
ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΥΠΟΛΟΓΙΣΤΙΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
http://www.cslab.ece.ntua.gr

# Εργαστήριο Λειτουργικών Συστημάτων

**7ο εξάμηνο, Ακαδημαϊκή περίοδος 2022–2023**

# Οδηγός προγραμματισμού σε περιβάλλον QEMU-KVM

Εργαστήριο Υπολογιστικών Συστημάτων Ε.Μ.Π.
os-lab@lists.cslab.ece.ntua.gr

**Οκτώβριος 2022**

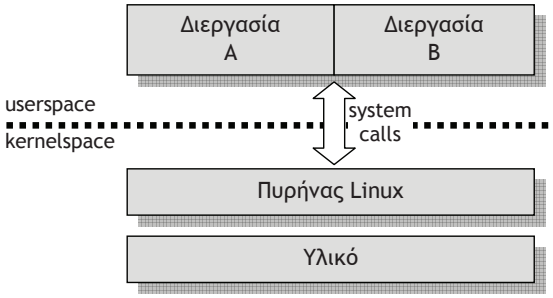
# Περιεχόμενα




*Επιμέλεια: Δ. Σιακαβάρας, Β. Καρακάσης, Γ. Κουβέλη, Ευ. Κούκης*

# 1 Εισαγωγή – Χώρος πυρήνα και χώρος χρήστη

Το Linux είναι ένα σύγχρονο λειτουργικό σύστημα που ακολουθεί την παράδοση του Unix. Υποστηρίζει πολλούς ταυτόχρονους χρήστες (είναι δηλαδή multi-user) και πολλές ταυτόχρονες διεργασίες (είναι multi-tasking), οι οποίες μοιράζονται το χρόνο της CPU. Η οργάνωση του λογισμικού ενός υπολογιστικού συστήματος όταν χρησιμοποιείται το Linux παρουσιάζεται στο παρακάτω σχήμα.




Σχήμα 1: Οργάνωση σε χώρους πυρήνα και χρήστη

Εφόσον πολλές διαφορετικές διεργασίες πολλών διαφορετικών χρηστών εκτελούνται στο ίδιο υπολογιστικό σύστημα, είναι ανάγκη να υπάρχουν συγκεκριμένοι μηχανισμοί απομόνωσης και ελέγχου πρόσβασης, ώστε διεργασίες που ανήκουν σε κάποιον χρήστη να μην έχουν πρόσβαση σε δεδομένα άλλων χρηστών, μία διεργασία που δυσλειτουργεί να μην επηρεάζει τη λειτουργία των υπολοίπων αλλά και να εξασφαλίζεται η δίκαιη κατανομή των πόρων του συστήματος ανάμεσα στις διεργασίες (χρόνος CPU, μνήμη, πρόσβαση σε συσκευές I/O).

Για να ικανοποιηθούν οι παραπάνω περιορισμοί, κάθε διεργασία εκτελείται απομονωμένη από τις υπόλοιπες, στον δικό της εικονικό χώρο διευθύνσεων αλλά και απομονωμένη από το hardware (δεν επιτρέπεται η εκτέλεση εντολών I/O). Για να μπορέσει να χρησιμοποιήσει τους πόρους του υπολογιστικού συστήματος, χρησιμοποιεί τις υπηρεσίες του πυρήνα του Linux μέσα από τον μηχανισμό των κλήσεων συστήματος. Ο πυρήνας είναι το μόνο πρόγραμμα που έχει τη δυνατότητα για απευθείας προσπέλαση στο υλικό. Ελέγχει κάθε κλήση συστήματος και αν επιτρέπεται η πρόσβαση στη συγκεκριμένη διεργασία, ικανοποιεί το αίτημά της και επιστρέφει τα αποτελέσματα.

Είναι φανερή λοιπόν η διάκριση ανάμεσα στο χώρο πυρήνα (kernelspace) και το χώρο χρήστη (userspace). Ο κώδικας των διεργασιών, που εκτελείται στο userspace, δεν επιτρέπεται να προσπελάσει απευθείας το υλικό και δεν έχει πρόσβαση στο χώρο μνήμης άλλων διεργασιών. Αντίθετα, ο κώδικας του πυρήνα, που εκτελείται στο kernelspace έχει ελεύθερη πρόσβαση στο υλικό και στη μνήμη του συ-

στήματος. Επιπλέον, πιθανή δυσλειτουργία του μπορεί να αποβεί μοιραία για τη λειτουργία του υπολογιστικού συστήματος.

# 2 Virtualization (Εικονικοποίηση)

Στις ασκήσεις του Εργαστηρίου Λειτουργικών Συστημάτων σας ζητείται η τροποποίηση του πυρήνα του Linux, ώστε να προστεθούν νέες λειτουργίες (κλήση συστήματος, υποστήριξη συσκευών). Παρόλο που για την διαδικασία ανάπτυξης (τροποποίηση κώδικα, μεταγλώττιση) δεν χρειάζονται αυξημένα δικαιώματα, μόνο ο διαχειριστής του συστήματος (χρήστης root) έχει το δικαίωμα τόσο να εγκαταστήσει ένα νέο πυρήνα όσο και να εισαγάγει κάποιο kernel module στον πυρήνα που ήδη τρέχει. Ο λόγος πίσω από αυτό τον περιορισμό είναι λίγο-πολύ προφανής: ο,τιδήποτε τρέχει στον χώρο πυρήνα, μπορεί να κάνει *τα πάντα* σε ένα σύστημα. Επομένως, εάν όλοι οι χρήστες είχαν αυτό το δικαίωμα, τότε η ασφάλεια και η σταθερότητα του συστήματος θα είχαν καταστρατηγηθεί. Ακόμη όμως και αν κανείς είναι ο διαχειριστής ενός συστήματος, δεν είναι σοφή ιδέα να δοκιμάζει τις αλλαγές που έκανε στον πυρήνα απευθείας σε ένα μηχάνημα, ειδικά εάν αυτό το χρησιμοποιούν και άλλοι χρήστες, γιατί και με αυτό τον τρόπο διακυβεύεται η ασφάλεια και η σταθερότητα του συστήματος. Η λύση στα παραπάνω ζητήματα που αφορούν στον προγραμματισμό του πυρήνα του Linux, είναι η χρήση εικονικοποίησης, η οποία αναλύεται στην συνέχεια.

Με τον όρο *εικονικοποίηση* (virtualization), αναφερόμαστε συνήθως σε μεθόδους και τεχνικές για τη δημιουργία «εικονικών» αντικειμένων, αντίστοιχων των «φυσικών», τα οποία μπορούν να χρησιμοποιηθούν με ισοδύναμο τρόπο. Στην περίπτωσή μας αναφερόμαστε σε εικονικοποίηση υλικού (full virtualization), όπου ο τελικός στόχος είναι η κατασκευή ολόκληρων *εικονικών μηχανών* (virtual machines): προσομοιώνουμε τη λειτουργία ενός ολόκληρου υπολογιστικού συστήματος σε λογισμικό. Μέσα στην εικονική μηχανή που προκύπτει, ο χρήστης μπορεί να τρέχει το Λειτουργικό Σύστημα και εφαρμογές με τον ίδιο ακριβώς τρόπο όπως και στο πραγματικό μηχάνημα. Τα πλεονεκτήματα είναι πολλά, ανάμεσα στα άλλα: (i) ευελιξία – ταυτόχρονη εκτέλεση πολλών διαφορετικών ΛΣ στο ίδιο φυσικό σύστημα, (ii) φορητότητα – προγράμματα εκτελούνται σε σε διαφορετικό υλικό από αυτό για το ποίο έχουν γραφτεί/μεταγλωττιστεί, (iii) ασφάλεια – κάθε εικονική μηχανή εκτελείται απομονωμένα από όλες τις υπόλοιπες και από το φυσικό υλικό, (iv) αποδοτικότερη χρήση των φυσικών πόρων – μεγάλο πλήθος εικονικών μηχανών μπορεί να μοιράζεται περιορισμένους φυσικούς πόρους, (v) βελτιωμένη αξιοπιστία – η εικονική μηχανή μπορεί να μεταφερθεί σε διαφορετικό φυσικό σύστημα, ώστε να συνεχίσει τη λειτουργία χωρίς διακοπή, ξεπερνώντας προβλήματα του υλικού, π.χ. μια επερχόμενη διακοπή ρεύματος.

Η προσομοίωση της λειτουργίας ενός υπολογιστικού συστήματος εξ ολοκλήρου σε λογισμικό είναι μια υπολογιστικά απαιτητική διαδικασία. Η ανάγκη για μετάφραση από το σύνολο εντολών της εικονικής μηχανής ("*guest*") στο σύνολο εντολών του φυσικούς μηχανήματος ("*host*") εισάγει σημαντική επιβάρυνση στην ταχύτητα εκτέλεσης της εικονικής μηχανής.

Για το λόγο αυτό, συνήθως δεν γίνεται μετάφραση κάθε εντολής, αλλά η εικονική μηχανή εκτελείται σε φυσικό σύστημα με το *ίδιο* σύνολο εντολών. Έτσι, τα πράγματα απλοποιούνται σημαντικά: το λογισμικό δεν χρειάζεται να μεταφράζει κάθε εντολή, αλλά χρειάζεται να παρεμβαίνει για να συντηρήσει την ψευδαίσθηση μόνο όταν η εικονική μηχανή χρειάζεται να εκτελέσει μια *προνομιούχο* εντολή.

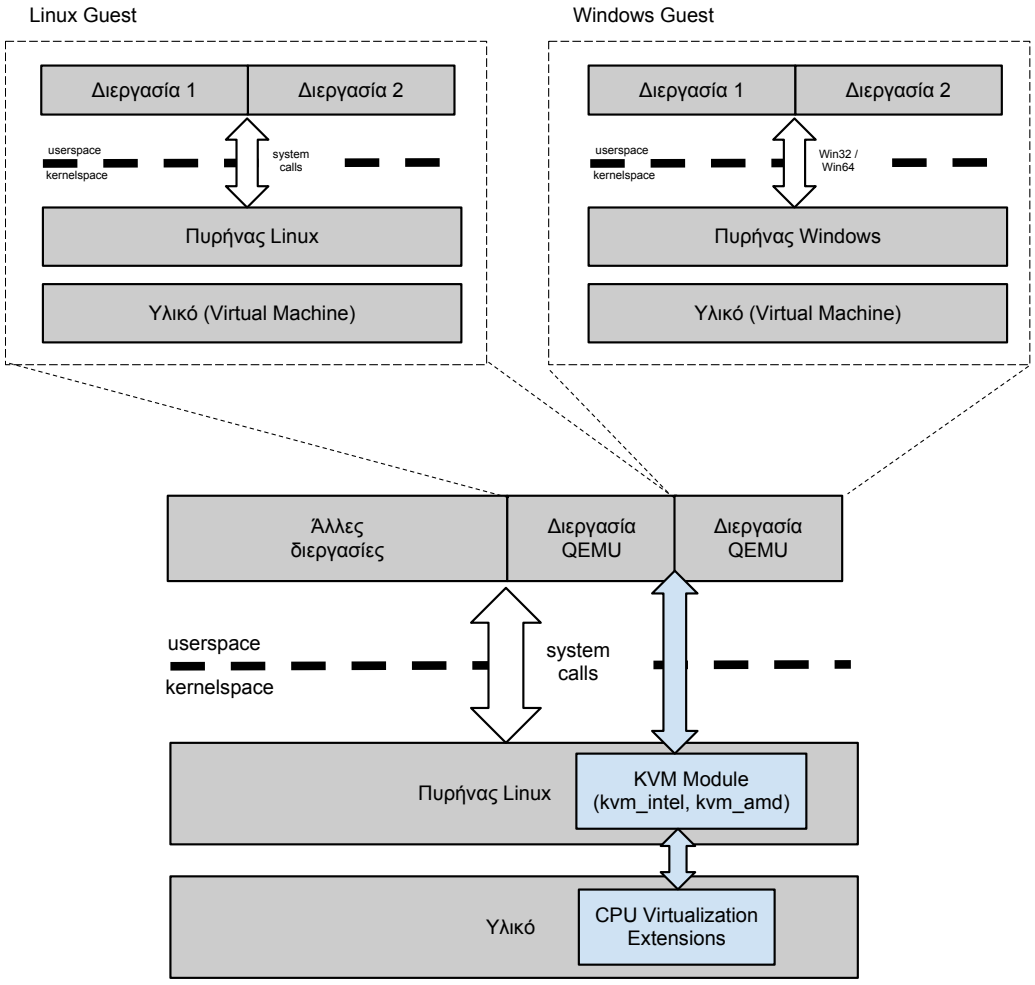
Επιπλέον, η αυξημένη ζήτηση για αποδοτική εκτέλεση εικονικών μηχανών, οδήγησαν στην ανάπτυξη ειδικών επεκτάσεων υλικού (π.χ. ειδικές εντολές στο σύνολο εντολών των σύγχρονων επεξεργαστών), έτσι ώστε το *υλικό* να υποστηρίζει αποδοτικότερα λειτουργίες εικονικοποίησης. Οι νεότεροι επεξεργαστές x86, τόσο από την AMD όσο και από την Intel, διαθέτουν τέτοιες επεκτάσεις, επιτρέποντας την εκτέλεση εικονικών μηχανών με απόδοση που προσεγγίζει αυτή του πραγματικού μηχανήματος.

Παραδείγματα λογισμικού virtualization αποτελούν τα VMWare ESXi, Microsoft Hyper-V, Oracle VM VirtualBox, Xen και QEMU. Στο εργαστήριο θα χρησιμοποιήσουμε για τη δημιουργία εικονικών μηχανών το QEMU-KVM, δηλαδή το λογισμικό QEMU σε συνδυασμό με το υποσύστημα KVM του πυρήνα του Linux.

Το KVM (Kernel-based Virtual Machine) είναι ένα υποσύστημα εικονικοποίησης που έχει ενσωματωθεί στον πηγαίο κώδικα του πυρήνα του Linux από την έκδοση 2.6.20, το οποίο προσφέρει έναν ενιαίο τρόπο για χρήση των επεκτάσεων εικονικοποίησης του επεξεργαστή. Αποτελείται από δύο modules, το `kvm.ko` και το `kvm_intel.ko` ή το `kvm_amd.ko` ανάλογα με το μοντέλο του επεξεργαστή του φυσικού μηχανήματος.

Το QEMU (Quick EMUlator) είναι μία πλατφόρμα για τη δημιουργία και διαχείριση εικονικών μηχανών. Το QEMU εκτελείται σε επίπεδο χρήστη και υποστηρίζει εικονικοποίηση είτε αμιγώς μέσω λογισμικού, είτε εικονικοποίηση μέσω υλικού. Στην πρώτη περίπτωση δεν απαιτεί ειδική υποστήριξη από το υλικό αλλά είναι πολύ αργό, ειδικά αν το σύνολο εντολών της εικονικής μηχανής διαφέρει από το σύνολο εντολών του φυσικού μηχανήματος. Στη δεύτερη περίπτωση, χρησιμοποιεί το KVM που είδαμε προηγουμένως ώστε να έχει πρόσβαση στις επεκτάσεις εικονικοποίησης του επεξεργαστή. Τότε, η εικονική μηχανή είναι απαραίτητο να είναι της ίδιας αρχιτεκτονικής με το μηχάνημα στο οποίο εκτελείται.

Εκτελώντας μία διεργασία `qemu` στο χώρο χρήστη του πραγματικού συστήματος, θα μπορείτε να έχετε τη δική σας εικονική μηχανή, στην οποία θα έχετε πλήρες δι-




Σχήμα 2: Δημιουργία εικονικών μηχανών με χρήση των εργαλείων QEMU και KVM.

καίωμα διαχειριστή και δυνατότητα για οποιαδήποτε μεταβολή στο σύστημα. Ταυτόχρονα όμως θα είστε περιορισμένοι στο χώρο χρήστη του πραγματικού συστήματος. Το πώς ακριβώς γίνεται αυτό, αναλύεται στη συνέχεια. Στις επόμενες ενότητες θα αναφερόμαστε με τον όρο *host* στο πραγματικό σας μηχάνημα στο οποίο εκτελείται το QEMU, ενώ με τον όρο *guest* θα εννοούμε την εικονική μηχανή.

# 3 Προετοιμασία host μηχανήματος

Για την εκτέλεση του QEMU-KVM όπως περιγράφεται στον παρόντα οδηγό, θα χρειαστείτε πρόσβαση σε κάποιο μηχάνημα το οποίο διαθέτει επεξεργαστή με δυ-

νατότητες εικονικοποίησης και τρέχει κάποια διανομή Linux. Μπορείτε να χρησιμοποιήσετε ένα δικό σας μηχάνημα, ή, αν θέλετε να ακολουθήσετε ακριβώς τις οδηγίες του παρόντος οδηγού, να δημιουργήσετε μία εικονική μηχανή στην υπηρεσία ~okeanos (http://okeanos.grnet.gr, βλ. ενότητα 5).

Αν αισθάνεστε περιπετειώδεις, μπορείτε να χρησιμοποιήσετε και μηχάνημα Windows 11 22H2 με Windows Subsystem for Linux έκδοσης 2 (WSL 2) ως host.

### Προσοχή!

Το WSL2 απέκτησε υποστήριξη για nested virtualization πολύ πρόσφατα. Για WSL2 έχουμε ελέγξει αυτόν τον οδηγό μόνο σε Windows 11 22H2, με τελευταία έκδοση του WSL2 από το Microsoft store και επεξεργαστή Intel. Ενδεικτικά:

```
C:\>wsl.exe --version
WSL version: 0.70.4.0
Kernel version: 5.15.68.1
[...]
Windows version: 10.0.22621.755
```

Μπορείτε να αποκτήσετε πλήρη γραφική πρόσβαση στο host μηχάνημά σας, αλλά και στο guest VM αργότερα, χρησιμοποιώντας το σύστημα X2Go. Ο server του X2Go είναι ήδη εγκατεστημένος στις υποστηριζόμενες εικόνες μηχανών του ~okeanos, αλλά και στο root filesystem που σας δίνεται για την εικονική μηχανή ανάπτυξης. Η σύνδεση X2Go πραγματοποιείται πάνω από κρυπτογραφημένο κανάλι ssh. Ακολουθήστε τις οδηγίες σύνδεσης όπως περιγράφονται στην ενότητα 5.

### Προσοχή!

Στα ακόλουθα υποθέτουμε μια διανομή Debian ή βασισμένη στο Debian. Οι οδηγίες έχουν δοκιμαστεί πλήρως σε διανομή Ubuntu 22.04.1 LTS 64-bit. Θεωρούμε ότι υπάρχει λογαριασμός απλού χρήστη `user` κι εκτελείτε τις εντολές ως `user`. Η εικόνα της εικονικής μηχανής που σας δίνεται για τον guest (`cslab_rootfs`) είναι μηχάνημα Debian bullseye x86-64, οπότε απαιτείται κι ο host να είναι εγκατάσταση x86-64 για να εκτελέσει εικονική μηχανή αυτού του τύπου με KVM.

Στα επόμενα παραδείγματα υποθέτουμε ότι "`$`" είναι το πρωτεύον prompt (`PS1`) του χρήστη, "`#`" του `root`, και "`>`" το δευτερεύον (`PS2`), που δείχνει ότι μια εντολή συνεχίζεται και στην επόμενη γραμμή.

**Προσοχή!**

Εφεξής θα χρησιμοποιούμε το prompt "`#`" για να δείξουμε εντολές που πρέπει να εκτελέσετε ως root και το prompt "`$`" για να δείξουμε εντολές που πρέπει να εκτελέσετε ως απλός χρήστης.

Σας δείχνουμε τις εντολές όπως εμφανίζονται στο τερματικό σας, και είναι έτοιμες για επικόλληση κι εκτέλεση. Στην περίπτωση αυτή πρέπει να παραλείπετε το κάθε prompt.

Αρχικά, βεβαιωθείτε ότι έχετε θέσει το password του `root`, και μπορείτε να αποκτήσετε root shell, δίνοντας `su -`:

```
$ sudo su -
... enter user's password here, if requested ...
# passwd
... change password for root, pick a STRONG password ...
# exit
$ su -
... enter password for root again ...
# ... you are now the superuser, note the different prompt.
# exit
$ ... back to unprivileged user shell.
```

Στη συνέχεια, εγκαταστήστε στον host πακέτα από τα repositories της διανομής που θα χρειαστείτε παρακάτω για να χτίσετε το QEMU από τον πηγαίο κώδικά του:

```
# apt-get install build-essential vim bzip2 gzip xz-utils
# apt-get install socat git wget libglib2.0-dev
# apt-get install libfdt-dev libpixman-1-dev zlib1g-dev
$ apt-get install ninja-build libcap-ng-dev libattr1-dev
```

## 3.1 Έλεγχος για CPU virtualization extensions

Για να επιβεβαιώσετε ότι ο υπολογιστής σας παρέχει επεκτάσεις εικονικοποίησης ανοίξτε ένα τερματικό και δώστε την παρακάτω εντολή:

```
$ egrep '(vmx|svm)' /proc/cpuinfo
```

Αν η έξοδος της εντολής περιλαμβάνει τις λέξεις `vmx` (Intel) ή `svm` (AMD), τότε ο επεξεργαστής έχει δυνατότητες εικονικοποίησης και υποστηρίζει τη λειτουργία KVM του πυρήνα. Για παράδειγμα, η έξοδος της παραπάνω εντολής για επεξεργαστή Intel δίνει έξοδο παρόμοια με την παρακάτω:

```
flags       : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr
   pge mca cmov pat pse36 clflush dts acpi mmx fxsr sse sse2 ss
   ht tm pbe nx lm constant_tsc arch_perfmon pebs bts aperfmperf
    pni dtes64 monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr
   pdcm sse4_1 xsave lahf_lm dtherm tpr_shadow vnmi flexpriority
```

## 3.2 Φόρτωση KVM modules

Αφού έχετε βεβαιωθεί ότι ο επεξεργαστής σας παρέχει τις επεκτάσεις εικονικοποίησης, πρέπει να ελέγξετε αν ο πυρήνας Linux που τρέχετε περιλαμβάνει τα modules του KVM. Αν η εντολή

```
$ lsmod | grep kvm
```

δεν δώσει κάποια έξοδο σημαίνει ότι τα modules είτε δεν περιλαμβάνονται στον πυρήνα σας είτε ότι απλά δεν έχουν φορτωθεί. Στην πρώτη περίπτωση θα πρέπει να κάνετε χειροκίνητη εγκατάσταση του KVM αλλά κάτι τέτοιο δεν είναι στα πλαίσια αυτού του μαθήματος και παραλείπεται. Στην δεύτερη περίπτωση αρκεί να δώσετε ως root (εντολή su)

```
# modprobe kvm_amd
```

(ή `kvm_intel`, ανάλογα με την αρχιτεκτονική) η οποία θα φορτώσει αυτόματα και το module `kvm`, απ' όπου εξαρτώνται τα `kvm_{amd, intel}`.

Τέλος, με

```
$ lsmod | grep kvm
```

πρέπει να βλέπετε τα δύο φορτωμένα modules του πυρήνα.

**Σημείωση**

Πολύ σπάνια, ο πυρήνας έχει ενσωματωμένη υποστήριξη για KVM, χωρίς modules. Στο WSL2 αυτό ισχύει, οπότε προχωρήστε απευθείας στην 3.3.

```
$ dmesg|grep -i kvm
[    0.140543] KVM: vmx: using Hyper-V Enlightened VMCS
[    0.289889] kvm: already loaded the other module
$ cat /sys/devices/virtual/misc/kvm/dev
10:232
```

Σε περίπτωση που χρειάστηκε να φορτώσετε τα modules με το χέρι μπορείτε να ρυθμίσετε τη διανομή σας να τα φορτώνει κατά την εκκίνηση του υπολογιστή. Για να γίνει αυτό προσθέστε στο αρχείο `/etc/modules` τις γραμμές `kvm` και `kvm_intel` ή `kvm_amd`. Για παράδειγμα:

```
# echo -e "kvm\nkvm_amd" >> /etc/modules
```

## 3.3 Πρόσβαση στο /dev/kvm

Σε αυτό το σημείο αφού έχετε φορτώσει τα modules του KVM στον πυρήνα σας, πρέπει να βλέπετε το ειδικό αρχείο `/dev/kvm`, που το προσπελάζει όποιο πρόγραμμα θέλει να χρησιμοποιήσει τις επεκτάσεις εικονικοποίησης, για να καλέσει τις υπηρεσίες του KVM. Για να σηκώσετε μία εικονική μηχανή με χρήση QEMU-KVM πρέπει να έχετε δικαιώματα εγγραφής-ανάγνωσης σε αυτό το αρχείο. Αρχικά, το αρχείο αυτό δημιουργείται με δικαιώματα ανάγνωσης-εγγραφής (`rw`) για το χρήστη `root` και το group `root`. Επειδή θέλουμε για λόγους ασφάλειας κι απομόνωσης οι εικονικές μηχανές να εκτελούνται με δικαιώματα χρηστών εκτός του `root`, θα δημιουργήσετε ένα group `kvm`, αν δεν υπάρχει, στο οποίο θα προσθέσετε τον χρήστη με τον οποίο εσείς χρησιμοποιείτε το σύστημα (π.χ. `user`), και θα ρυθμίσετε το αρχείο `/dev/kvm` να δημιουργείται με δικαιώματα στο group αυτό. Αυτό γίνεται ως εξής:

```
$ su -
# groupadd kvm
# usermod -aG kvm user
# echo KERNEL==\"kvm\", GROUP=\"kvm\" >> \
> /etc/udev/rules.d/40-permissions.rules
# modprobe -r kvm_amd
# modprobe -r kvm
# modprobe kvm_amd
# exit
$ groups
user adm cdrom sudo dip plugdev lpadmin sambashare
```

**Σημείωση**

Το `udev` δεν τρέχει σε διανομές κάτω από WSL2, οπότε οι κανόνες στο `/etc/udev/rules.d` δεν θα έχουν αποτέλεσμα. Στην περίπτωση αυτή, μπορείτε να ρυθμίσετε το `/etc/wsl.conf` ως εξής:

```
$ cat /etc/wsl.conf
```

```
[boot]
command="chown root:kvm /dev/kvm && chmod g+rw /dev/kvm"
```

Επειδή μόλις προσθέσατε τον χρήστη `user` στο group `kvm`, ο τρέχων φλοιός του χρήστη εκτελείται χωρίς δικαίωμα στο group `kvm`. Για να αποκτήσει δικαίωμα στο group kvm, ο απλούστερος τρόπος είναι ο χρήστης να αποσυνδεθεί (logout) και να συνδεθεί εκ νέου. Τότε, θα ανήκει στο group, και θα έχει δικαίωμα εγγραφής στο `/dev/kvm`:

```
$ groups
user adm cdrom sudo dip plugdev lpadmin sambashare kvm
$ ls -l /dev/kvm
crw-rw----+ 1 root kvm 10, 232 Mar  9 19:53 /dev/kvm
```

## 3.4 Εγκατάσταση του QEMU

Το QEMU είναι διαθέσιμο ως έτοιμο πακέτο από τα repositories των περισσότερων διανομών αλλά σε αυτήν την ενότητα θα δείξουμε πως μπορείτε να το χτίσετε κατευθείαν από τον πηγαίο κώδικά του, καθώς θα σας φανεί χρήσιμο και για επόμενες εργαστηριακές ασκήσεις. Για το σκοπό αυτό, κατεβάζετε τον πηγαίο κώδικα (www.qemu.org), τον αποσυμπιέζετε και τον μεταγλωττίζετε.

Στα επόμενα θα χτίσετε την έκδοση QEMU 7.1.0, που αυτή τη στιγμή είναι η πιο πρόσφατη έκδοσή του.

Κατεβάστε και αποσυμπιέστε τον κώδικα του QEMU:

```
$ wget https://download.qemu.org/qemu-7.1.0.tar.xz
$ tar xvJf qemu-7.1.0.tar.xz
```

Ρυθμίστε τις παραμέτρους της μεταγλώττισης: Ενεργοποιήστε υποστήριξη για KVM και για προσάρτηση αρχείων του host με χρήση του συστήματος αρχείων "9p", και επιλέξτε να εγκαταστήσετε το QEMU σε κατάλογο όπου ο απλός χρήστης έχει πρόσβαση για εγγραφή, π.χ. τον κατάλογο `~/opt/qemu`:

```
$ cd qemu-7.1.0
$ ./configure --prefix=$HOME/opt/qemu --enable-kvm \
> --target-list=x86_64-softmmu --enable-virtfs
```

Και τέλος μεταγλωττίστε και εγκαταστήστε το QEMU:

```
$ make -j2
$ make install
```

Το χτίσιμο του QEMU θα πάρει λίγη ώρα, η παράμετρος `-j` ζητά από το `make` να τρέχει δύο διεργασίες ταυτόχρονα. Μπορείτε να δοκιμάσετε μικρότερες ή μεγαλύτερες τιμές, αναλόγως με τη μνήμη και το πλήθος cores που έχει ο host σας.

Αν όλα πάνε καλά, θα έχετε το QEMU και όλα τα συνοδευτικά του αρχεία εγκατεστημένα κάτω από το `/home/user/opt/qemu` που επιλέξατε ως στόχο της εγκατάστασης.

Σε αυτό το σημείο το μηχάνημα το οποίο θα χρησιμοποιήσετε ως host για τις εικονικές μηχανές σας είναι έτοιμο και έχετε χτίσει κι εγκαταστήσει το QEMU.

# 4 Χρήση του QEMU - εικονική μηχανή ανάπτυξης

## 4.1 Εκκίνηση του QEMU

Για τη δική σας ευκολία, σας παρέχουμε δύο αρχεία για να να ξεκινήσετε τη λειτουργία της εικονικής σας μηχανής για *ανάπτυξη* του κώδικά σας: (i) `utopia.sh`, το οποίο εκκινεί την εικονική μηχανή, και (ii) `utopia.config`, το οποία περιέχει τις απαραίτητες ρυθμίσεις ώστε να μπορέσει να εκκινήσει σωστά η εικονική μηχανή. Το script `utopia.sh` διαβάζει τις παραμέτρους στο `utopia.config` και ξεκινά την εικονική μηχανή. Εάν συμβεί κάποιο λάθος, θα το επισημάνει και θα προτείνει κάποια πιθανή λύση. Αρχικά, όλες οι παράμετροι στο `utopia.config` είναι σε σχόλια, οπότε θα πρέπει να τις ενεργοποιήσετε για να τρέξει επιτυχώς το `utopia.sh`. Το `utopia.config` περιέχει στην ουσία μία σειρά από μεταβλητές περιβάλλοντος που χρησιμοποιεί το `utopia.sh` κατά τη λειτουργία του. Οι πιο σημαντικές από τις μεταβλητές αυτές είναι:

`QEMU_BUILD_DIR` Αυτός είναι ο κατάλογος στον οποίο έχει γίνει η εγκατάσταση του QEMU. Αν ο κατάλογος υπάρχει, το `utopia.sh` θα ψάξει εκεί για το εκτελέσιμο `bin/qemu-system-x86_64`, το οποίο και θα χρησιμοποιήσει για να εκκινήσει την εικονική μηχανή.

`QCOW2_BACKING_FILE` Αυτό είναι το αρχικό Image που θα χρησιμοποιήσει η εικονική μηχανή για να τρέξει ένα πλήρες σύστημα Linux. Στο filesystem που σας παρέχεται στο εργαστήριο, υπάρχει εγκατεστημένο Debian 11 (bullseye). Τίποτε δεν γράφεται σε αυτό το αρχείο. Όλες οι αλλαγές σας καταλήγουν στο αρχείο `$QCOW2_PRIVATE_FILE`, δηλ. στο `private.qcow2`. Σβήνοντάς το, μπορείτε να επαναφέρετε τη μηχανή στην αρχική της κατάσταση.

`SHARED_FS_DIR` Το QEMU θα κάνει διαθέσιμο τον κατάλογο `SHARED_FS_DIR` του host στον guest, ώστε το Linux να μπορεί να προσπελάσει [“mount”] αυτόν

τον κατάλογο, μέσω της τεχνολογίας virtio-9p. Αυτό σημαίνει ότι ο κατάλογος αυτός θα είναι κοινός ανάμεσα στον host και τον guest, δεν θα επηρεάζεται από τη διαγραφή του `private.qcow2`, και μπορείτε να αφήνετε για ευκολία εκεί τον κώδικά σας. Περισσότερα για αυτή τη λειτουργικότητα στην ενότητα 4.4.

Για να ξεκινήσετε, κατεβάστε το root filesystem της εικονικής σας μηχανής από το site του μαθήματος, έστω `cslab_rootfs_20221102.raw.gz`, αποσυμπιέστε το, με χρήση του εργαλείου `gunzip`, και μετακινήστε το σε φάκελο της επιλογής σας, έστω `~/utopia`:

```
$ mkdir ~/utopia && cd ~/utopia
$ wget http://www.cslab.ece.ntua.gr/courses/compsyslab/files/\
> 2022-23/cslab_rootfs_20221102.raw.gz
$ gunzip -k cslab_rootfs_20221102.raw.gz
```

Αφού πλέον έχετε ρυθμίσει σωστά την εικονική σας μηχανή, μπορείτε να την εκκινήσετε εκτελώντας το `utopia.sh`.

Η εικονική μηχανή θα ξεκινήσει, και το `utopia.sh` θα εκτυπώσει διαγνωστικά μηνύματα για το πώς μπορείτε να συνδεθείτε σε αυτή:

```
$ ./utopia.sh

 *** Reading configuration

UTOPIA_CONFIG=./utopia.config
[...]
 *** Starting your Virtual Machine ...

To connect with X2Go: See below for SSH settings
To connect with SSH: ssh -p 22223 root@localhost
To connect with vncviewer: vncviewer localhost:0
```

Συγχαρητήρια! Ξεκινήσατε μόλις την εικονική σας μηχανή, τώρα πρέπει να συνδεθείτε σε αυτή, για να τη μπορέσετε να τη χρησιμοποιήσετε.

## 4.2 Σύνδεση με το QEMU

Η εικονική μηχανή που σας δίνεται έχει ρυθμισμένους δύο χρήστες, τους `user` και `root`, με passwords `user` και `root` αντίστοιχα.

Υπάρχουν οι εξής τρόποι σύνδεσης στην εικονική σας μηχανή:

- Με VNC viewer, από το φυσικό μηχάνημα, ή το εξωτερικό VM:
  `vncviewer localhost:0`.
  Είναι αργός τρόπος, αλλά διευκολύνει πολύ το debugging, γιατί μπορείτε να δείτε την εκκίνηση της μηχανής από την πρώτη στιγμή. *Υπόδειξη:* Πατήστε F8 στο παράθυρο του VNC viewer για να μπορέσετε να στείλετε τον συνδυασμό πλήκτρων Ctrl-Alt-Del στην εικονική μηχανή.

- Με ssh από τον host / εξωτερική εικονική μηχανή, στο port 22223, με χρήση της εντολής `ssh -p 22223 root@localhost`. Η εσωτερική εικονική μηχανή ακούει στο TCP port 22223 του localhost (εξωτερικού VM).

- Με χρήση του `x2goclient` από την εξωτερική μηχανή στην εσωτερική μηχανή, πάλι με σύνδεση ssh στο port 22223 του localhost.

Μπορείτε να αλλάξετε τη μεταβλητή `UTOPIA_SSH_INTERFACE` στο `utopia.config`, ώστε η εικονική μηχανή ανάπτυξης να επιτρέπει συνδέσεις από οπουδήποτε. Με το τρόπο αυτό θα μπορείτε να συνδέεστε απευθείας προς αυτή, από οποιοδήποτε εξωτερικό μηχάνημα, στη θύρα 22223 του εξωτερικού μηχανήματος, π.χ. `snf-xxxx.vm.okeanos.grnet.gr`.

**Προσοχή!**

Βεβαιωθείτε ότι έχετε θέσει *ισχυρά* passwords και για τους *δύο* χρήστες, `user` και `root`, πριν ενεργοποιήσετε μια τέτοια επιλογή.

## 4.3 Τερματισμός του QEMU

Ακριβώς όπως και για ένα πραγματικό μηχάνημα, η λειτουργία της εικονικής μηχανής πρέπει να τερματίζεται σωστά. Ως `root`, χρησιμοποιήστε την εντολή `halt` ή `poweroff` για να «σβήσετε» την εικονική μηχανή, ή την εντολή `reboot` για να κάνετε επανεκκίνηση. Εάν για κάποιο λόγο η εικονική μηχανή δεν αποκρίνεται, π.χ. ο τροποποιημένος πυρήνας σας εκτελέσει κάποια εσφαλμένη λειτουργία, μπορείτε να τραβήξετε στην ουσία την εικονική μηχανή από την πρίζα, δίνοντας Ctrl-C στο τερματικό που εκτελείται το `utopia.sh`. Αν δεν υπάρξει αποτέλεσμα, εκτελέστε στο host `killall qemu-system-x86_64` ή `killall -9 qemu-system-x86_64`.

## 4.4 Διαμοιρασμός αρχείων με το QEMU

Σας συνιστούμε να κρατάτε τα αρχεία της άσκησης έξω από την εικονική μηχανή ενώ δουλεύετε, ώστε να είναι ασφαλή ακόμη και σε περίπτωση που το σύστημα αρχείων της εικονικής μηχανής πάθει κάποια βλάβη (π.χ., λόγω προγραμματιστικού

σφάλματος κατά την εκτέλεση του πυρήνα σας ή «βίαιου» shutdown της εικονικής μηχανής), ή σε περίπτωση που επιλέξετε να επαναφέρετε την εικονική μηχανή σβήνοντας το αρχείο `private.qcow2`.

Για το σκοπό αυτό, το `utopia.sh` ρυθμίζει τον κατάλογο `$SHARED_FS_DIR` ως μοιραζόμενο κατάλογο ανάμεσα στον host και τον guest [βλ. ενότητα 4.1]. Έχουμε ήδη επίσης ρυθμίσει τον guest [δείτε το αρχείο `/etc/fstab`] ώστε ο κατάλογος `./shared` του host να είναι προσβάσιμος ως `/home/user/shared` στον guest.

Οπότε, το μόνο που χρειάζεται να κάνετε είναι να αφήνετε τον κώδικά σας στον κατάλογο `./shared`, και θα είναι προσβάσιμος τόσο από τον host, όσο και από τον guest.

Σας προτείνουμε να κρατάτε ολόκληρο τον πηγαίο κώδικα της άσκησης στον κατάλογο `./shared` στον host, δηλαδή στον κατάλογο `/home/user/shared` στον guest, ώστε ακόμα και αν επαναφέρετε την εικονική μηχανή στην αρχική της κατάσταση διαγράφοντας το τοπικό COW αρχείο (`private.qcow2`), τα αρχεία αυτά να μην επηρεαστούν.

## 4.5 Μεταγλώττιση modules στον guest

Το μόνο που μένει για να μπορείτε να δουλέψετε με τον κώδικα της 2ης άσκησης είναι να εγκαταστήσετε τα απαραίτητα αρχεία μέσα στον guest για να μπορείτε να μεταγλωττίσετε modules για τον πυρήνα του Linux.

Για το σκοπό αυτό, θα εγκαταστήσετε το πακέτο `linux-headers` του Debian που αντιστοιχεί στον πυρήνα που τρέχετε αυτή τη στιγμή μέσα στην εικονική μηχανή, και για τον οποίο θα μεταγλωττίσετε το module σας.

Για παράδειγμα:

```
# uname -r
5.10.0-19-amd64
# apt-get update
Hit:1 http://deb.debian.org/debian bullseye InRelease
[...]
# apt-cache show linux-headers-$(uname -r)
Package: linux-headers-5.10.0-19-amd64
Source: linux
[...]
# apt-get install linux-headers-$(uname -r)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
```

```
linux-headers-5.10.0-19-amd64 is already the newest version (5.10.149-2).
```

Το πακέτο `linux-headers` εγκαθιστά τα απολύτως απαραίτητα κάτω από τον κατάλογο `/lib/modules/` ώστε να έχετε κατάλληλο `$KERNELDIR` για τη μεταγλώττιση δικών σας kernel modules.

# 5 Συχνές ερωτήσεις

Κατά την διαδικασία υλοποίησης της άσκησης μπορούν προκύψουν διάφορα τεχνικά προβλήματα, τα πιο συχνά από τα οποία αναφέρονται στην συνέχεια.

***Ερώτηση:***
*Τι είναι ο ∼okeanos και πως μπορώ να δημιουργήσω μία εικονική μηχανή εκεί;*

***Απάντηση:***
Πληροφορίες σχετικά με την υπηρεσία ∼okeanos και τη δημιουργία και χρήση VM εκεί μπορείτε να βρείτε στο site της υπηρεσίας https://okeanos.grnet.gr/home. Μπορείτε να βρείτε απαντήσεις σε συχνές ερωτήσεις για την υπηρεσία στο https://okeanos.grnet.gr/support/faq, καθώς και οδηγό για τη δημιουργία εικονικής μηχανής:
https://okeanos.grnet.gr/support/user-guide/cyclades-how-to-create-a-vm

***Ερώτηση:***
*Πώς μπορώ να συνδεθώ με γραφικό περιβάλλον στην εικονική μου μηχανή;*

***Απάντηση:***
Ακολουθήστε τις οδηγίες σύνδεσης με χρήση του συστήματος X2Go, τις οποίες μπορείτε να βρείτε στον οδηγό χρήσης του ∼okeanos:
https://okeanos.grnet.gr/support/user-guide/cyclades-how-do-i-connect-to-a-vm

***Ερώτηση:***
*Πώς μπορώ να δημιουργήσω το δικό μου root filesystem;*

***Απάντηση:***
Αυτή είναι μια αρκετά επίπονη διαδικασία. Αν θέλετε να πειραματιστείτε, μπορείτε να βρείτε οδηγίες στα http://wiki.colar.net/creating_a_qemu_image_and_installing_debian_in_it, http://wiki.debian.org/QEMU.

# 6 Περισσότερες πληροφορίες

Περισσότερα για το QEMU, το KVM, το σχεδιασμό, την εγκατάσταση και τη χρήση τους μπορείτε να μάθετε στις σελίδες

- http://www.linux-kvm.org
- http://www.qemu.org

Περισσότερα στοιχεία για την υπηρεσία ~okeanos και το open source λογισμικό Synnefo που την υποστηρίζει μπορείτε να βρείτε στις σελίδες

- http://okeanos.grnet.gr
- http://www.synnefo.org