

Εργασία 1

ΕΝΣΩΜΑΤΩΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΡΑΓΜΑΤΙΚΟΥ ΧΡΟΝΟΥ

Απρίλιος 2018



του φοιτητή

Παπαδόπουλου Κωνσταντίνου
ΑΕΜ 8677

Εισαγωγή:

Λειτουργικό Σύστημα

Η εργασία, λόγω θεμάτων συμβατότητας του λογισμικού που χρησιμοποιείται, διεκπεραιώνεται με λιγότερα προβλήματα δουλεύοντας αρχικά με το λειτουργικό σύστημα *Ubuntu 16.04* (βέβαια αυτό δεν είναι η μόνη λύση).

Εγκατάσταση του QEMU

Εκτελούμε στο terminal των Linux την εντολή *sudo apt-get install qemu*.

Δημιουργία εικόνας OpenWrt Malta Mips

Αρχικά κατεβάζουμε το αρχείο *openwrt-zsun-zsun* από το Github (κάνουμε *unzip* το αρχείο αν χρειάζεται).

Στη συνέχεια τρέχουμε στο terminal:

```
cd ./ & sudo apt-get install qemu
```

make menuconfig – κατεβάζουμε όποια dependencies απαιτούνται και ρυθμίζουμε τις παραμέτρους που είναι συμβατές με το QEMU στο GUI που εμφανίζεται

```
make
```

Φόρτωση εικόνας στο QEMU

Η εκκίνηση του QEMU γίνεται με την εντολή

```
qemu-system-mips -kernel ~/path/to/openwrt-malta-be-vmlinux-initramfs.elf -redir  
tcp:2222::22 -nographic
```

Cross-compiling

Προσθέτω στο path τον ειδικό gcc compiler με την εντολή *source ./loadenv.sh*, όπου το *loadenv.sh* περιέχει τον κώδικα:

```
export STAGING_DIR=/path/to/openwrt/staging_dir/toolchain-mips_34kc_gcc-4.6-  
linaro_uClibc-0.9.33.2  
export PATH=$PATH:$STAGING_DIR/bin
```

Κάνω compile πχ με

```
mips-openwrt-linux-gcc ~/Desktop/hello_world.c -o hello (-static) ή  
mips-openwrt-linux-gcc ~/Desktop/hello_world_threads.c -o hello_threads -lpthread
```

Γενικά

Κάθε φορά με την εκκίνηση της εικόνας πρέπει να αλλάξουμε κωδικό με την εντολή `passwd`

Για να τρέξουμε τα threads πρέπει επίσης να ενσωματώσουμε την βιβλιοθήκη `libpthread.so.0`, αυτό γίνεται με
`scp -P 2222 /path/to/file.../libpthread.so.0 root@hostIP:/lib/`

Ομοίως αντιγράφω και κάνω επικόλληση το εκτελέσιμο που έχω μεταγλωττίσει με τον cross-compiler.

Ζητούμενα:

Γενικά

Στην συγκεκριμένη εργασία, δημιουργούμε μία σύνδεση `cp` ανάμεσα στις δύο συσκευές. Εφόσον έχει επιτευχθεί η σύνδεση, τα προγράμματα πρέπει να ανταλλάσσουν μηνύματα μεταξύ των δύο συσκευών. Συγκεκριμένα, η εφαρμογή `server` περιμένει την αποστολή μιας συμβολοσειράς (string) από τον `client`.

Μόλις αυτή σταλθεί και ληφθεί επιτυχώς, ο `server` εκτυπώνει το μήνυμα στην κονσόλα και περιμένει εκ νέου την αποστολή νέου μηνύματος.

Ειδικά

Το πρόγραμμα `server` πρέπει να

- κάνει χρήση των συναρτήσεων `socket`, `bind`, `listen`, `accept` και `connect`,
- χρησιμοποιεί τις αντίστοιχες βιβλιοθήκες, οι οποίες υπάρχουν και στο `toolchain` του OpenWRT,
- επιτρέπει την ταυτόχρονη εξυπηρέτηση πολλών συσκευών με την εφαρμογή `client`.

Για την ταυτόχρονη εξυπηρέτηση πολλών συσκευών, υλοποιούμε δύο εκδόσεις του `server`, μία με χρήση `processes` και μία με χρήση `threads`.

Client program

Η εφαρμογή `client` δέχεται δύο ορίσματα εισόδου: την IP της συσκευής όπου εκτελείται το πρόγραμμα `server` και την αντίστοιχη πόρτα.

Το πρόγραμμα client κάνει χρήση των συναρτήσεων `socket` και `connect` για την επικοινωνία με τον server και την αποστολή ενός μηνύματος. Ύστερα από την αποστολή, το πρόγραμμα τερματίζει.

Έχουν προστεθεί σχόλια στον κώδικα του προγράμματος που εξηγούν αναλυτικότερα τη λειτουργία του.

Server program

Η εφαρμογή πρέπει να δέχεται ένα όρισμα εισόδου με τον αριθμό της πόρτας στο μηχάνημα.

Η εφαρμογή server είναι υπεύθυνη να ανοίξει ένα `socket` σε μία καθορισμένη πόρτα της συσκευής όπου εκτελείται. Η εφαρμογή εκτελείται συνέχεια, αναμένοντας κάποιο χρήστη να συνδεθεί στη συγκεκριμένη πόρτα ώστε να ξεκινήσει η επικοινωνία.

Server program – Threads

Γίνεται χρήση του `pthread.h` header και βασικών συναρτήσεων για τη δημιουργία νημάτων, όπως η `pthread_create`, καθώς και μια συνάρτηση `connection_handler` που δημιουργήσαμε για να ελέγχουμε τη σύνδεση για κάθε client.

Server program – Processes

Χρησιμοποιούμε τη συνάρτηση `fork()` για τη δημιουργία διεργασιών που θα εξυπηρετούν τον κάθε client.

Έχουν προστεθεί σχόλια στον κώδικα του προγράμματος που εξηγούν αναλυτικότερα τη λειτουργία του.

*****Ο κώδικας της εργασίας βρίσκεται στο:**

<https://www.dropbox.com/sh/ltiszd04yb4ggey/AACyzTrfpqFYrwRtpZrTEDmda?dl=0>