

QEMU installation instructions

Για την εκτέλεση ενός προγράμματος γραμμένου σε assembly του επεξεργαστή ARM, έχουμε τρεις επιλογές:

1. Χρήση υλικού το οποίο χρησιμοποιεί επεξεργαστή ARM. Παράδειγμα μιας τέτοιας πλατφόρμας είναι το BeagleBoard που διαθέτει το εργαστήριο.
2. Χρήση ενός προσομοιωτή ενός συστήματος με επεξεργαστή ARM (virtualization). Ο προσομοιωτής θα τρέχει σε ένα σύστημα το οποίο θα χρησιμοποιεί διαφορετικό επεξεργαστή, κατά κανόνα έναν τύπου x86.
3. Χρήση ενός toolchain το οποίο θα μπορεί να κάνει compile προγράμματα γραμμένα σε assembly του ARM και να τα εκτελεί σε ένα περιβάλλον x86. Χαρακτηριστικό παράδειγμα τέτοιου toolchain είναι το GNU ARM (<http://www.gnuarm.com/>).

Στα πλαίσια αυτών των ασκήσεων θα γίνει αρχικά χρήση της 2^η μεθόδου. Ο προσομοιωτής που θα χρησιμοποιηθεί είναι ο QEMU (Quick Emulator - http://wiki.qemu.org/Main_Page) ο οποίος είναι open source και έχει την δυνατότητα προσομοίωσης μεγάλου αριθμού συστημάτων και επεξεργαστών.

Οδηγίες εγκατάστασης του προσομοιωτή σε περιβάλλον Linux

Υπό φυσιολογικές συνθήκες, το πρόγραμμα μπορεί να εγκατασταθεί κάνοντας χρήση του package manager της διανομής σας (πχ `apt-get install qemu-system-arm`, `pacman -S qemu` κτλ).

Σε περίπτωση που αυτό δεν λειτουργεί πρέπει να κάνετε manual εγκατάσταση του προγράμματος από τον πηγαίο κώδικα του. Ενδεικτικά τα βήματα για την εγκατάσταση του πλήρους πακέτου qemu-system (όχι μόνο του qemu-system-arm) είναι:

1. `sudo apt-get remove qemu-system` (σε περίπτωση που θέλετε να σβήσετε μια παλιά αποτυχημένη εγκατάσταση.)
2. `sudo apt-get install dh-autoreconf`
3. `git clone git://git.linaro.org/qemu/qemu-linaro.git`
4. `cd qemu-linaro`
5. `./configure --target-list=arm-softmmu,arm-linux-user` (Σε αυτό το βήμα ενδεχομένως να σας ζητήσει `pixman` και `dtc`. Θα βγάλει δύο επιλογές. Εκτελέστε την εντολή από την επιλογή 2)
6. `make`
7. `sudo make install`

Για περισσότερες πληροφορίες http://en.wikibooks.org/wiki/QEMU/Installing_QEMU .

Στο σημείο αυτό τα εκτελέσιμα του qemu βρίσκονται σε κάποιο subdirectory του `/usr/`. Σε περίπτωση που θέλετε να μπορούν να εκτελεστούν από οποιοδήποτε directory, πρέπει να βεβαιωθείτε ότι το directory των εκτελέσιμων του qemu συμπεριλαμβάνεται στο `$PATH`. Εφόσον έχει εγκατασταθεί ο qemu στο host μηχάνημα (το pc σας), πρέπει να δημιουργήσουμε ένα εικονικό guest μηχάνημα που θα τρέχει πάνω σε έναν εικονικό επεξεργαστή ARM. Δημιουργήστε έναν φάκελο που αντιστοιχεί σε αυτό το μηχάνημα και εκεί μέσα κατεβάστε τα εξής αρχεία:

- i. **debian_wheezy_armel_standard.qcow2**
(https://people.debian.org/~aurel32/qemu/armel/debian_wheezy_armel_standard.qcow2). Το αρχείο αυτό αντιστοιχεί στον σκληρό δίσκο του guest συστήματος σας. Εκεί έχει εγκατασταθεί ένα debian linux.
- ii. **initrd.img-3.2.0-4-versatile**
(<https://people.debian.org/~aurel32/qemu/armel/initrd.img-3.2.0-4-versatile>).
- iii. **vmlinuz-3.2.0-4-versatile** (<https://people.debian.org/~aurel32/qemu/armel/vmlinuz-3.2.0-4-versatile>).

Εάν όλα έχουν ολοκληρωθεί σωστά τότε μπορείτε να εκτελέσετε το guest μηχανήμα κάνοντας χρήση της εντολής:

```
sudo qemu-system-arm -M versatilepb -kernel vmlinuz-3.2.0-4-versatile -initrd initrd.img-3.2.0-4-versatile -hda debian_wheezy_armel_standard.qcow2 -append "root=/dev/sda1" -net nic -net user,hostfwd=tcp:127.0.0.1:22223-:22
```

Θα εμφανιστεί μια οθόνη η οποία λειτουργεί κανονικά ως τερματικό, επειδή όμως δεν σας επιτρέπει copy – paste και μεταφορά αρχείων, προτείνεται να δουλεύετε με ssh και sftp με τις εντολές να είναι:

i. `ssh -p 22223 root@localhost`

ii. `sftp -P 22223 root@localhost`

Αν θέλετε να χρησιμοποιήσετε κάποιον GUI-based code editor όπως το VS Code μπορείτε να χρησιμοποιήσετε την εντολή:

iii. `sshfs -p 22223 root@localhost:/remote/path /local/mount/point`

Τροποποιώντας τα αντίστοιχα directories, ώστε το remote directory στο οποίο δουλεύετε να γίνει mount σε κάποιο τοπικό που θα ανοίξετε με τον editor (το ssh extension του VS Code δε θα λειτουργήσει, γιατί η αρχιτεκτονική ARM δεν υποστηρίζεται από τον VS Code server που πρέπει να στηθεί remotely).

Ο default user είναι root με password root. Κατά την χρήση του vm θα χρειαστεί να εγκαταστήσετε όποια προγράμματα σας χρειάζονται. (πχ vim, gcc, build-essential).

Αφού εισέλθετε με επιτυχία στο vm θα χρειαστεί να αλλάξετε το περιεχόμενο του `/etc/apt/sources.list` του vm. Σε αυτό το σημείο να είστε προσεκτικοί προκειμένου να **MHN** αλλάξετε καταλάθος το αντίστοιχο αρχείο του host μηχανήματος, δηλαδή του υπολογιστή σας. Το αντίστοιχο αρχείο μπορείτε να τον βρείτε στο Έγγραφο → Lab_Material → lab3 → Extra Υλικό → sources.list.

Αφού αντιγράψετε τα παραπάνω εκτελέστε την εντολή:

```
apt-get update --allow-unauthenticated
```

Πιθανόν να σας εμφανίσει ορισμένα σφάλματα σχετικά με τα κλειδιά των distributions και των packages που γίνονται update, όπως φαίνεται στην παρακάτω εικόνα :

```
W: GPG error: http://archive.debian.org jessie/updates Release: The following signatures couldn't be verified
because the public key is not available: NO_PUBKEY 9D6D8F6BC857C906 NO_PUBKEY AA8E81B4331F7F50
```

Προκειμένου να επιλύσετε το παραπάνω πρόβλημα, αρκεί να εκτελέσετε την ακόλουθη εντολή για όλα τα κλειδιά που εμφανίζει το μήνυμα NO_PUBKEY <key_code>, όπου key_code ο αντίστοιχος κωδικός κλειδιού που εμφανίζει. Αν είναι περισσότεροι από ένας οι κωδικοί αυτοί, τότε η παρακάτω εντολή χρειάζεται να εκτελεστεί για κάθε ένα από τα κλειδιά αυτά:

```
apt-key adv --keyserver keyserver.ubuntu.com --recv-keys <key_code>
```

Μπορείτε να αγνοήσετε το παρακάτω warning:

```
W: GPG error: http://archive.debian.org jessie Release: The following signatures were invalid: KEYEXPIRED
1587841717
```

Στο link <https://people.debian.org/~aurel32/qemu/armel/> υπάρχουν και άλλες εκδόσεις του debian για τον qemu και μπορείτε να χρησιμοποιήσετε κάποια από αυτές εφόσον σας βολεύει.