

```

qemu-system-aarch64 \
  -machine virt \
  -cpu cortex-a57 \
  -smp 4 \
  -m 512 \
  -nographic \
  -kernel tmp/deploy/images/qemuarm64/Image \
  -drive file=tmp/deploy/images/qemuarm64/core-image-minimal-
qemuarm64-20250721222805.rootfs.ext4,format=raw,if=virtio \
  -append "console=ttyAMA0 root=/dev/vda rw"

```

```

qemu-system-aarch64 \
  -machine virt \
  -cpu cortex-a57 \
  -smp 4 \
  -m 512 \
  -nographic \
  -kernel tmp/deploy/images/qemuarm64/Image \
  -drive file=$(ls -t tmp/deploy/images/qemuarm64/core-image-minimal-qemuarm64-*.rootfs.ext4 |
head -1),format=raw,if=virtio \
  -append "console=ttyAMA0 root=/dev/vda rw"

```

debug:

```

qemu-system-aarch64 \
  -machine virt \
  -cpu cortex-a57 \
  -smp 4 \
  -m 512 \
  -nographic \
  -kernel tmp/deploy/images/qemuarm64/Image \
  -drive file=$(ls -t tmp/deploy/images/qemuarm64/core-image-minimal-qemuarm64-*.rootfs.ext4 |
head -1),format=raw,if=virtio \
  -append "console=ttyAMA0 root=/dev/vda rw nokaslr" \
  -s -S

```

```
gdb-multiarch tmp/work/qemuarm64-poky-linux/linux-yocto/*/linux-qemuarm64-standard-build/vmlinux
```

```
gdb-multiarch -nx vmlinux
```

```

//////////build//////////
source oe-init-build-env
cd build
bitbake -c menuconfig virtual/kernel
bitbake -c savedefconfig virtual/kernel

bitbake -c diffconfig virtual/kernel

```

```

# tmp/work/qemuarm64-poky-linux/linux-yocto/*/linux-qemuarm64-standard-build/.config
mkdir -p poky/meta/recipes-kernel/linux/linux-yocto/qemuarm64

```

```

cp poky/build/tmp/work/qemuarm64-poky-linux/linux-yocto/*/fragment.cfg \
  poky/meta/recipes-kernel/linux/linux-yocto/qemuarm64/debug-info.cfg

```

```

cat > poky/meta/recipes-kernel/linux/linux-yocto_%.bbappend <<'EOF'
FILESEXTRAPATHS:prepend := "${THISDIR}/linux-yocto/qemuarm64:"
SRC_URI += "file://debug-info.cfg"
EOF

```

```
cat poky/meta/recipes-kernel/linux/linux-yocto_%.bbappend
```

```
bitbake-layers show-appends | grep linux-yocto
```

```
at poky/meta/conf/layer.conf | grep BBFILES
```

```
BBFILES += "${LAYERDIR}/recipes-*/*/*.bb \  
           ${LAYERDIR}/recipes-*/*/*.bbappend"
```

```
////
```

```
cd poky/build  
bitbake virtual/kernel -c cleanall  
bitbake virtual/kernel
```

```
grep CONFIG_DEBUG_INFO tmp/work/qemuarm64-poky-linux/linux-yocto/*/linux-qemuarm64-standard-  
build/.config  
CONFIG_DEBUG_INFO=y  
# CONFIG_DEBUG_INFO_REDUCED is not set  
# CONFIG_DEBUG_INFO_COMPRESSED is not set  
# CONFIG_DEBUG_INFO_SPLIT is not set  
CONFIG_DEBUG_INFO_DWARF_TOOLCHAIN_DEFAULT=y  
# CONFIG_DEBUG_INFO_DWARF4 is not set  
# CONFIG_DEBUG_INFO_DWARF5 is not set  
# CONFIG_DEBUG_INFO_BTf is not set  
harrison@armubuntu:~/yocto-arm64/poky/build$ file tmp/work/qemuarm64-poky-linux/linux-yocto/*/  
linux-qemuarm64-standard-build/vmlinux  
tmp/work/qemuarm64-poky-linux/linux-yocto/5.15.186+gitAUTOINC+5932fcfa69_9e9701d723-r0/linux-  
qemuarm64-standard-build/vmlinux: ELF 64-bit LSB pie executable, ARM aarch64, version 1 (SYSV),  
statically linked, BuildID[sha1]=b47e9bdc56bce7079fe886e777a78ba8ef4f7bf6, with debug_info, not  
stripped
```

```
//////////add perf,ftrace,bpftrace etc.///
```

1. 创建目录

```
mkdir -p poky/meta/recipes-kernel/linux/linux-yocto/qemuarm64
```

2. 写入 fragment.cfg

```
nano poky/meta/recipes-kernel/linux/linux-yocto/qemuarm64/trace-perf-bpf.cfg
```

```
# ==== 基础 Debug & tracefs 支持 ====
```

```
CONFIG_DEBUG_FS=y  
CONFIG_TRACEFS_SUPPORT=y  
CONFIG_TRACING=y  
CONFIG_FTRACE=y  
CONFIG_FUNCTION_TRACER=y  
CONFIG_FUNCTION_GRAPH_TRACER=y  
CONFIG_DYNAMIC_FTRACE=y  
CONFIG_DYNAMIC_FTRACE_WITH_REGS=y  
CONFIG_KPROBES=y  
CONFIG_KPROBE_EVENTS=y  
CONFIG_UPROBES=y  
CONFIG_UPROBE_EVENTS=y  
CONFIG_STACKTRACE=y
```

```
# ==== perf 所需 ====
```

```
CONFIG_PERF_EVENTS=y  
CONFIG_PROFILING=y  
CONFIG_EVENT_TRACING=y
```

```
# ==== BPF/eBPF 基础 ====
```

```
CONFIG_BPF=y  
CONFIG_BPF_SYSCALL=y  
CONFIG_BPF_JIT=y  
CONFIG_BPF_JIT_ALWAYS_ON=y  
CONFIG_HAVE_EBPF_JIT=y  
CONFIG_DEBUG_INFO_BTf=y
```

```

# ==== BTF & BPF CO-RE 支持 ====
CONFIG_DEBUG_INFO=y
CONFIG_DEBUG_INFO_DWARF_TOOLCHAIN_DEFAULT=y
CONFIG_DEBUG_INFO_BTF=y

# ==== 让 bpftrace 可以用 ====
CONFIG_FTRACE_SYSCALLS=y
CONFIG_CONTEXT_SWITCH_TRACER=y
CONFIG_SCHED_TRACER=y
CONFIG_TASKS_TRACE_RCU=y
# 把上面的内容粘进去，保存退出
3. 在 bbappend 里挂载
echo 'FILESEXTRAPATHS:prepend := "${THISDIR}/linux-yocto/qemuarm64:"' >> poky/meta/recipes-
kernel/linux/linux-yocto_%.bbappend
echo 'SRC_URI += "file://trace-perf-bpf.cfg"' >> poky/meta/recipes-kernel/linux/linux-
yocto_%.bbappend
4. 重新编译内核
bitbake -c cleanall virtual/kernel
bitbake virtual/kernel
5. 重启 qemu 后验证
mount | grep tracefs          # 应该能看到 tracefs
perf --version                # perf 可用
bpftrace -l | head            # bpftrace 可用

6. 给 rootfs 增加 perf & bpftrace
Yocto 里 core-image-minimal.bb 默认只带基础 busybox，要扩展：
方法 1：在 local.conf 里加包
echo 'IMAGE_INSTALL:append = " perf bpftrace bcc"' >> poky/build/conf/local.conf
方法 2：自定义 image recipe
复制 core-image-minimal.bb 改成 core-image-debug.bb，额外加：
IMAGE_INSTALL:append = " perf bpftrace bcc"
bitbake core-image-debug

```