

# ΜΥΕ029 – Προσομοίωση και Μοντελοποίηση Υπολογιστικών Συστημάτων

---

## 1η Εργαστηριακή Άσκηση

### Ανάλυση της απόδοσης του συστήματος αρχείων EXT4 σε περιβάλλον εικονικοποίησης

(Παράδοση έως 20/4/2022)

---

## 1. Εισαγωγή

Στην παρούσα εργαστηριακή άσκηση θα ασχοληθείτε με την **ανάλυση της απόδοσης του ημερολογιακού (journaled) συστήματος αρχείων EXT4** στο περιβάλλον εικονικοποίησης VMWare Workstation Player. Πιο συγκεκριμένα, **θα συγκρίνετε την απόδοση διαφόρων μεθόδων λειτουργίας του ημερολογίου που διατηρεί το ext4**. Η εικονική μηχανή που θα εξετάσετε προορίζεται να εκτελείται σε ένα περιβάλλον εξυπηρέτησης δυναμικών ιστοσελίδων.

Το σύστημα αρχείων είναι λογισμικό επιπέδου συστήματος, το οποίο παρέχει μια πλούσια διεπαφή για την αλληλεπίδραση του χρήστη με μια συσκευή αποθήκευσης. **Ένα είδος διεπαφής που υποστηρίζεται από μια πληθώρα συστημάτων αρχείων περιλαμβάνει ως βασικές δομές τα αρχεία και τους φακέλους**. Έτσι, **ο χρήστης ενός συστήματος οργανώνει τα δεδομένα του χρησιμοποιώντας αυτές τις δομές χρησιμοποιώντας υψηλού επιπέδου λειτουργίες**. Το σύστημα αρχείων **αναλαμβάνει να μετατρέψει αυτές τις λειτουργίες σε λειτουργίες επιπέδου μπλοκ ώστε να αποκτήσει πρόσβαση σε μια συσκευή αποθήκευσης**. Το σύστημα αρχείων EXT4 είναι η τελευταία έκδοση μιας οικογένειας συστημάτων αρχείων που σχεδιάστηκαν εξ' ολοκλήρου για το Λειτουργικό Σύστημα Linux.

## 2. Λειτουργία ημερολογίου στο EXT4

Το σύστημα αρχείων **EXT4 είναι ένα ημερολογιακό σύστημα αρχείων**. Πιο συγκεκριμένα, διατηρεί ένα **αρχείο καταγραφής που ονομάζεται ημερολόγιο**, στο οποίο **καταγράφει τις λειτουργίες που πρόκειται να εκτελέσει στο δίσκο πριν την πραγματοποίησή τους**. Το ημερολόγιο είναι ιδιαίτερα **χρήσιμο για την ανάκαμψη του συστήματος από σφάλματα σε περιπτώσεις βίαιης κατάρρευσης** του συστήματος. Ειδικότερα, **όταν συμβαίνει μια κατάρρευση, το EXT4 εξετάζει το ημερολόγιο ώστε να επαναλάβει λειτουργίες που δε πρόλαβαν να ολοκληρωθούν σωστά**.

Το **ημερολόγιο του EXT4 υποστηρίζει τρεις μεθόδους λειτουργίας: α) journaled, β) ordered, και γ) writeback**. Όταν χρησιμοποιείται η μέθοδος λειτουργίας **journaled**, το **EXT4 πριν προχωρήσει με την εκτέλεση μιας λειτουργίας, καταγράφει στο ημερολόγιο κάθε λειτουργία, καθώς και τα δεδομένα και**

τα μεταδεδομένα που αντιστοιχούν σε αυτή. Αυτή η μέθοδος είναι η ασφαλέστερη, καθώς σε περίπτωση κατάρρευσης του συστήματος οι λειτουργίες που διακόπηκαν μπορούν να επαναληφθούν. Όταν χρησιμοποιείται η μέθοδος λειτουργίας ordered, το EXT4 γράφει τα δεδομένα μιας λειτουργίας στην τελική τους θέση στο δίσκο και στη συνέχεια καταγράφει τη λειτουργία και τα μεταδεδομένα στο ημερολόγιο. Αυτή η μέθοδος εγγυάται τη σωστή διάταξη των δεδομένων σε περίπτωση κατάρρευσης. Τέλος, στη μέθοδο λειτουργίας writeback, το ext4 γράφει τα δεδομένα στην τελική τους θέση αφού πρώτα καταγράψει στο ημερολόγιο τις λειτουργίες και τα μεταδεδομένα. Αυτή η μέθοδος δε προσφέρει εγγυήσεις.

### 3. Filebench

Το filebench είναι ένα ευέλικτο πρόγραμμα αναφοράς για την σύγκριση της απόδοσης συστημάτων αρχείων και αποθηκευτικών συστημάτων. Το βασικό χαρακτηριστικό του είναι η συμπερίληψη ενός αριθμού από τεχνητά φορτία εργασίας (Πίνακας 1) που αντιπροσωπεύουν με μεγάλη ακρίβεια συγκεκριμένες κατηγορίες εφαρμογών. Επιπλέον, υποστηρίζει μια γλώσσα υψηλού επιπέδου που ονομάζεται Workload Model Language (WML), η οποία μπορεί να χρησιμοποιηθεί για τη δημιουργία νέων, προσαρμοσμένων φορτίων εργασίας.

Το filebench εκτελεί ένα πείραμα σε δύο στάδια. Στο πρώτο στάδιο προ-δημιουργεί ένα τεχνητό σύνολο δεδομένων που αποτελείται από μια δομή φακέλων και αρχεία δεδομένων. Στο δεύτερο στάδιο, εκτελεί το φορτίο εργασίας. Κάποια φορτία εργασίας δεν απαιτούν την προ-δημιουργία ενός συνόλου δεδομένων (π.χ., δημιουργούν αρχεία και φακέλους στο δεύτερο στάδιο), οπότε το πρώτο στάδιο παραλείπεται.

**Πίνακας 1: Βασικά τεχνητά φορτία εργασίας που υποστηρίζει το Filebench**

Φορτίο Εργασίας	Περιγραφή
singlestreamread	Ακολουθιακή ανάγνωση ενός μεγάλου (5 GB) αρχείου. Χρησιμοποιούνται λειτουργίες ανάγνωσης μεγέθους 1MB.
singlestreamwrite	Ακολουθιακή εγγραφή ενός αρχείου. Χρησιμοποιούνται λειτουργίες εγγραφής μεγέθους 1MB.
randomread	Τυχαία ανάγνωση ενός μεγάλου (5 GB) αρχείου. Χρησιμοποιούνται λειτουργίες ανάγνωσης μεγέθους 8 KB.
randomwrite	Τυχαία εγγραφή ενός μεγάλου (5 GB) αρχείου. Χρησιμοποιούνται λειτουργίες εγγραφής μεγέθους 8 KB. Το αρχείο προ-δημιουργείται στην αρχή της εκτέλεσης του πειράματος.
createfiles	Δημιουργεί 50,000 αρχεία σε ένα δέντρο φακέλων. Το μέγεθος

	των αρχείων επιλέγεται με βάση κατανομή γάμμα (μέσο 16 KB). Ολοκληρώνεται μετά τη δημιουργία των αρχείων.
webserver	Προσομοιώνει έναν από εξυπηρετητή ιστοσελίδων. Δημιουργεί μια ακολουθία αιτήσεων άνοιγμα-ανάγνωση-κλείσιμο σε διάφορα αρχεία καθώς και μια λειτουργία εγγραφής δεδομένων στο τέλος ενός αρχείου καταγραφής.
fileserver	Προσομοιώνει έναν εξυπηρετητή αρχείων. Το φορτίο εργασίας περιλαμβάνει μια ακολουθία από δημιουργίες, διαγραφές, εγγραφές στο τέλος ενός αρχείου, αναγνώσεις, εγγραφές, και λειτουργίες μεταδεδομένων σε ένα δέντρο φακέλων.
varmail	Προσομοιώνει έναν εξυπηρετητή ηλεκτρονικού ταχυδρομείου. Το φορτίο εργασίας περιλαμβάνει μια ακολουθία από λειτουργίες δημιουργίας-εγγραφής στο τέλος-συγχρονισμού, ακολουθούμενη από μια ακολουθία λειτουργιών ανάγνωσης και διαγραφής σε ένα φάκελο.
oltp	Προσομοιώνει μια βάση δεδομένων. Το φορτίο εργασίας εκτελεί λειτουργίες συστήματος αρχείων βασισμένο στο μοντέλο E/E Oracle 9i. Δοκιμάζει την απόδοση μικρών τυχαίων αναγνώσεων και εγγραφών και είναι σημαντικά ευαίσθητο στην καθυστέρηση μεσαίου μεγέθους εγγραφών (128+ KB) σε ένα αρχείο καταγραφής.
webproxy	Προσομοιώνει έναν Web proxy. Περιλαμβάνει μια μίξη λειτουργιών δημιουργίας-εγγραφής-κλεισίματος, ανοίγματος-ανάγνωσης-κλεισίματος, και διαγραφής.
videoserver	Προσομοιώνει έναν εξυπηρετητή βίντεο. Περιλαμβάνει δύο συλλογές. Η μια συλλογή περιλαμβάνει βίντεο που παρακολουθούν οι χρήστες και η δεύτερη περιλαμβάνει βίντεο που είναι διαθέσιμα αλλά δε παρακολουθούνται. Ένα νήμα γράφει νέα βίντεο στη δεύτερη συλλογή, ενώ \$nthreads εξυπηρετούν βίντεο από την πρώτη συλλογή.

Τα φορτία εργασίας που υποστηρίζει το filebench είναι παραμετροποιήσιμα από τον χρήστη. Ο πίνακας 2 συνοψίζει κάποιες από τις βασικές παραμέτρους. Στο τέλος κάθε φορτίου εργασίας προστίθεται η εντολή:

**run XX**

όπου XX το χρονικό διάστημα σε δευτερόλεπτα που θα εκτελεστεί το πείραμα. Αν αυτή η γραμμή λείπει, τότε πρέπει να την προσθέσετε.

**Πίνακας 2: Βασικές παράμετροι φορτίου εργασίας.**

Παράμετρος	Περιγραφή
\$dir	Φάκελος όπου θα εκτελεστεί το πείραμα. Για τις ανάγκες της μελέτης σας ορίστε σε αυτήν την παράμετρο την τιμή /mnt.
\$filesize	Μέγεθος αρχείου
\$filesize ( παράμετρος mean:)	Το μέγεθος αρχείου ακολουθεί μια κατανομή με μέσο όρο την τιμή που αναγράφεται δίπλα στο mean.
\$iosize	Το μέγεθος κάθε ξεχωριστής λειτουργίας ανάγνωσης ή εγγραφής.
\$nfiles	Ο αριθμός των αρχείων που θα δημιουργηθούν.
\$nlogfiles	Αριθμός αρχείων καταγραφής (π.χ., βάσης δεδομένων)
\$nshadows	Αριθμός διεργασιών που εκτελούν λειτουργίες ανάγνωσης.
\$nbwriters	Αριθμός διεργασιών που εκτελούν λειτουργίες εγγραφής.
\$meandirwidth	Το μέσο πλήθος αρχείων που περιλαμβάνει ένας φάκελος.
\$nthreads	Ο αριθμός των νημάτων.
\$meanappendsize	Μέσο μέγεθος εγγραφής στο τέλος ενός αρχείου.
\$runtime	Χρόνος σε λεπτά εκτέλεσης του πειράματος. Θα χρειαστεί να επιλέξετε με προσοχή αυτή την παράμετρο έτσι ώστε από την μια πλευρά ο όγκος του φορτίου εργασίας να είναι ικανοποιητικός και να δοκιμάζει την απόδοση του δίσκου και από την άλλη ο όγκος του φορτίου εργασίας να μη ξεπερνάει το μέγεθος του δίσκου τον οποίο δοκιμάζει το πείραμα.
run XX (π.χ., run 60)	

Πριν εκτελέσετε το filebench αντιγράψτε τα φορτία εργασίας που θα χρησιμοποιήσετε από τον φάκελο /root/filebench-1.5-alpha3/workloads/ στον φάκελο /root/workloads. Στη συνέχεια, προσαρμόστε τις τιμές των παραμέτρων στις ανάγκες σας και εκτελέστε το filebench με την ακόλουθη εντολή (π.χ., για το φορτίο εργασίας fileserver.f):

```
$ filebench -f /root/workloads/fileserver.f
```

Υπάρχει η περίπτωση το φορτίο εργασίας που εκτελείται να δημιουργεί όγκο δεδομένων μεγαλύτερο από το μέγεθος του δίσκου στο οποίο εκτελείται το EXT4. Σε αυτή την περίπτωση, το filebench θα

τερματιστεί βίαια και θα σας ενημερώσει ότι δεν υπάρχουν διαθέσιμα αποτελέσματα. Για να αποφύγετε αυτή την περίπτωση θα πρέπει να μειώσετε είτε τον αριθμό των νημάτων, είτε το μέγεθος των αρχείων, είτε το χρόνο εκτέλεσης του πειράματος, είτε ένα συνδυασμό αυτών των παραμέτρων.

Όταν η εκτέλεση του filebench ολοκληρώνεται σωστά, τυπώνονται τα αποτελέσματα που συγκέντρωσε το πρόγραμμα.

## 4. Περιβάλλον

### 4.1.1 Εικονική Μηχανή

Το περιβάλλον στο οποίο θα εκτελέσετε την πειραματική μελέτη αποτελείται από μια **εικονική μηχανή VMWare που έχει εγκατεστημένο το Λειτουργικό Σύστημα Debian 11**. Για να εκτελέσετε την εικονική μηχανή, θα χρειαστείτε το σύστημα εικονικοποίησης **VMWare Workstation Player**, το οποίο μπορείτε να το κατεβάσετε δωρεάν για περιβάλλον Windows ή Linux από τον σύνδεσμο:

<https://www.vmware.com/products/workstation-player/workstation-player-evaluation.html>

Αφού κατεβάσετε και εγκαταστήσετε το Player, μπορείτε να κατεβάσετε την εικονική μηχανή από τον παρακάτω σύνδεσμο, αφού πρώτα συνδεθείτε στον ιδρυματικό σας λογαριασμό:

<https://drive.google.com/file/d/1yfdNzQMLYyb1Cu033BeNZWf1AMGTET8e/view?usp=sharing>

Η εικονική μηχανή βρίσκεται στο συμπιεσμένο αρχείο mye029vm.zip. Θα χρειαστεί να αποσυμπίεσετε το αρχείο αυτό σε κάποιο φάκελο στον υπολογιστή σας. Για να προσθέσετε την εικονική μηχανή στο Player, ανοίξτε με διπλό κλικ το αρχείο MYE029VM.vmx.

Η εικονική μηχανή είναι ρυθμισμένη ώστε να χρησιμοποιεί **2 GB μνήμη RAM**, **έναν πυρήνα επεξεργαστή**, και **έναν εικονικό δίσκο με μέγιστο μέγεθος 12 GB**. Το δίκτυο της εικονικής μηχανής έχει ρυθμιστεί με την επιλογή NAT, ώστε να διαμοιράζεται τη σύνδεση δικτύου του συστήματος φιλοξενίας. Για να μεταφέρετε αρχεία από και προς την εικονική μηχανή μπορείτε να συνδέσετε μια συσκευή flash επιλέγοντας το μενού “Virtual Machine” και στη συνέχεια επιλέγοντας τη συσκευή από την κατηγορία “Removable Devices”. Εναλλακτικά, μπορείτε να χρησιμοποιήσετε τον ssh server που είναι εγκατεστημένος στη συσκευή για να συνδεθείτε στην εικονική μηχανή με την εντολή ssh από το σύστημα φιλοξενίας (ή το πρόγραμμα PuTTY), ή να μεταφέρετε αρχεία με την εντολή scp, ή με κάποιο πρόγραμμα μεταφοράς αρχείων, όπως το Filezilla.

Μπορείτε να συνδεθείτε στην εικονική μηχανή είτε ως απλός χρήστης χρησιμοποιώντας ως όνομα χρήστη και κωδικό πρόσβασης τη συμβολοσειρά “user”, είτε ως διαχειριστής χρησιμοποιώντας ως όνομα χρήστη και κωδικό πρόσβασης τη συμβολοσειρά “root”.

Στην εικονική μηχανή είναι εγκατεστημένο το εργαλείο filebench, καθώς και μια πληθώρα από εργαλεία παρακολούθησης πόρων. Επιπλέον είναι εγκατεστημένος ο επεξεργαστής κειμένου Visual Studio Code, καθώς και άλλοι γνωστοί επεξεργαστές κειμένου, όπως ο emacs, ο nano και το vi.

Μπορείτε να εγκαταστήσετε ένα νέο πρόγραμμα, ανοίγοντας ένα τερματικό ως διαχειριστής και δίνοντας τις εντολές: `apt update ; apt install package-name`. Για να αναζητήσετε ένα πακέτο χρησιμοποιήστε την εντολή `apt search package-name`.

### 4.1.2 Διαχείριση αποθήκευσης

Ο εικονικός δίσκος της εικονικής μηχανής περιλαμβάνει τη διαμέριση `/dev/sda3` που θα χρησιμοποιήσετε για την πειραματική σας μελέτη. Αυτή η διαμέριση έχει μέγεθος 5.1 GB.

Στο λογαριασμό του χρήστη διαχειριστή θα βρείτε το φάκελο `scripts`. Εκεί, βρίσκονται δύο σενάρια κελύφους (`scripts`) τα οποία μπορείτε να χρησιμοποιήσετε για να δημιουργήσετε και να προσαρτήσετε ή να αποπροσαρτήσετε το σύστημα αρχείων EXT4 με λειτουργίες ημερολογίου `journal` (`ext4-journal`), `ordered` (`ext4-ordered`), και `writeback` (`ext4-writeback`). Για παράδειγμα, για να ξεκινήσετε το σύστημα αρχείων EXT4 σε λειτουργία ημερολογίου `ordered` συνδεθείτε ως χρήστης `root` και εκτελέστε την παρακάτω εντολή:

```
$ /root/scripts/start-disk.sh ext4-ordered
```

Αυτό το σενάριο θα δημιουργήσει ένα νέο σύστημα αρχείων EXT4 στη διαμέριση `/dev/sda3` και θα το προσαρτήσει στο φάκελο `/mnt`. Για να αποπροσαρτήσετε το EXT4 εκτελέστε την εντολή:

```
$ /root/scripts/stop-disk.sh
```

### 4.1.3 Διαχείριση κρυφής μνήμης

Η διαχείριση της κρυφής μνήμης μπορεί να γίνει αλλάζοντας τις παραμέτρους στα κατάλληλα αρχεία κάτω από τον φάκελο `/proc/sys/vm`. Ο πίνακας 3 συνοψίζει κάποιες βασικές παραμέτρους.

Πίνακας 3: Βασικές παράμετροι κρυφής μνήμης συστήματος αρχείων.	
<code>/proc/sys/vm/drop_caches</code>	Γράφοντας την τιμή 3 σε αυτό το αρχείο το σύστημα καθαρίζει τις κρυφές μνήμες δεδομένων και μεταδεδομένων του συστήματος αρχείων. Π.χ., <code>\$ echo 3 &gt; /proc/sys/vm/drop_caches</code>
<code>/proc/sys/vm/dirty_expire_centicecs</code>	Καθορίζει μετά από πόσο διάστημα τα δεδομένα στην κρυφή μνήμη πρέπει να γραφούν στο δίσκο. Οι μονάδες μέτρησης είναι σε csec και η προκαθορισμένη τιμή είναι 30 sec.
<code>/proc/sys/vm/dirty_writeback_centicecs</code>	Ο πυρήνας του συστήματος χρησιμοποιεί έναν αριθμό νημάτων τα οποία μεταφέρουν δεδομένα από την κρυφή μνήμη στο δίσκο. Αυτή η παράμετρος καθορίζει το χρονικό διάστημα (σε csec) μεταξύ δύο διαδοχικών εκτελέσεων αυτών των νημάτων. Η προκαθορισμένη τιμή είναι 5 sec.

## 5. Βήματα

Σκοπός της παρούσας εργαστηριακής άσκησης είναι να μελετήσετε την απόδοση του συστήματος αρχείων EXT4 και πιο συγκεκριμένα της λειτουργίας ημερολογίου. Γι' αυτό το σκοπό προτείνεται να ακολουθήσετε τα ακόλουθα βήματα:

1. Από τα φορτία εργασίας που προσφέρει το filebench, τα οποία μπορείτε να βρείτε στο φάκελο `/root/filebench-1.5-alpha3/workloads/` επιλέξτε κάποια που θεωρείτε ενδεικτικά για τη μελέτη σας και αντιγράψτε τα στο φάκελο `/root/workloads/`. Για να κάνετε αυτή την επιλογή βασιστείτε στην υπόθεση ότι στο περιβάλλον που μελετάτε θα εκτελούνται εφαρμογές εξυπηρέτησης δυναμικών ιστοσελίδων.
2. Ρυθμίστε την κρυφή μνήμη του συστήματος έτσι ώστε να διατηρεί τα δεδομένα το πολύ για 5 sec πριν ξεκινήσει την εγγραφή τους στο δίσκο. Επιπλέον, ρυθμίστε το χρόνο μεταξύ δύο διαδοχικών εκτελέσεων των νημάτων πυρήνα που κάνουν τη μεταφορά δεδομένων στο 1 sec.
3. Από τις παραμέτρους που παρέχει κάθε φορτίο εργασίας επιλέξτε ως παράγοντες το μέσο μέγεθος αρχείων και τον αριθμό νημάτων ή διεργασιών. Επιλέξτε γι' αυτούς τους παράγοντες κατάλληλες τιμές/επίπεδα. Για τις υπόλοιπες παραμέτρους επιλέξτε κατάλληλες σταθερές τιμές (π.χ., χρόνος εκτέλεσης πειράματος, αριθμός νημάτων). Οι επιλογές που θα κάνετε είναι σημαντικό να οδηγούν στην εξέταση της απόδοσης του δίσκου. Για παράδειγμα, αν ο χρόνος εκτέλεσης του πειράματος είναι μικρότερος από το χρόνο που καθορίζει το αρχείο `dirty_expire centiseconds`, τότε τα δεδομένα μπορεί να μην προλάβουν να γραφούν στο δίσκο, με αποτέλεσμα να αξιολογείτε την απόδοση μόνο της μνήμης. Τέλος, ρυθμίστε κατάλληλα το φάκελο που θα γίνει η δοκιμή ώστε να περιλαμβάνει το `"/mnt"` στο οποίο προσαρτάται το EXT4. Σε αυτό το σημείο, είναι χρήσιμο να πραγματοποιήσετε δοκιμαστικές εκτελέσεις του filebench με κάθε ένα από τα φορτία εργασίας που επιλέξατε, έτσι ώστε διασφαλίσετε ότι οι επιλογές που κάνατε οδηγούν στη συμπεριφορά που επιδιώκετε.
4. Για το φορτίο εργασίας που θεωρείτε πιο απαιτητικό με βάση τις δοκιμές που κάνατε στο βήμα 3, εκτελέστε το filebench για κάθε μία από τις τρεις μεθόδους λειτουργίας του ημερολογίου στο EXT4 και ελέγξτε τη χρησιμοποίηση του επεξεργαστή με την `mpstat`, της μνήμης με την `vmstat`, και του δίσκου με την `iostat`. Χρησιμοποιήστε τον παρατηρητή `perf` ώστε να βρείτε και να καταγράψετε: α) τα τμήματα λογισμικού που χρησιμοποιούν περισσότερο τον επεξεργαστή (`perf record/top/report`), β) διάφορα στατιστικά μετρητών απόδοσης, όπως εναλλαγές κατάστασης, και χρήση κρυφών μνημών (`perf stat`).
5. Εκτελέστε το σύνολο των πειραμάτων σας. Είναι σημαντικό να εκτελέσετε έναν αριθμό από επαναλήψεις και να παρουσιάσετε τη μέση ρυθμικό απόδοση καθώς και το 95% διάστημα εμπιστοσύνης. Μπορείτε να ακολουθήσετε μια στρατηγική που αποφασίζει δυναμικά τον αριθμό των επαναλήψεων. Ξεκινήστε με 3 επαναλήψεις. Όσο το 95% διάστημα εμπιστοσύνης δεν έχει ικανοποιητικό μέγεθος, συνεχίστε τις επαναλήψεις. Καθώς εκτελούνται τα πειράματα μετρήστε τη χρησιμοποίηση του επεξεργαστή, του δίσκου, και της μνήμης, χρησιμοποιώντας τα



προγράμματα `mpstat`, `vmstat`, και `iostat`. Επειδή ο αριθμός των εκτελέσεων μπορεί να είναι μεγάλος, θα σας βοηθήσει ιδιαίτερα αν αυτοματοποιήσετε τη διαδικασία εκτέλεσης των πειραμάτων γράφοντας σενάρια κελύφους Bash ή χρησιμοποιώντας κάποια γλώσσα προγραμματισμού (π.χ., Python). Για το σκοπό αυτό μπορείτε να βασιστείτε στον ψευδοκώδικα που θα βρείτε στον Πίνακα 4 του Παραρτήματος.

6. Δημιουργήστε γραφικές παραστάσεις για να παρουσιάσετε τα αποτελέσματά σας. Θα πρέπει να παρουσιάσετε τόσο τα αποτελέσματα που πήρατε από το `filebench`, όσο και διάφορα αποτελέσματα που πήρατε χρησιμοποιώντας τα εργαλεία παρατήρησης της χρησιμοποίησης των πόρων.
7. Γράψτε μια αναφορά στην οποία θα συμπεριλάβετε τα εξής:
  - α) Μια περιγραφή του συστήματος φιλοξενίας, καθώς και της εικονικής μηχανής όπου τρέχετε τα πειράματά σας.
  - β) Τις γραφικές παραστάσεις που δημιουργήσατε.
  - γ) Κείμενο που να αναλύει τα αποτελέσματα της μελέτης σας.
  - δ) Τα συμπεράσματα της μελέτης.

## 6. Παράδοση

Μπορείτε να παραδώσετε την εργασία είτε ατομικά, είτε σε ομάδες των δύο. Είναι σημαντικό να υποβάλετε το αρχείο `report.pdf` που περιλαμβάνει την πλήρη αναφορά, καθώς και τα scripts που δημιουργήσατε για να αυτοματοποιήσετε τη διαδικασία αξιολόγησης.

Μη ξεχάσετε να συμπεριλάβετε στην πρώτη σελίδα του αρχείου `report.pdf` τα ονόματά και τους αριθμούς μητρώου και των δύο μελών της ομάδας.

Υποβάλετε τη λύση σας με την εντολή:

```
$ turnin lab1_21@mye029 report.pdf file1 file2...
```

Οδηγίες για τη χρήση του `turnin` μπορείτε να βρείτε στον ακόλουθο σύνδεσμο:

<https://support.cs.uoi.gr/doku.php?id=cse:%CE%B5%CF%81%CE%B3%CE%B1%CF%83%CF%84%CE%AE%CF%81%CE%B9%CE%B1:turnin>



## Παράρτημα

**Πίνακας 4: Ψευδοκώδικας για την αυτοματοποίηση της εκτέλεσης των πειραμάτων.**

```
runs = 0;                                # counts the experiment repetitions
alpha = 0.05                             # for 95% confidence interval
need_more_runs = true;                   # flag used to stop experiment
ext4_journal = ...;                     # the ext4 journal type
while need_more_runs == true do
    # Start a clean filesystem
    /root/scripts/start-disk.sh ext4-${ext4_journal}

    # Start resource usage monitors and write output to files
    mpstat [parameters] > /tmp/mpstat.out
    iostat [parameters] > /tmp/iostat.out
    vmstat [parameters] > /tmp/vmstat.out

    # clean filesystem caches
    echo 3 > /proc/sys/vm/drop_caches

    # Run filebench and write output to a file
    filebench -f /roots/workloads/<workload name>.f > /tmp/results

    # Stop resource usage monitors
    pkill -2 mpstat
    pkill -2 iostat
    pkill -2 vmstat

    # Update number of runs
    runs += 1

    # update statistical metrics
    cpu_usr_avg = ...
    cpu_sys_avg = ...
    cpu_iowait_avg = ...
    disk_read_tput_avg = ...
    disk_write_tput_avg = ...
    ...

    # update result metrics
    total_tput_avg = ...
    read_tput_avg = ...
    write_tput_avg = ...
    ...

    # Choose one metric and calculate:
```

```
# 1. Standard deviation
# 2. 95% confidence interval

# Check if more runs are needed:
if 95_confidence_interval / average_of_chosen_metric < alpha
then
    need_more_runs = false
endif

# Stop filesystem
/root/scripts/stop-disk.sh
done

print-statistics-and-results
```