Running Under an emulator

We will be using SPIKE instead of QEMU.QEMU requires OpenSBI firmware and it needs to be linked to the RISC V toolchain.I tried that but it keeps on failing even though the RISC V toolchain has the required Linker file.

The errors:

kor-tac@kor-tac-VirtualBox:~/Desktop/RISC$ qemu-system-riscv32 -machine virt -nographic -kernel hello.elf qemu-system-riscv32: Unable to find the RISC-V BIOS "opensbi-riscv32-generic-fw\_dynamic.bin"

kor-tac@kor-tac-VirtualBox:~/Desktop$ sudo apt install build-essential git device-tree-compiler gcc-riscv32-unknown-elf Reading package lists... Done Building dependency tree... Done Reading state information... Done E: Unable to locate package gcc-riscv32-unknown-elf

kor-tac@kor-tac-VirtualBox:~/Desktop/opensbi$ make PLATFORM=generic FW\_ABI=ilp32 CROSS\_COMPILE=riscv32-unknown-elf- PLATFORM\_RISCV\_ISA=rv32imac Makefile:196: \*\*\* Your linker does not support creating PIEs, opensbi requires this.. Stop.

So we will be using SPIKE

For SPIKE we need a Proxy kernel(PK) so install that

git clone <https://github.com/riscv/riscv-pk.git>

cd riscv-pk

mkdir build

cd build

I got this error

kor-tac@kor-tac-VirtualBox:~/Downloads/riscv-pk/build$ make

riscv32-unknown-elf-gcc -MMD -MP -Wall -Werror -D\_\_NO\_INLINE\_\_ -mcmodel=medany -O2 -std=gnu99 -Wno-unused -Wno-attributes -fno-delete-null-pointer-checks -fno-PIE -DBBL\_LOGO\_FILE=\"bbl\_logo\_file\" -DMEM\_START=0x80000000 -fno-stack-protector -U\_FORTIFY\_SOURCE -DBBL\_PAYLOAD=\"bbl\_payload\" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../pk/file.c

riscv32-unknown-elf-gcc -MMD -MP -Wall -Werror -D\_\_NO\_INLINE\_\_ -mcmodel=medany -O2 -std=gnu99 -Wno-unused -Wno-attributes -fno-delete-null-pointer-checks -fno-PIE -DBBL\_LOGO\_FILE=\"bbl\_logo\_file\" -DMEM\_START=0x80000000 -fno-stack-protector -U\_FORTIFY\_SOURCE -DBBL\_PAYLOAD=\"bbl\_payload\" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../pk/syscall.c

../pk/pk.h: Assembler messages:

../pk/pk.h:59: Error: unrecognized opcode `csrr a5,timeh', extension `zicsr' required

../pk/pk.h:60: Error: unrecognized opcode `csrr a3,time', extension `zicsr' required

../pk/pk.h:61: Error: unrecognized opcode `csrr a4,timeh', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr a5,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr a1,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a4,cycleh', extension `zicsr' required

../pk/pk.h:91: Error: unrecognized opcode `csrr a5,instreth', extension `zicsr' required

../pk/pk.h:92: Error: unrecognized opcode `csrr a1,instret', extension `zicsr' required

../pk/pk.h:93: Error: unrecognized opcode `csrr a4,instreth', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr a1,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr a0,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a5,cycleh', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr a4,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr a5,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a3,cycleh', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr a1,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr a0,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a5,cycleh', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr s0,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr s1,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a5,cycleh', extension `zicsr' required

../pk/pk.h:75: Error: unrecognized opcode `csrr s0,cycleh', extension `zicsr' required

../pk/pk.h:76: Error: unrecognized opcode `csrr s1,cycle', extension `zicsr' required

../pk/pk.h:77: Error: unrecognized opcode `csrr a5,cycleh', extension `zicsr' required

make: \*\*\* [Makefile:336: syscall.o] Error 1

To solve first we need to find the endcoding file

find . -name encoding.h

kor-tac@kor-tac-VirtualBox:~/Downloads/riscv-pk$ find . -name encoding.h ./machine/encoding.h

Then

kor-tac@kor-tac-VirtualBox:~/Downloads/riscv-pk/build$ make CFLAGS="-march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE=\"bbl\_logo\_file\" -DBBL\_PAYLOAD=\"bbl\_payload\""

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE="bbl\_logo\_file" -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../pk/mmap.c

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE="bbl\_logo\_file" -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../pk/usermem.c

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE="bbl\_logo\_file" -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../pk/entry.S

riscv32-unknown-elf-ar rcv libpk.a file.o syscall.o handlers.o frontend.o elf.o console.o mmap.o usermem.o entry.o

a - file.o

a - syscall.o

a - handlers.o

a - frontend.o

a - elf.o

a - console.o

a - mmap.o

a - usermem.o

a - entry.o

riscv32-unknown-elf-ranlib libpk.a

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE="bbl\_logo\_file" -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../bbl/logo.c

touch bbl\_logo\_file

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_LOGO\_FILE="bbl\_logo\_file" -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../bbl/raw\_logo.S

../bbl/raw\_logo.S: Assembler messages:

../bbl/raw\_logo.S:8: Error: missing string

make: \*\*\* [Makefile:336: raw\_logo.o] Error 1

To debug this we need to the raw\_logo\_.S file

find . -name raw\_logo.S

Then the file will look like this:

// See LICENSE for license details.

#include "encoding.h"

.section .rodata

.globl logo

logo:

.incbin BBL\_LOGO\_FILE

.byte 0

~

~

Convert that .incbin BBL\_LOGO\_FILE to .incbin “bbl\_logo\_file”

Then

kor-tac@kor-tac-VirtualBox:~/Downloads/riscv-pk/build$ make CFLAGS="-march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD=\"bbl\_payload\""

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../bbl/raw\_logo.S

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../dummy\_payload/dummy\_payload.c

echo "int \_dummy\_payload( int arg ) { return arg; }" > \_dummy\_payload.c

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c \_dummy\_payload.c

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../dummy\_payload/dummy\_entry.S

../dummy\_payload/dummy\_entry.S: Assembler messages:

../dummy\_payload/dummy\_entry.S:23: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:24: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:25: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:26: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:27: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:28: Warning: unterminated string; newline inserted

../dummy\_payload/dummy\_entry.S:29: Warning: unterminated string; newline inserted

riscv32-unknown-elf-ar rcv libdummy\_payload.a \_dummy\_payload.o dummy\_entry.o

a - \_dummy\_payload.o

a - dummy\_entry.o

riscv32-unknown-elf-ranlib libdummy\_payload.a

riscv32-unknown-elf-gcc -Wl,--build-id=none -nostartfiles -nostdlib -static -fno-stack-protector -o dummy\_payload dummy\_payload.o -L. -ldummy\_payload -lgcc -Wl,--defsym=MEM\_START=0x80000000,-T,../dummy\_payload/dummy\_payload.lds

/home/kor-tac/Downloads/opt/riscv/bin/../lib/gcc/riscv32-unknown-elf/14.2.0/../../../../riscv32-unknown-elf/bin/ld: warning: dummy\_payload has a LOAD segment with RWX permissions

if riscv32-unknown-elf-readelf -h dummy\_payload 2> /dev/null > /dev/null; then riscv32-unknown-elf-objcopy -O binary --set-section-flags .bss=alloc,load,contents dummy\_payload bbl\_payload; else cp dummy\_payload bbl\_payload; fi

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -DBBL\_PAYLOAD="bbl\_payload" -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../bbl/payload.S

../bbl/payload.S: Assembler messages:

../bbl/payload.S:18: Error: missing string

make: \*\*\* [Makefile:336: payload.o] Error 1

For this find the payload.S

In that change .incbin BBL\_PAYLOAD this to .incbin "bbl\_payload"

In file included from ../machine/misaligned\_ldst.c:5:

../machine/unprivileged\_memory.h:69:49: warning: 'always\_inline' function might not be inlinable unless also declared 'inline' [-Wattributes]

69 | static uintptr\_t \_\_attribute\_\_((always\_inline)) get\_insn(uintptr\_t mepc, uintptr\_t\* mstatus)

| ^~~~~~~~

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../machine/misaligned\_vec\_ldst.c

In file included from ../machine/misaligned\_vec\_ldst.c:5:

../machine/unprivileged\_memory.h:69:49: warning: 'always\_inline' function might not be inlinable unless also declared 'inline' [-Wattributes]

69 | static uintptr\_t \_\_attribute\_\_((always\_inline)) get\_insn(uintptr\_t mepc, uintptr\_t\* mstatus)

| ^~~~~~~~

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsr -mabi=ilp32 -DMEM\_START=0x80000000 -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../machine/flush\_icache.c

../machine/flush\_icache.c: Assembler messages:

../machine/flush\_icache.c:4: Error: unrecognized opcode `fence.i', extension `zifencei' required

make: \*\*\* [Makefile:336: flush\_icache.o] Error 1

For this : make CFLAGS="-march=rv32imaczicsrzifencei -mabi=ilp32 -DMEM\_START=0x80000000"

kor-tac@kor-tac-VirtualBox:~/Downloads/riscv-pk/build$ make CFLAGS="-march=rv32imaczicsrzifencei -mabi=ilp32 -DMEM\_START=0x80000000"

riscv32-unknown-elf-gcc -MMD -MP -march=rv32imaczicsrzifencei -mabi=ilp32 -DMEM\_START=0x80000000 -I. -I../pk -I../bbl -I../softfloat -I../dummy\_payload -I../machine -I../util -c ../machine/flush\_icache.c

riscv32-unknown-elf-gcc: error: '-march=rv32imaczicsrzifencei': extension 'zicsrzifencei' starts with 'z' but is unsupported standard extension

make: \*\*\* [Makefile:336: flush\_icache.o] Error 1

For this: make CFLAGS="-march=rv32imac\_zicsr\_zifencei -mabi=ilp32 -DMEM\_START=0x80000000"

Now in the folder containing the C file create a linker file with the details(hello.lds)

ENTRY(\_start)

SECTIONS {

. = 0x80000000;

.text : { \*(.text\*) }

.rodata : { \*(.rodata\*) }

.data : { \*(.data\*) }

.bss : { \*(.bss\*) }

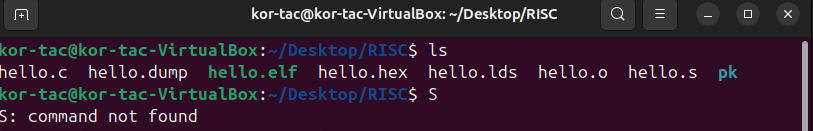
\_end = .;

}

kor-tac@kor-tac-VirtualBox:~/Desktop/RISC$ touch hello.lds kor-tac@kor-tac-VirtualBox:~/Desktop/RISC$ riscv32-unknown-elf-ld -T hello.lds hello.o -o hello.elf riscv32-unknown-elf-ld: warning: cannot find entry symbol \_start; defaulting to 80000000 riscv32-unknown-elf-ld: hello.o: in function main': hello.c:(.text+0x10): undefined reference to printf'

Then either add the pk file to path or copy and paste it to the current working directory

ln -s /home/kor-tac/Downloads/riscv-pk/build/pk pk



Then to get the output of the C file type

spike --isa=rv32imac pk hello.elf

