

RISC-V Spike Simulator SDK

Fast Instruction To Simulate a Linux Kernel

In the recent version of the riscv-tools and freedom-u-sdk, both of them removed the support of the spike simulator, and tutorials about running Linux on spike is using static compiled busybox, which is not suitable for real test environments. Spike is the simplest simulator of RISC-V, it has a very clear description of the instructions, you can apply your ideas and check it for a quick try. The **RISC-V Spike Simulator SDK** wants to help people to test design with 64-bit Linux environment on Spike easily, the basic framework is based on Freedom U SDK version 1.0.

This SDK will provide great convenience for the following people who:

- want to run upstream version Linux on Spike
- want to test design in Linux environment with modified spike, gcc and other extension in toolchain
- want to have a overview about embedded system development

This SDK follows the newest Linux Kernel, GNU toolchain and Spike, the functions of each folder in the project are as follows:

Folder	Description	Version
repo/buildroot	Build initramfs	2021.2.x
repo/linux	Linux Kernel	5.12.0
repo/riscv-gnu-toolchain	GNU Compiler Toolchain	gcc 10.2.0 ld 2.36
repo/riscv-(isa-sim,pk)	Simulator & Bootloader	master
repo/opensbi	Supervisor / Bootloader	master
conf	config for SDK	

Quickstart

Build dependencies on Ubuntu 16.04/18.04/22.04:

```
$ sudo apt-get install device-tree-compiler autoconf automake autotools-dev \
curl libmpc-dev libmpfr-dev libgmp-dev gawk build-essential bison flex texinfo \
gperf libtool patchutils bc zlib1g-dev libexpat-dev python-dev python3-dev unzip \
libglib2.0-dev libpixmap-1-dev git rsync wget cpio libncurses-dev
```

Get the simple repository from git:

```
$ git clone https://github.com/riscv-zju/riscv-rss-sdk.git
```

Go to directory and run quickstart:

```
$ sh quickstart.sh
```

Get whole system dependencies of code:

```
$ git submodule update --init --recursive --progress
```

Set the RISC-V director as a variable and add it to PATH:

```
$ export RISC_V=/home/riscv/riscv-rss-sdk
```

```
$ export PATH=$PATH:$RISC_V/bin
```

Cross compile the gcc tool chain for the target ISA of processor for Now we compile for RV64i

```
$ cd repo/riscv-gnu-toolchain/  
$ mkdir build  
$ cd build  
$ ../configure --prefix=$RISCV --with-arch=rv64i  
$ make
```

after complete the compilation you can see the all GCC compiler and library in RISCV and RISCV/bin directories. The commands started with “riscv64-unknown-elf-” that used for cross compile a code for RISC-V processor. In addition we need cross compile gcc for linux kernel with next command.

```
$ make linux
```

After that, the compiler for linux will be appear in RISCV/bin directory with start in name “riscv64-unknown-linux-gnu-”. Now you have all tools for compile all we need for simulations.

Now, to build the Proxy Kernel should execute following instructions:

```
$ cd $RISCV/repo/riscv-pk  
$ mkdir build  
$ cd build  
$ ../configure --prefix=$RISCV --host=riscv64-unknown-elf  
$ make  
$ make install
```

Finally to build Spike is need to run next instructions:

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

Now check the simulator with a simple code:

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

Then

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

`$RISCV/repo/riscv-isa-sim/build/spike --isa=RV64I $RISCV/repo/riscv-pk/build/pk a.out`
if “Hello world!” appeared in the terminal it’s mean you have every thing for compilation and simulations for linux kernel.

Now we should select the configuration of the linux we want to compile and try to select configuration you needed. If you don’t have any idea what you should use remember in “Target options” is need to select the right processor’s architecture. So try the bellow commands to see the configuration menu.

```
$ cd $RISCV/  
$ make buildroot_initramfs-menuconfig
```

finally for run the bellow command:

```
$ make
```