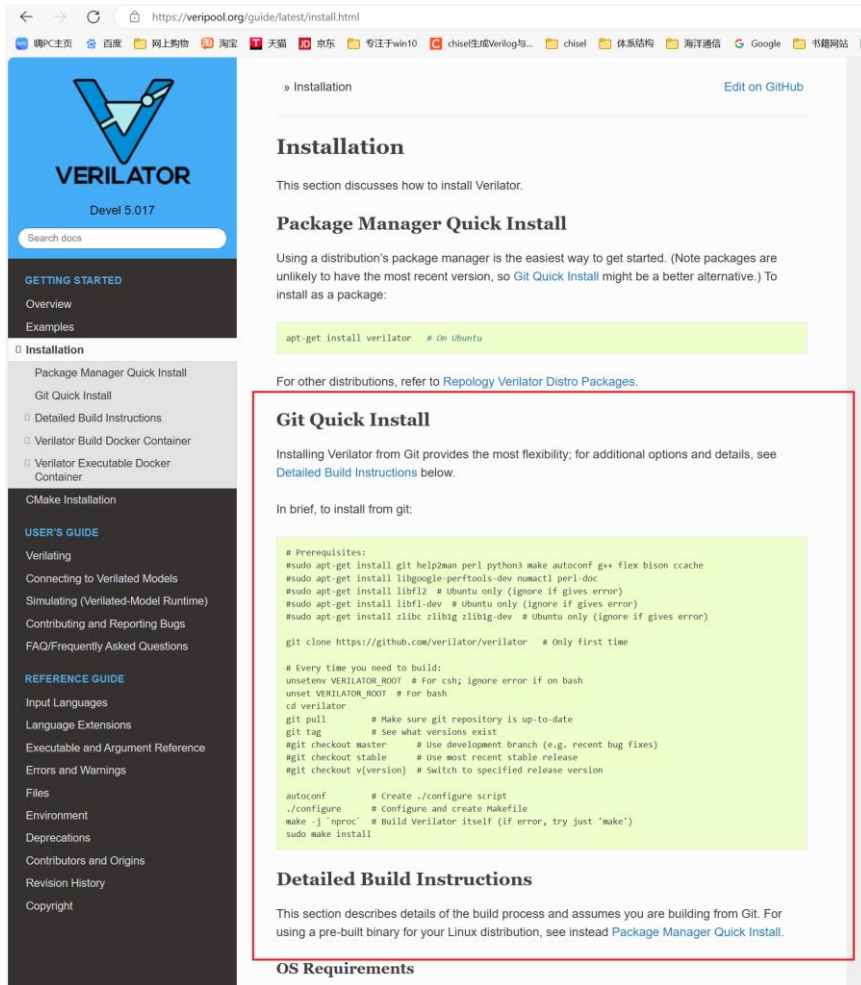


# 多版本 Verilator 的安装与管理

verilator 在官网下载仓库，自己进行构建。<https://github.com/verilator/verilator.git>



verilator 官方的帮助界面：<https://veripool.org/guide/latest/install.html>

采用默认的编译命令，verilator 将编译安装在 `usr/local/bin` 目录下。如果不需要保留旧版本，可直接覆盖安装即可。如果之前安装了 verilatorv4.210 版本，并且需要保留旧版本，这里将提供一个版本管理和切换的思路。

首先是按照这里的指导一步步的将 verilator 安装好。这里还是有一个坑的，就是一个包的版本对不上，我的处理方法是直接去下载那个包并使用 `dpkg -i` 强制安装即可。（zlibc）

然后就是 make 前（autoconf 后）的配置，按照 Eventual Installation Options 里的指导，按照第一个思路，直接输入以下命令，设置当前目录为编译好的二进制文件夹所在目录。

```
export VERILATOR_ROOT=`pwd` # if your shell is bash
setenv VERILATOR_ROOT `pwd` # if your shell is csh
```

## Eventual Installation Options

Before configuring the build, you must decide how you're going to eventually install Verilator onto your system. Verilator will be compiling the current value of the environment variables

`VERILATOR_ROOT`, `SYSTEMC_INCLUDE`, and `SYSTEMC_LIBDIR` as defaults into the executable, so they must be correct before configuring.

These are the installation options:

### 1. Run-in-Place from VERILATOR\_ROOT

Our personal favorite is to always run Verilator in-place from its Git directory (don't run `make install`). This allows the easiest experimentation and upgrading, and allows many versions of Verilator to co-exist on a system.

```
export VERILATOR_ROOT=`pwd` # if your shell is bash
setenv VERILATOR_ROOT `pwd` # if your shell is csh
./configure
# Running will use files from $VERILATOR_ROOT, so no install needed
```

Note after installing (see [Installation](#)), a calling program or shell must set the environment variable `VERILATOR_ROOT` to point to this Git directory, then execute `$VERILATOR_ROOT/bin/verilator`, which will find the path to all needed files.

### 2. Install into a Specific Prefix

You may be an OS package maintainer building a Verilator package, or you may eventually be installing onto a project/company-wide "CAD" tools disk that may support multiple versions of every tool. Tell configure the eventual destination directory name. We recommend that the destination location include the Verilator version name:

```
unset VERILATOR_ROOT # if your shell is bash
unsetenv VERILATOR_ROOT # if your shell is csh
# For the tarball, use the version number instead of git describe
./configure --prefix /CAD_DISK/verilator/`git describe` | sed "s/verilator_/_/"
```

Note after installing (see [Installation](#)), you need to add the path to the `bin` directory to your `PATH`. Or, if you use `modulecmd`, you'll want a module file like the following:

```
set install_root /CAD_DISK/verilator/{version-number-used-above}
unsetenv VERILATOR_ROOT
prepend-path PATH $install_root/bin
prepend-path MANPATH $install_root/man
prepend-path PKG_CONFIG_PATH $install_root/share/pkgconfig
```

### 3. Install System Globally

The final option is to eventually install Verilator globally, using configure's default system paths:

然后就是使用 `ubuntu` 自带的 `update-alternatives` 进行软件版管理和切换，原理是进行双重软链接，将要进行版本管理的软件注册到 `update-alternatives` 数据库中，并在 `/usr/bin` 目录下设置软链接。

建议使用 `makefile` 进行命令的执行，因为命令行较长，手打难以清晰辨认注册 `verilator` 旧版：

|   |
|---|
| <code>update-alternatives --install /usr/bin/verilator verilator /usr/local/bin/verilator 100 \</code>                          |
| <code>--slave /usr/bin/verilator_bin verilator_bin /usr/local/bin/verilator_bin \</code>  |
| <code>--slave /usr/bin/verilator_bin_dbg verilator_bin_dbg /usr/local/bin/verilator_bin_dbg \</code>                            |
| <code>--slave /usr/bin/verilator_coverage verilator_coverage /usr/local/bin/verilator_coverage \</code>                         |
| <code>--slave /usr/bin/verilator_coverage_bin_dbg verilator_coverage_bin_dbg /usr/local/bin/verilator_coverage_bin_dbg \</code> |
| <code>--slave /usr/bin/verilator_gantt verilator_gantt /usr/local/bin/verilator_gantt \</code>                                  |
| <code>--slave /usr/bin/verilator_profconfc verilator_profconfc /usr/local/bin/verilator_profconfc</code>                        |

注册 `verilator 5.016`：

|   |
|---|
| update-alternatives --install /usr/bin/verilator verilator /home/kyxp-zp/programs/verilator5/verilator/bin/verilator 200                          |
| --slave /usr/bin/verilator_bin verilator_bin /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_bin  |
| --slave /usr/bin/verilator_bin_dbg verilator_bin_dbg /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_bin_dbg                            |
| --slave /usr/bin/verilator_coverage verilator_coverage /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_coverage                         |
| --slave /usr/bin/verilator_coverage_bin_dbg verilator_coverage_bin_dbg /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_coverage_bin_dbg |
| --slave /usr/bin/verilator_gantt verilator_gantt /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_gantt                                  |
| --slave /usr/bin/verilator_proffunc verilator_proffunc /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_proffunc                         |
| --slave /usr/bin/verilator_ccache_report verilator_ccache_report /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_ccache_report          |
| --slave /usr/bin/verilator_difftree verilator_difftree /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_difftree                         |
| --slave /usr/bin/verilator_includer verilator_includer /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_includer                         |

然后就可以看现在系统中有什么 verilator 的版本了

```

update-alternatives --display verilator
kyxp-zp@kyxpzp-VirtualBox:~$ update-alternatives --display verilator
verilator - auto mode
# link best version is /home/kyxp-zp/programs/verilator5/verilator/bin/verilator
# link currently points to /home/kyxp-zp/programs/verilator5/verilator/bin/verilator
# link verilator is /usr/bin/verilator
# slave verilator_bin is /usr/bin/verilator_bin
# slave verilator_bin_dbg is /usr/bin/verilator_bin_dbg
# slave verilator_ccache_report is /usr/bin/verilator_ccache_report
# slave verilator_coverage is /usr/bin/verilator_coverage
# slave verilator_coverage_bin_dbg is /usr/bin/verilator_coverage_bin_dbg
# slave verilator_difftree is /usr/bin/verilator_difftree
# slave verilator_gantt is /usr/bin/verilator_gantt
# slave verilator_includer is /usr/bin/verilator_includer
# slave verilator_proffunc is /usr/bin/verilator_proffunc
/home/kyxp-zp/programs/verilator5/verilator/bin/verilator - priority 200
# slave verilator_bin: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_bin
# slave verilator_bin_dbg: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_bin_dbg
# slave verilator_ccache_report: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_ccache_report
# slave verilator_coverage: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_coverage
# slave verilator_coverage_bin_dbg: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_coverage_bin_dbg
# slave verilator_difftree: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_difftree
# slave verilator_gantt: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_gantt
# slave verilator_includer: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_includer
# slave verilator_proffunc: /home/kyxp-zp/programs/verilator5/verilator/bin/verilator_proffunc
/usr/local/bin/verilator - priority 100
# slave verilator_bin: /usr/local/bin/verilator_bin
# slave verilator_bin_dbg: /usr/local/bin/verilator_bin_dbg
# slave verilator_coverage: /usr/local/bin/verilator_coverage
# slave verilator_coverage_bin_dbg: /usr/local/bin/verilator_coverage_bin_dbg
# slave verilator_gantt: /usr/local/bin/verilator_gantt
# slave verilator_proffunc: /usr/local/bin/verilator_proffunc

```