

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

КАТЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Архітектура комп'ютерів

Лабораторна робота №3

«BeagleBone Black: Platform Bring Up with Upstream Components»

Виконав:
студент групи ІО-32
Крадожон М.

Перевірів(-ла):
Каплунов А.

Лабораторна робота №3

Мета

Створити невеликий C++ проект з трьома файлами (1 заголовок + 2 спп), створити Makefile і CMakeLists.txt, зібрати статичну або динамічну бібліотеку та виконуваний файл.

Хід роботи

Підготовка середовища

docker-compose.yml

```
services:
  lab3:
    image: ubuntu:24.04
    container_name: lab3
    tty: true
    stdin_open: true
    privileged: true
    environment:
      - DEBIAN_FRONTEND=noninteractive
    volumes:
      - ./workspace
    working_dir: /workspace
    command: bash /workspace/setup.sh
```

setup.sh

```
#!/bin/bash
set -e

apt-get update
apt-get install -y \
    git \
    vim \
    tree \
    curl \
    ccache \
    wget \
    make \
    libncurses-dev \
    libssl-dev \
    bc \
    bison \
    flex \
    xz-utils \
    gcc-9 \
    libgnutls28-dev \
    g++ \
    libmpc-dev \
    libmpfr-dev \
    libgmp-dev \
    gcc-arm-linux-gnueabi \
    g++-arm-linux-gnueabi \
    bzip2 \
    kmod \
    qemu-system-arm \
    cpio

update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 9

git config --global user.name "Lab3 User"
git config --global user.email "lab3@example.com"

ccache -M 5G

mkdir -p /workspace/repos
cd /workspace/repos

EABI_ARCHIVE=gcc-arm-8.3-2019.03-x86_64-arm-eabi.tar.xz
```

```
GNUEABI_ARCHIVE=gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf.tar.xz
```

```
if [ ! -d "/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi" ]; then
    if [ ! -f "$EABI_ARCHIVE" ]; then
        wget https://developer.arm.com/-/media/Files/downloads/gnu-a/8.3-2019.03/binre
    fi
    tar xJvf $EABI_ARCHIVE -C /opt/
fi
```

```
if [ ! -d "/opt/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabihf" ]; then
    if [ ! -f "$GNUEABI_ARCHIVE" ]; then
        wget https://developer.arm.com/-/media/Files/downloads/gnu-a/8.3-2019.03/binre
    fi
    tar xJvf $GNUEABI_ARCHIVE -C /opt/
fi
```

```
if [ ! -d "u-boot" ]; then
    git clone https://gitlab.denx.de/u-boot/u-boot.git
fi
```

```
cd /workspace/repos/u-boot
```

```
LAST_TAG=$(git tag | grep -v rc | tail -n 1)
git checkout "$LAST_TAG"
```

```
curl https://patchwork.ozlabs.org/series/130450/mbox/ | git am
```

```
cd /workspace/repos
```

```
if [ ! -d "linux-stable" ]; then
    git clone git://git.kernel.org/pub/scm/linux/kernel/git/stable/linux-stable.git
fi
```

```
cd /workspace/repos/linux-stable
git checkout linux-4.19.y
```

```
cd /workspace/repos
```

```
if [ ! -d "busybox" ]; then
    git clone git://git.busybox.net/busybox
fi
```

```
fi
```

```
cd /workspace/repos/busybox
```

```
git checkout 1_31_stable
```

```
cd /workspace/repos
```

```
exec bash
```

Побудова програмного забезпечення

Програмне забезпечення будувалося великою кількістю команд, тому тут буде лише список цих команд

```
#!/bin/bash
set -e

# U-Boot
cd /workspace/repos/u-boot

export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:$PATH
export CROSS_COMPILE='ccache arm-eabi-'
export ARCH=arm

make am335x_boneblack_defconfig

make -j4


# Linux kernel
cd /workspace/repos/linux-stable

export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-eabi/bin:$PATH
export CROSS_COMPILE='ccache arm-eabi-'
export ARCH=arm

mkdir -p fragments

cat > fragments/bbb.cfg <<'EOF'
# Use multi_v7_defconfig as a base for merge_config.sh
# --- USB ---
# Enable USB on BBB (AM335x)
CONFIG_USB_ANNOUNCE_NEW_DEVICES=y
CONFIG_USB_EHCI_ROOT_HUB_TT=y
CONFIG_AM335X_PHY_USB=y
CONFIG_USB_MUSB_TUSB6010=y
CONFIG_USB_MUSB_OMAP2PLUS=y
CONFIG_USB_MUSB_HDRC=y
CONFIG_USB_MUSB_DSPS=y
CONFIG_USB_MUSB_AM35X=y
CONFIG_USB_CONFIGFS=y
CONFIG_NOP_USB_XCEIV=y
# For USB keyboard and mouse
CONFIG_USB_HID=y
CONFIG_USB_HIDDEV=y
# For PL2303, FTDI, etc
CONFIG_USB_SERIAL=y
CONFIG_USB_SERIAL_PL2303=y
CONFIG_USB_SERIAL_GENERIC=y
```

```

CONFIG_USB_SERIAL_SIMPLE=y
CONFIG_USB_SERIAL_FTDI_SIO=y
# For USB mass storage devices (like flash USB stick)
CONFIG_USB_ULPI=y
CONFIG_USB_ULPI_BUS=y
# --- Networking ---
CONFIG_BRIDGE=y
# --- Device Tree Overlays (.dtbo support) ---
CONFIG_OF_OVERLAY=y
EOF

./scripts/kconfig/merge_config.sh \
arch/arm/configs/multi_v7_defconfig fragments/bbb.cfg

make -j4 zImage modules am335x-boneblack.dtb

# Busybox
cd /workspace/repos/busybox

export ARCH=arm
export PATH=/opt/gcc-arm-8.3-2019.03-x86_64-arm-linux-gnueabi/bin:$PATH
export CROSS_COMPILE="ccache arm-linux-gnueabi-"

make defconfig

make -j4

make install

mkdir -p _install/{boot,dev,etc\init.d,lib,proc,root,sys\kernel\debug,tmp}
mkdir -p _install/etc/init.d

cat > _install/etc/init.d/rcS << 'EOF'
#!/bin/sh
mount -t sysfs none /sys
mount -t proc none /proc
mount -t debugfs none /sys/kernel/debug
echo /sbin/mdev > /proc/sys/kernel/hotplug
mdev -s
EOF

chmod +x _install/etc/init.d/rcS
ln -s bin/busybox _install/init

```

```
INST_DIR="/workspace/repos/linux-stable"
cd _install/boot
cp ${INST_DIR}/arch/arm/boot/zImage .
cp ${INST_DIR}/arch/arm/boot/dts/am335x-boneblack.dtb .
cp ${INST_DIR}/System.map .
cp ${INST_DIR}/.config ./config
cd -

cd /workspace/repos/linux-stable
export INSTALL_MOD_PATH=/workspace/repos/busybox/_install
export ARCH=arm
make modules_install
cd -

${CROSS_COMPILE}readelf -d _install/bin/busybox | grep NEEDED

cd _install/lib
libc_dir=$( ${CROSS_COMPILE}gcc -print-sysroot)/lib
cp -a $libc_dir/*.so* .
cd -

echo '$MODALIAS=.* root:root 660 @modprobe "$MODALIAS"' > \
_install/etc/mdev.conf

echo 'root:x:0:' > _install/etc/group
echo 'root:x:0:0:root:/root:/bin/sh' > _install/etc/passwd
echo 'root::10933:0:99999:7:::' > _install/etc/shadow
echo "nameserver 8.8.8.8" > _install/etc/resolv.conf

# Flashing
cd /workspace/repos/busybox
cd _install
find . | cpio -o -H newc | gzip > ../rootfs.cpio.gz
cd ..

# Run the kernel
qemu-system-arm -kernel _install/boot/zImage -initrd rootfs.cpio.gz \
-machine virt -nographic -m 512 \
--append "root=/dev/ram0 rw console=ttyAMA0,115200 mem=512M"
```


Тестування програмного забезпечення

```
/ # uname -a
Linux (none) 4.19.325 #1 SMP Tue Oct 21 10:47:34 UTC 2025 armv7l GNU/Linux
/ # ls -l
total 0
drwxr-xr-x  2 root    root          0 Oct 21 12:48 bin
drwxr-xr-x  2 root    root          0 Oct 21 12:50 boot
drwxr-xr-x  3 root    root          0 Oct 21 13:02 dev
drwxr-xr-x  3 root    root          0 Oct 21 12:56 etc
lrwxrwxrwx  1 root    root        11 Oct 21 12:49 init -> bin/busybox
drwxr-xr-x  3 root    root          0 Oct 21 12:55 lib
lrwxrwxrwx  1 root    root        11 Oct 21 12:48 linuxrc -> bin/busybox
dr-xr-xr-x 91 root    root          0 Jan  1 1970 proc
drwxr-xr-x  2 root    root          0 Oct 21 12:49 root
drwxr-xr-x  2 root    root          0 Oct 21 12:48 sbin
dr-xr-xr-x 12 root    root          0 Oct 21 13:02 sys
drwxr-xr-x  2 root    root          0 Oct 21 12:49 tmp
drwxr-xr-x  4 root    root          0 Oct 21 12:48 usr
/ # dmesg | grep init
[  0.000000] random: crng init done
[  0.000000] Memory: 405360K/524288K available (12288K kernel code, 1618K rwddata, 479
[  0.000000]          .init : 0x(ptrval) - 0x(ptrval)  (2048 kB)
[  0.227325] devtmpfs: initialized
[  0.298684] pinctrl core: initialized pinctrl subsystem
[  0.720711] SCSI subsystem initialized
[  1.089029] Trying to unpack rootfs image as initramfs...
[  4.360057] Freeing initrd memory: 25136K
[  4.975861] SuperH (H)SCI(F) driver initialized
[  4.981538] msm_serial: driver initialized
[  4.983332] STMicroelectronics ASC driver initialized
[  4.989863] STM32 USART driver initialized
[  5.837387] Run /init as init process
/ # busybox --help | head -15
BusyBox v1.31.1 (2025-10-21 12:24:58 UTC) multi-call binary.
BusyBox is copyrighted by many authors between 1998-2015.
Licensed under GPLv2. See source distribution for detailed
copyright notices.

Usage: busybox [function [arguments]...]
or: busybox --list[-full]
or: busybox --show SCRIPT
or: busybox --install [-s] [DIR]
or: function [arguments]...
```

BusyBox is a multi-call binary that combines many common Unix

utilities into a single executable. Most people will create a [link](#) to busybox [for](#) each [function](#) they wish to use and BusyBox will act like whatever it was invoked as.

```
/ # poweroff
/ # umount: can't unmount /: Invalid argument
swapoff: can't open '/etc/fstab': No such file or directory
The system is going down NOW!
Sent SIGTERM to all processes
Sent SIGKILL to all processes
Requesting system poweroff
[ 91.086498] reboot: Power down
root@0d6b761d24fa:/workspace/repos/busybox#
```

Висновки

- Налаштовано середовище: cross-компілятор `arm-eabi`, змінні `ARCH=arm`, `CROSS_COMPILE`.
- Зібрано ядро: `v4.19`, `zImage`, `am335x-boneblack.dtb`, модулі.
- Сконфігуровано rootfs: встановлено BusyBox, додано необхідні бібліотеки (`libc.so`, `libm.so`, `libresolv.so`, `ld-linux.so`).
- Створено init-скрипт для запуску rootfs (`mdev`, монтування `/proc`, `/sys`).
- Тестування: rootfs завантажується, працює shell і базові драйвери пристроїв.