Installing the Spike RISC-V Simulator for Linux

Spike is a RISC-V Simulator. Here are the steps to installing Spike from source for 32bit simulation.

For Windows OS, you can use WSL or virtual machine.

Dependencies:

Please first executing the following command

```
sudo apt-get install autoconf automake autotools-dev curl python3 \
libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex \
texinfo gperf libtool patchutils bc zlib1g-dev libexpat-dev
```

Device Tree Compiler

```
# Arch Linux
sudo pacman -S dtc

# Ubuntu / Debian
sudo apt install device-tree-compiler
```

Installing Spike

Make a new directory for Spike:

```
mkdir RISCV
cd RISCV
```

Clone the repos for the RISCV GNU toolchain, proxy kernel (pk), and Spike itself

git clone --recursive https://github.com/riscv/riscv-gnu-toolchain
The toolchain source code is bigger than 3G, so you can download it manually from our course homepage

Or

Baidu: https://pan.baidu.com/s/1104aCZiIUAEcHONKcI8GKw
Passwd: ui4j

git clone https://github.com/riscv/riscv-pk
git clone https://github.com/riscv/riscv-isa-sim

Set environment variables (preferably move these commands into .bashrc, .zshrc, or something similar)

export RISCV=/path/from/home/to/RISCV (Your own RISCV path)
export PATH=$PATH:$RISCV/bin

Build the toolchain (this will take a while so make some tea or something)

cd riscv-gnu-toolchain
mkdir build
cd build
../configure --prefix=$RISCV --with-arch=rv32i
make

Now, to build the Proxy Kernel

cd ../riscv-pk
mkdir build
cd build
../configure --prefix=$RISCV --host=riscv32-unknown-elf
make
make install

Finally to build Spike

cd ../riscv-isa-sim

```
mkdir build
cd build
../configure --prefix=$RISCV --enable-histogram
make
make install
```

Check if the installation is successful

Create a test.c, and enter the following codes:

```c
#include <stdio.h>
int main() {
    printf("Hello world!\n");
}
```

Then

```
riscv32-unknown-elf-gcc test.c

/path/from/home/to/RISCV/riscv-isa-sim/build/spike --isa=RV32IMAC \
/path/from/home/to/RISCV/riscv-pk/build/pk a.out
```