

Installing LLVM

■ Installing

■ Installing



Binary

Source

■ Installing



Binary

Faster installation

Installing

Binary

Faster installation

Smaller

■ Installing



Source

Development

■ Installing



Source

Development

Distribute!

■ Installing



Source

Development

Distribute! (Apache License v2.0)

■ Installing (Binary)

Installing (Binary)

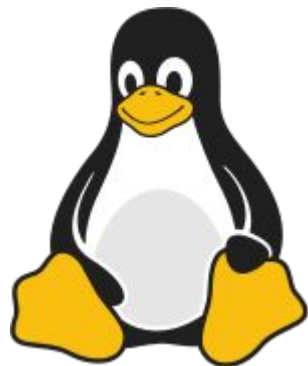
```
$ brew install llvm@10
```



OS X

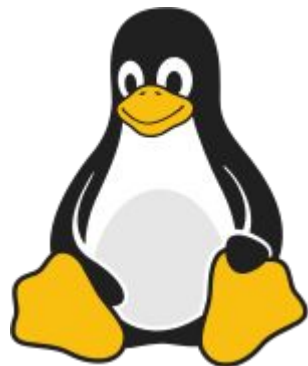
Installing (Binary)

```
$ sudo apt-get install -y llvm-10
```



Installing (Binary)

```
$ sudo apt-get install -y llvm-10 \
  llvm-10-dev llvm-10-tools clang-10
```



■ Installing (Source)

■ Installing (Source) - Preferable

Installing (Source) - Preferable

1. Download
2. Compile

Installing (Source) - Preferable

1. Download

```
$ git clone https://github.com/llvm/llvm-project.git
```


Installing (Source) - Preferable

1. Download

2. Compile

```
$ cd llvm-project/
```

```
$ mkdir build
```

```
$ cd build/
```

Installing (Source) - Preferable

1. Download
2. Compile

```
$ cmake ../llvm
```

Installing (Source) - Preferable

1. Download
2. Compile

```
$ cmake ../llvm
```

```
$ cd ~/llvm-project/llvm  
$ ls CMakeLists.txt  
CMakeLists.txt
```

Installing (Source) - Preferable

1. Download
2. Compile

```
$ cmake ../llvm
```

```
$ cd ~/llvm-project/llvm  
$ ls CMakeLists.txt  
CMakeLists.txt
```

Installing (Source) - Preferable

1. Download
2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \
```

Installing (Source) - Preferable

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=path/to/build \  

```

Installing (Source) - Preferable

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/llvm-project/build \  

```

Installing (Source) - Preferable

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/.llvm-project/build \  
-DBUILD_SHARED_LIBS=on \  

```


Installing (Source) - Preferable

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/.llvm-project/build \  
-DBUILD_SHARED_LIBS=on \  
-DLLVM_ENABLE_PROJECTS=clang
```

Installing (Source) - Preferable

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/.llvm-project/build \  
-DBUILD_SHARED_LIBS=on \  
-DLLVM_ENABLE_PROJECTS=clang
```

Installing (Source) - The Gold Linker

1. Download

2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/.llvm-project/build \  
-DBUILD_SHARED_LIBS=on \  
-DLLVM_ENABLE_PROJECTS=clang \  
-DLLVM_USE_LINKER=gold
```

Installing (Source) - The Gold Linker

1. Download
2. Compile

```
$ cmake ../llvm -G "Unix Makefiles" \  
-DCMAKE_INSTALL_PREFIX=~/.llvm-project/build \  
-DBUILD_SHARED_LIBS=on \  
-DLLVM_ENABLE_PROJECTS=clang \  
-DLLVM_USE_LINKER=gold
```

<https://llvm.org/docs/GoldPlugin.html>

Installing (Source) - Preferable

1. Download
2. Compile

```
$ make -j4
```

Installing (Source) - Preferable

1. Download
2. Compile

```
$ make -j4
```

What was built?

What was built?

```
$ cd ~/llvm-project/build/bin
```

```
$ ls
```


What was built?



FileCheck	lli	llvm-install-name-tool	llvm-rc
apinotes-test	lli-child-target	llvm-isel-fuzzer	llvm-readelf
arcmt-test	llvm-PerfectShuffle	llvm-itanium-demangle-fuzzer	llvm-readobj
bugpoint	llvm-addr2line	llvm-jitlink	llvm-reduce
c-arcmt-test	llvm-ar	llvm-jitlink-executor	llvm-rtdyld
c-index-test	llvm-as	llvm-lib	llvm-size
clang	llvm-bcanalyzer	llvm-libtool-darwin	llvm-special-case-list-fuzzer
clang++	llvm-bitcode-stp	llvm-link	llvm-split
clang-12	llvm-c-test	llvm-lipo	llvm-stress
clang-check	llvm-cat	llvm-lit	llvm-strings
clang-cl	llvm-cfi-verify	llvm-locstats	llvm-strip
clang-cpp	llvm-config	llvm-lto	llvm-symbolizer
clang-diff	llvm-cov	llvm-lto2	llvm-tblgen
clang-extdef-map	llvm-cvtres	llvm-mc	llvm-undname
clang-format	llvm-cxxdump	llvm-mca	llvm-xray
clang-import-test	llvm-cxxfilt	llvm-ms-demangle-fuzzer	llvm-yaml-numeric-parser-fuzzer
clang-offld-bundler	llvm-cxxmap	llvm-ml	not
clang-offld-wrapper	llvm-diff	llvm-modextract	obj2yaml
clang-refactor	llvm-dis	llvm-mt	opt
clang-rename	llvm-dlltool	llvm-nm	sancov
clang-scan-deps	llvm-dwarfdump	llvm-objcopy	sanstats
clang-tblgen	llvm-dwp	llvm-objdump	scan-build
count	llvm-elfabi	llvm-opt-fuzzer	scan-view
diagtool	llvm-exegesis	llvm-opt-report	split-file
dsymutil	llvm-extract	llvm-pdbutil	verify-uselistorder
hmaptool	llvm-gsymutil	llvm-profdata	yaml-bench
l1c	llvm-ifs	llvm-ranlib	yaml2obj

Contact & References



Fernando Magno Quintão Pereira

fernando@dcc.ufmg.br

<http://lac.dcc.ufmg.br>

References

<https://llvm.org/docs/GettingStarted.html>