■ RISC-V 関連のツールを整理

わからない状態であれこれやってた環境を一度整理.
https://github.com/riscv を 参照しつつ.

コンパイラツールチェイン

https://github.com/riscv/riscv-gnu-toolchain を 参照して，手順通り，

```
$ git clone --recursive https://github.com/riscv/riscv-gnu-toolchain
$ ./cnfigure --prefix=$HOME/tools/riscv
$ make
$ make linux
```

おわったらパス通しとくといい.

```
$ export RISCV=$HOME/tools/riscv
$ export PATH=$RISCV/bin:$PATH
```

シミュレーション環境 (Qemu)

https://github.com/riscv/riscv-qemu によると,
upstream にマージされたよう.
https://wiki.qemu.org/Documentation/Platforms/RISCV を参照して,

```
$ wget https://download.qemu.org/qemu-3.1.0.tar.xz
$ tar xvf qemu-3.1.0.tar.xz
$ cd qemu-3.1.0
$ ./configure --target-list=riscv64-softmmu
$ make -j8
$ make install
```

なお，configure 時に,

```
WARNING: Use of SDL 1.2 is deprecated and will be removed in
WARNING: future releases. Please switch to using SDL 2.0
```

とかいわれたので,

```
sudo apt install libsdl2-dev
```

とした.

Linux 動かす (Fedora のイメージもってきて動かす)

https://wiki.qemu.org/Documentation/Platforms/RISCV を 参照しつつ,
[RISC-V 版 QEMU で Linux を立ち上げる試行 http://msyksphinz.hatenablog.com/entry/2018/05/07/040000] に従って,

```
$ wget https://fedorapeople.org/groups/risc-v/disk-images/bbl
```

```
$ wget https://fedorapeople.org/groups/risc-v/disk-images/vmlinux
$ wget https://fedorapeople.org/groups/risc-v/disk-images/stage4-disk.img.xz
$ xzdec -d stage4-disk.img.xz > stage4-disk.img
```

とリソースを用意して，実行.

```
$ qemu-system-riscv64 \
  -nographic \
  -machine virt \
  -smp 4 \
  -m 2G \
  -kernel bbl \
  -object rng-random,filename=/dev/urandom,id=rng0 \
  -device virtio-rng-device,rng=rng0 \
  -append "console=ttyS0 ro root=/dev/vda" \
  -device virtio-blk-device,drive=hd0 \
  -drive file=stage4-disk.img,format=raw,id=hd0 \
  -device virtio-net-device,netdev=usernet \
  -netdev user,id=usernet,hostfwd=tcp::10000-:22
```

xzdec がなければ,

```
$ sudo apt install xzdec
```

としてインストールすればいい.
Fedora が起動すると，root/riscv でログイン．gcc とかも入っている.

```
[root@stage4 ~]# uname -a
Linux stage4.fedoraproject.org 4.19.0-rc8 #1 SMP Wed Oct 17 15:11:25 UTC 2018 riscv64 riscv64
riscv64 GNU/Linux
[root@stage4 ~]# gcc --version
gcc (GCC) 7.3.1 20180303 (Red Hat 7.3.1-5)
Copyright (C) 2017 Free Software Foundation, Inc.
This is free software; see the source for copying conditions.  There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

[root@stage4 ~]#
```

命令レベルシミュレータ (spike) 動かす

https://github.com/riscv/riscv-tools を 参照しつつ,

```
$ git clone https://github.com/riscv/riscv-tools
$ git submodule update --init --recursive
$ export RISCV=$HOME/tools/riscv
$ ./build.sh
```

インストールできたら,

```
$ cat > hello.c
#include <stdio.h>
int main(int argc, char **argv)
{
    printf("Hello RISC-V\n");
}
$ riscv64-unknown-elf-gcc -o hello hello.c
$ spike pk test
```

とか.

```
$ spike -d pk hello
```

として

```
:r
```

とかすると，実行命令列のダンプが表示される．
詳細は, https://github.com/riscv/riscv-isa-sim/tree/2710fe575e7e6a4e2418224f8d254d5ca31f6c0e を見る

自分で Linux ビルドする (BusyBear 使う)

https://risc-v-getting-started-guide.readthedocs.io/en/latest/linux-qemu.html がわかりやすいかな？

```
$ export WORKDIR=$(pwd)/riscv-linux
$ mkdir $WORKDIR; cd $WORKDIR
$ git clone https://github.com/torvalds/linux
$ git clone https://github.com/riscv/riscv-pk
$ git clone https://github.com/michaeljclark/busybear-linux
```

まずは Linux

```
$ cd $WORKDIR/linux
$ git checkout v4.19
$ cp ../busybear-linux/conf/linux.config .config
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- olddefconfig
```

カーネルコンフィグレーションが正しいか,

```
$ cat .config | grep -e ARCH_RV64I -e CMODEL_MEDANY -e CONFIG_SIFIVE_PLIC
```

とかして,

```
CONFIG_ARCH_RV64I=y
CONFIG_CMODEL_MEDANY=y
CONFIG_SIFIVE_PLIC=y
```

となっているのを確認する．なんか違ったら,

```
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- nconfig
```

で設定．最後にカーネルのビルド．

```
$ make ARCH=riscv CROSS_COMPILE=riscv64-unknown-linux-gnu- vmlinux -j8
```

次に BBL 作る．

```
$ cd $WORKDIR/riscv-pk
$ mkdir build && cd build
$ ../configure --enable-logo --host=riscv64-unknown-elf --with-payload=../../linux/vmlinux
$ make -j8
```

最後に Busybear 作る

```
$ cd cd $WORKDIR/busybear-linux
$ make -j8
```

途中，BBL 作ろうとするところでパスの問題でコケル.
うまくパスあわせてもいいんだろうけど，既に別に作っているので,
scripts/build.sh の該当部分をコメントアウトしてしまうことにする.

```
#test -d build/riscv-pk || mkdir build/riscv-pk
#test -x build/riscv-pk/bbl || (
#    cd build/riscv-pk
#    ../../src/riscv-pk/configure \
#        --host=${CROSS_COMPILE%-} \
#        --with-payload=../linux-${LINUX_KERNEL_VERSION}/vmlinux
#    make -j$(nproc)
#)
```

最後，ディスクイメージ作る時に sudo で root 権限が要求される.
起動.

```
$ qemu-system-riscv64 \
    -nographic \
    -machine virt \
    -kernel riscv-pk/build/bbl \
    -append "root=/dev/vda ro console=ttyS0" \
    -drive file=busybear-linux/busybear.bin,format=raw,id=hd0 \
    -device virtio-blk-device,drive=hd0
```

root/busybear でログインできた.

■たてなおし

バタバタとしてたのが一段落 & 来週もバタバタするのが分かっているので,
書き散らかしになってたスクリプトやら，やるべきタスクやらを整理して,
たてなおし.