Build and install software from source code, familiarity with make and cmake

utilities expected.

Procedure:

Step 1: Set Up the Repository

sudo apt install cmake

Step 2: Install following packages

sudo apt install \

clang-tools lld llvm-dev libclang-dev liblld-10-dev \

libpng-dev libjpeg-dev libgl-dev \

python3-dev python3-numpy python3-scipy python3-imageio python3-pybind11

\libopenblas-dev libeigen3-dev libatlas-base-dev \

doxygen ninja-build

Step 3: Building halide with cmake

Halide$ cmake -G Ninja -DCMAKE\_BUILD\_TYPE=Release -S . -B build

dev@host:~/Halide$ cmake --build ./build

Step 4: CMake Presets

If you are using CMake 3.19+, we provide several presets to make the above

commands more convenient. The following CMake preset commands correspond

to the longer ones above.

> cmake --preset=msvc-release # Ninja generator, MSVC compiler, Release

build

> cmake --preset=win64 # VS 2019 generator, 64-bit build

> cmake --preset=win32 # VS 2019 generator, 32-bit build

$ cmake --preset=gcc-release # Ninja generator, GCC compiler, Release build

$ cmake --list-presets # Get full list of presets.

Step 5: Installing

Once built, Halide will need to be installed somewhere before using it in a

separate project. On any platform, this means running the cmake --

install command in one of two ways. For a single-configuration generator (like

Ninja), run either:

dev@host:~/Halide$ cmake --install ./build --prefix /path/to/Halide-install

> cmake --install .\build --prefix X:\path\to\Halide-install

For a multi-configuration generator (like Visual Studio) run:

dev@host:~/Halide$ cmake --install ./build --prefix /path/to/Halide-install --

config Release

> cmake --install .\build --prefix X:\path\to\Halide-install --config Release

