ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

МОСКОВСКИЙ ИНСТИТУТ ЭЛЕКТРОНИКИ И МАТЕМАТИКИ
им. А.Н. ТИХОНОВА

Рекомендации по работе с QEMU и Docker на Unix-системах в рамках выполнения лабораторных работ курса «Программирование на языке ассемблера RISC-V»

Москва 2025

Оглавление

1 Введение

1.1 Установка QEMU

Для установки QEMU на машину достаточно следовать инструкциям с официального сайта: https://www.qemu.org/download/#source.

Так, для установки пакета, как правило, достаточно воспользоваться пакетным менеджером используемой системы:

– для Debian/Ubuntu – apt:

Листинг 1 – Установка пакета QEMU и необходимых зависимостей на Ubuntu

```
sudo apt-get install -y qemu-system-riscv64 opensbi u-boot-qemu
```

– для macos – brew:

Листинг 2 – Установка пакета QEMU на macos

```
brew install qemu
```

1.2 Установка Docker

Для установки Docker необходимо воспользоваться документацией с официального сайта: https://docs.docker.com/engine/install/.

2 Работа с QEMU

Ниже приведены рекомендации по работе с QEMU, однако руководствоваться можно и официальной документацией установки RISC-V Ubuntu в QEMU: https://canonical-ubuntu-boards.readthedocs-hosted.com/en/latest/how-to/qemu-riscv/.

Для создания виртуальной машины на базе RISC-V требуется скачать предустановленный образ Ubuntu Server (LTS): https://ubuntu.com/download/risc-v, а также следующие системные файлы в одну папку с образом сервера:

- fw_jump.el,
- uboot.elf.

2.1 Создание виртуальной машины

Для удобства использования ранее скачанные файлы необходимо разместить в одной директории (лучше избегать кириллицы в пути директории).

Кроме того, перед запуском QEMU, рекомендуется увеличить доступный для виртуальной машины (далее ВМ) размер хранилища:

Листинг 3 – Увеличение размера образа для ВМ

```
sudo qemu-img resize ubuntu-24.04.1-server-riscv64.img +10G
```

Далее рекомендуется создать bash-скрипт, который позволит упростить запуск ВМ (см. листинг 4).

Листинг 4 – bash-скрипт запуска ВМ QEMU

```
#!/bin/bash

SCRIPT_DIR=$(cd -- "$(dirname -- "${BASH_SOURCE[0]}")"
&>/dev/null && pwd)

qemu-system-riscv64 \
  -machine virt -nographic -m 2048 -smp 4 \
  -bios $SCRIPT_DIR/qemu-elfs/fw_jump.elf \
  -kernel $SCRIPT_DIR/qemu-elfs/uboot.elf \
  -drive file=$SCRIPT_DIR/ubuntu-24.04.1-server-
riscv64.img,format=raw,if=virtio \
  -device virtio-net-device,netdev=net0 -netdev
user,id=net0,hostfwd=tcp::2222-:22 \
  -append "root=LABEL=rootfs console=ttyS0" \
  -full-screen
```

Инструкции выше позволяют запустить виртуальную машину, к которой можно будет подключиться на локальном хосте на порту 2222.

2.2 Настройка ssh

Для подключения к запущенной ВМ можно использовать ssh, который необходимо установить в ВМ:

Листинг 5 – Установка и настройка ssh в ВМ

```
sudo apt update && sudo apt install -y openssh-server
sudo systemctl enable --now ssh
sudo systemctl status ssh
```

Последняя команда должна подтвердить успешный запуск ssh-сервера.

Теперь с хост-системы должно успешно выполняться подключение:

Листинг 6 – Подключение к ВМ по ssh

```
ssh -p 2222 localhost
```

2.3 Дополнительно

Для удобства работы с ВМ рекомендуется воспользоваться пакетом tmux, который позволит удобно создавать и управлять многими рабочими пространствами:

Листинг 7 – Установка tmux

```
sudo apt install -y tmux
```

Полезные материалы по использованию tmux:

– использование tmux,

– команды tmux.

При проблемах с корректным выводом консоли в ВМ, удобно восстановить консоль следующими командами:

Листинг 8 – Восстановление работы консоли

```
stty sane && echo -e "\033c"
```

3 Работа с Docker

Альтернативой использованию QEMU в Unix-системах может служить Docker, поскольку поддерживает сборку и запуск контейнеров под разные архитектуры, используя QEMU. Для изучения возможностей Docker можно обратиться к официальной документации, ниже приведены основные команды для подготовки образа Ubuntu RISC-V.

3.1 Настройка Docker

Настройка Docker для возможности использования QEMU отличается от платформы: на macos (arm-чипы) все должно работать по умолчанию из-за особенностей архитектуры, при этом на Linux необходимо заранее выполнить следующую команду:

Листинг 9 – Настройка Docker на Linux

```
docker run --privileged --rm tonistiigi/binfmt --install all
```

3.2 Подготовка образа

Для создания готового для работы образа Ubuntu архитектуры riscv64 можно взять за основу образ riscv64/ubuntu версии 24.04 (или любой другой поддерживаемой версии). Ниже приведен Dockerfile, устанавливающий все требуемые пакеты и настраивающий локаль для поддержки вывода кириллицы:

Листинг 10 – Dockerfile для подготовки образа Ubuntu riscv64

```
FROM riscv64/ubuntu:noble

ENV DEBIAN_FRONTEND=noninteractive

RUN apt-get update && \
    apt-get -y upgrade && \
    apt-get -y install \
    tmux git neofetch \
```

Листинг 10 – Продолжение

```
    locales language-pack-ru \
    build-essential vim nano \
    qemu-user-static

ARG USR=root
WORKDIR /home/${USR}/

ARG LOCALE=ru_RU.UTF-8
ENV LANG=${LOCALE}
ENV LANGUAGE=${LOCALE}
ENV LCC_LANG=${LOCALE}
ENV LCC_ALL=${LOCALE}
RUN locale-gen ${LOCALE} && dpkg-reconfigure locales && \
    cd /home/${USR} && \
    echo -e "export LANG=${LOCALE}\nexport LANGUAGE=${LOCALE}" >
.bash_profile && \
    echo -e "export LC_LANG=${LOCALE}\nexport LC_ALL=${LOCALE}"
>> .bash_profile && \
    echo "source ~/.bash_profile" >> .bashrc

CMD ["/bin/bash"]
```

Предложенный выше код необходимо сохранить в файле с названием «Dockerfile», после чего, в консоли перейти в директорию с данным файлом и выполнить сборку образа:

Листинг 11 – Сборка образа

```
docker buildx build --platform linux/riscv64 -t riscv-ubuntu .
```

3.3 Запуск контейнера

После успешной сборки образа можно создать контейнер на его основе:

Листинг 12 – Создание контейнера

```
docker run --privileged --name riscv -h riscv-ubuntu \
-v /tmp:/host -p 7777:7777 -w /home -it riscv-ubuntu
```

Команда выше создает контейнер с именем riscv и добавляет связь с временной директорией на хосте (директория /host в контейнере соответствует директории /tmp на хосте).

Для запуска контейнера после его завершения достаточно выполнить следующие команды:

Листинг 13 – Запуск контейнера

```
docker start riscv && docker exec -it riscv /bin/bash
```

Для завершения работы контейнера достаточно выполнить следующую команду:

Листинг 14 – Выключение контейнера

```
docker stop riscv
```