



Δημοκρίτειο Πανεπιστήμιο Θράκης
Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων



«

»

Δρ Κων/νος Δεμερτζής

Ορεστιάδα, Φεβρουάριος 2019



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

Κατάλογος Περιεχομένων

1	ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ.....	6
1.1	ΠΥΡΗΝΑΣ.....	7
1.2	ΚΕΛΥΦΟΣ.....	7
1.3	ΥΠΗΡΕΣΙΕΣ	8
1.4	ΣΥΓΧΡΟΝΑ ΛΕΙΤΟΥΡΓΙΚΑ ΣΥΣΤΗΜΑΤΑ	10
1.5	VIRTUALIZATION	12
2	ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ.....	16
2.1	ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ.....	16
2.2	ΥΠΗΡΕΣΙΕΣ ΔΙΑΔΙΚΤΥΟΥ	18
2.3	Ο ΜΕΛΛΟΝΤΙΚΟΣ ΙΣΤΟΣ	22
3	ΕΙΣΑΓΩΓΗ ΣΤΑ ΔΙΚΤΥΑ ΥΠΟΛΟΓΙΣΤΩΝ	28
3.1	ΚΑΤΗΓΟΡΙΕΣ ΔΙΚΤΥΩΝ.....	28
3.2	ΑΡΧΙΤΕΚΤΟΝΙΚΗ ΔΙΚΤΥΩΝ.....	32
3.3	IPV6 ΤΕΧΝΟΛΟΓΙΑ	34
3.4	CLOUD COMPUTING	38
4	ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΔΙΑΔΙΚΤΥΟΥ.....	41
4.1	ΑΡΧΕΣ ΨΗΦΙΑΚΗΣ ΑΣΦΑΛΕΙΑΣ	41
4.2	ΤΥΠΟΙ ΕΠΙΘΕΣΕΩΝ	42
4.3	ΠΟΛΙΤΙΚΕΣ ΑΣΦΑΛΕΙΑΣ	43
4.4	ΣΧΕΔΙΟ ΑΝΑΚΑΜΨΗΣ (DRP)	45
4.5	ΑΝΩΝΥΜΙΑ ΣΤΟ ΔΙΑΔΙΚΤΥΟ	46
5	ΠΕΡΙΒΑΛΛΟΝΤΙΚΗ ΠΛΗΡΟΦΟΡΙΚΗ	52
5.1	ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ (ARTIFICIAL INTELLIGENCE).....	52
5.2	ΜΗΧΑΝΙΚΗ ΜΑΘΗΣΗ (MACHINE LEARNING).....	53
5.3	ΑΣΑΦΗΣ ΛΟΓΙΚΗ (FUZZY LOGIC)	56
5.4	ΤΕΧΝΗΤΑ ΝΕΥΡΩΝΙΚΑ ΔΙΚΤΥΑ (ARTIFICIAL NEURAL NETWORKS)	57



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

6	ΤΕΧΝΟΛΟΓΙΕΣ ΔΙΑΔΙΚΤΥΟΥ.....	60
6.1	ΔΥΝΑΜΙΚΕΣ ΙΣΤΟΣΕΛΙΔΕΣ ΜΕ PHP ΚΑΙ MYSQL.....	60
6.2	ΦΙΛΟΞΕΝΙΑ ΙΣΤΟΣΕΛΙΔΩΝ	61
6.3	ΠΑΚΕΤΑ ΦΙΛΟΞΕΝΙΑΣ.....	61
6.4	WEB SERVERS	62
6.5	ΒΑΣΕΙΣ ΔΕΔΟΜΕΝΩΝ (DATABASES)	64
6.6	ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ (DBMS).....	65
6.7	ΔΙΑΔΙΚΤΥΑΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ.....	65
6.8	LAMP	67
6.9	CONTENT MANAGEMENT SYSTEMS – CMS.....	69



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

Κατάλογος Σχημάτων

Σχήμα 1: Αρχιτεκτονική διεπαφών	6
Σχήμα 2: Πυρήνας ΛΣ	7
Σχήμα 3: Shell OS.....	8
Σχήμα 4: Σύγχρονα Λειτουργικά Συστήματα	10
Σχήμα 5: Virtualization	13
Σχήμα 6: DSV & VDS	15
Σχήμα 7: Οι γενιές του διαδικτύου	17
Σχήμα 8: Social Networks.....	19
Σχήμα 9: LBS Google Latitude	21
Σχήμα 10: Internet of Things.....	23
Σχήμα 11: Internet of Things.....	25
Σχήμα 12: Digital Semantic Web	27
Σχήμα 13: Δίκτυα ανάλογα της γεωγραφικής κάλυψης.....	29
Σχήμα 14: Τεχνολογίες οπτικών ινών FTTx	31
Σχήμα 15: Network LAN architecture	33
Σχήμα 16: IPv4 vs IPv6.....	38
Σχήμα 17: Cloud models.....	40
Σχήμα 18: Cyber risk.....	41
Σχήμα 19: Security model and CIA	42
Σχήμα 20: Types of cyber-attacks in 2015 (www.govtech.com)	43
Σχήμα 21: Disaster Recovery Plan	45
Σχήμα 22: Tor network.....	48
Σχήμα 23: Bitcoin transaction	51
Σχήμα 24: Machine Learning.....	55
Σχήμα 25: Fuzzy System	57



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

Σχήμα 26: Artificial Neural Network	58
Σχήμα 27: Παράδειγμα χρήσης ΤΝΔ στην αναγνώριση κειμένου	59
Σχήμα 28: Virtual Private Hosting	62
Σχήμα 29: Nginx web server.....	63
Σχήμα 30: LAMP	68
Σχήμα 31: How CMS works	70
Σχήμα 32: Open e-class και Ελληνικά ΑΕΙ	73



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

1 Λειτουργικά Συστήματα

Λειτουργικό Σύστημα ΛΣ (Operating System ή OS) ονομάζεται μία συλλογή βασικών προγραμμάτων, η οποία χρησιμοποιείται ως υπόβαθρο, ελέγχει, τη λειτουργία του υπολογιστή, την εκτέλεση των υπόλοιπων προγραμμάτων, τη διαχείριση των περιφερειακών συσκευών, την εξασφάλιση της επικοινωνίας μεταξύ χρήστη και υπολογιστή, καθώς και την κατανομή των διαθέσιμων υπολογιστικών πόρων.



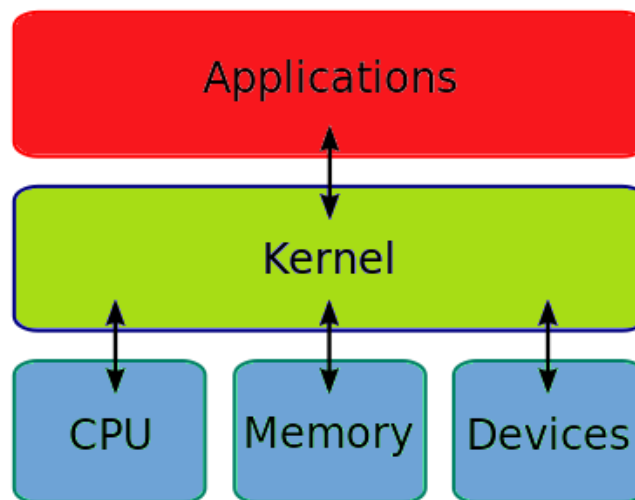
Σχήμα 1: Αρχιτεκτονική διεπαφών

Μια από τις κεντρικές αρμοδιότητες του λειτουργικού συστήματος είναι η διαχείριση των πόρων ενός υπολογιστή και η ελεγχόμενη κατανομή τους στις εκτελούμενες εφαρμογές, έτσι ώστε οι αυτές να είναι σε θέση να προσπελάσουν εύκολα τους πόρους και τις συσκευές του συστήματος χωρίς να χρειάζεται να γνωρίζουν με ακρίβεια τη δομή του υποκείμενου υλικού, αλλά και ώστε πολλαπλές εφαρμογές να μπορούν να εκτελούνται ταυτόχρονα χωρίς να έρχονται σε διένεξη μεταξύ τους ή με τον υπολογιστή. Είναι το πρώτο λογισμικό που «φορτώνεται» στη μνήμη του υπολογιστή μετά την εκτέλεση του BIOS. Οποιοδήποτε λογισμικό φορτωθεί στη συνέχεια βασίζεται στο ΛΣ για την παροχή όλων των υπηρεσιών οι οποίες απαιτούν πρόσβαση στο υλικό.



1.1 Πυρήνας

Ο όρος πυρήνας αφορά τα πιο βασικά μέρη ενός ΛΣ τα οποία αλληλεπιδρούν στενά με το υλικό. Συγκεκριμένα το ΛΣ περιέχει κώδικα χαμηλού επιπέδου (με γλώσσες όπως η C, είτε απευθείας σε συμβολική γλώσσα), αποκλειστικά για την αρχιτεκτονική του επεξεργαστή στην οποία εκτελείται, ο οποίος αναλαμβάνει την υλοποίηση όλων των μηχανισμών. Ο εν λόγω κώδικας ο οποίος κατά την κανονική λειτουργία του υπολογιστή είναι αποθηκευμένος στη μνήμη, αποτελεί τον «πυρήνα». Ο κώδικας των εκτελούμενων εφαρμογών («κώδικας χρήστη») δεν έχει άμεση πρόσβαση στον πυρήνα, ο οποίος όμως παρέχει μία προγραμματιστική διασύνδεση, τις κλήσεις συστήματος, με την οποία τα προγράμματα χρήστη μπορούν να καλούν με ελεγχόμενο τρόπο διαδικασίες που εξάγει ο πυρήνας και παρέχουν υπηρεσίες στον καλούντα. Η σύνταξη, η σημασιολογία και η ονοματολογία των κλήσεων συστήματος διαφέρει από ΛΣ σε ΛΣ, με αποτέλεσμα ένα εκτελέσιμο πρόγραμμα συνήθως να μπορεί να τρέξει μόνο σε ένα ΛΣ. Ο κώδικας χρήστη δεν μπορεί να προσπελάσει μόνος του τους πόρους του υπολογιστή (π.χ. μνήμη συστήματος, περιφερειακά, δίκτυο κλπ) αλλά μπορεί να ζητήσει ό,τι χρειάζεται (π.χ. άνοιγμα ενός αρχείου στον δίσκο) από τον πυρήνα μέσω των κλήσεων συστήματος. Τμήματα του ΛΣ που δεν ανήκουν στον πυρήνα μπορούν να θεωρηθούν π.χ. η διασύνδεση με τον χρήστη, τα πρωτόκολλα επικοινωνίας ανάμεσα στις εφαρμογές, τα πρωτόκολλα διαχείρισης περιόδων εργασιών χρηστών, τα πρωτόκολλα δικτύου κτλ.



Σχήμα 2: Πυρήνας ΛΣ

1.2 Κέλυφος

Το κέλυφος (shell), λέγεται το εν λόγω τμήμα του ΛΣ το οποίο παρέχει μία οπτική και αλληλεπιδραστική διασύνδεση προς τους χρήστες, είναι απολύτως δευτερεύον σε σχέση με τον πυρήνα και τους μηχανισμούς του, παρόλο που πρόκειται για το μόνο συστατικό ενός λειτουργικού συστήματος το οποίο είναι άμεσα ορατό από έναν απλό χρήστη. Υπάρχουν δύο βασικοί τύποι κελύφους: οι γραμμές εντολών, όπου οι χρήστες απλώς πληκτρολογούν εντολές για να διαχειριστούν τα αρχεία και τα προγράμματά τους και οι γραφικές διασυνδέσεις



χρήστη (GUI), όπου οι χρήστες χειρίζονται ένα γραφικό περιβάλλον στηριζόμενο στο ποντίκι και σε οπτικές μεταφορές (εικονίδια, παράθυρα κλπ).

```
tostoday@tostoday:~$ get:5 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main audacious-plugins-data all 3.4-0-webupd8-precise [1,887 kB]
get:6 http://us.archive.ubuntu.com/ubuntu/ precise/universe libmpg123-0 amd64 1:12.1-3.2ubuntu1 [130 kB]
get:7 http://us.archive.ubuntu.com/ubuntu/ precise/universe libguess1 amd64 1.1-1 [11.4 kB]
get:8 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main libaudcore1 amd64 3.4-0-webupd8-precise [100 kB]
get:9 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main libstdplayfp amd64 1.0-0-0-webupd8-precise [136 kB]
get:10 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main libsoxr0 amd64 0.1.1-0-webupd8-precise [91.5 kB]
get:11 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main audacious-plugins amd64 3.4-0-webupd8-precise [1,186 kB]
get:12 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main libmowl12 amd64 0.9.50-2-webupd8-natty2 [34.9 kB]
get:13 http://ppa.launchpad.net/n1ar1n0gard/webupd8/ubuntu/ precise/main audacious amd64 3.4-0-webupd8-precise [344 kB]
Fetched 3,348 kB in 1s (2,045 kB/s)
Selecting previously unselected package audacious-plugins-data.
(Reading database ... 758221 files and directories currently installed.)
Unpacking audacious-plugins-data (from .../audacious-plugins-data_3.4-0-webupd8-precise_all.deb) ...
Selecting previously unselected package libaudcore1.
Unpacking libaudcore1 (from .../libaudcore1_3.4-0-webupd8-precise_amd64.deb) ...
Selecting previously unselected package libbintol1db1.
Unpacking libbintol1db1 (from .../libbintol1db1_1.4-14_amd64.deb) ...
Selecting previously unselected package libb52b0.
Unpacking libb52b0 (from .../libb52b0_3.1.0+dfsg-2ubuntu1_amd64.deb) ...
Selecting previously unselected package libcue1.
Unpacking libcue1 (from .../libcue1_1.4.0-1_amd64.deb) ...
Selecting previously unselected package libfluidsynth1.
Unpacking libfluidsynth1 (from .../libfluidsynth1_1.1.5-2_amd64.deb) ...
Selecting previously unselected package libmpg123-0.
Unpacking libmpg123-0 (from .../libmpg123-0_1.12.1-3.2ubuntu1_amd64.deb) ...
Selecting previously unselected package libstdplayfp.
Unpacking libstdplayfp (from .../libstdplayfp_1.0-0-0-webupd8-precise_amd64.deb) ...
Selecting previously unselected package libsoxr0.
Unpacking libsoxr0 (from .../libsoxr0_0.1.1-0-webupd8-precise_amd64.deb) ...
Selecting previously unselected package audacious-plugins.
Unpacking audacious-plugins (from .../audacious-plugins_3.4-0-webupd8-precise_amd64.deb) ...
Selecting previously unselected package libmowl12.
Unpacking libmowl12 (from .../libmowl12_0.9.50-2-webupd8-natty2_amd64.deb) ...
Selecting previously unselected package libguess1.
Unpacking libguess1 (from .../libguess1_1.1-1_amd64.deb) ...
Selecting previously unselected package audacious.
Unpacking audacious (from .../audacious_3.4-0-webupd8-precise_amd64.deb) ...
Processing triggers for menu ...
Processing triggers for man-db ...
Processing triggers for hicolor-icon-theme ...
```

Σχήμα 3: Shell OS

1.3 Υπηρεσίες

Οι βασικοί υπηρεσίες ενός ΛΣ αφορούν, τη διαχείριση της εκτέλεσης των προγραμμάτων χρήστη (μέσω του μηχανισμού των διεργασιών και στα συστήματα τα οποία υποστηρίζουν ταυτοχρονισμό, πολυδιεργασία, πολυπρογραμματισμό, του κατάλληλου χρονοπρογραμματισμού τους), της επικοινωνίας μεταξύ τους (μέσω των μηχανισμών εικονικής μνήμης και διαδιεργασιακής επικοινωνίας) και των δεδομένων που αυτές χρησιμοποιούν μέσω του συστήματος αρχείων. Με τον τρόπο που είναι υλοποιημένοι οι εν λόγω μηχανισμοί διασφαλίζουν συνήθως τόσο την ασφάλεια του πυρήνα απέναντι στον κώδικα χρήστη, όσο και την ασφάλεια μεταξύ των προγραμμάτων χρήστη, ώστε κανένα να μην παρεμποδίζει ή να επηρεάζει αρνητικά την πρόσβαση των άλλων στους πόρους του συστήματος.

1.3.1 Διαχείριση διεργασιών

Ένα πρόγραμμα που εκτελείται στον υπολογιστή συνιστά μία ή περισσότερες διεργασίες. Πρόκειται για το βασικό μέσο εκτέλεσης προγραμμάτων σε ένα συνηθισμένο ΛΣ. Ο ίδιος κώδικας - πρόγραμμα μπορεί να εκτελείται ταυτόχρονα μέσα από πολλές διαφορετικές διεργασίες οι οποίες μπορεί να ανήκουν σε διαφορετικούς χρήστες. Στα πλαίσια της σειριακής αρχιτεκτονικής von Neumann και των υπολογιστών που έχουν οικοδομηθεί με βάση αυτήν, μόνο μια διεργασία μπορεί να εκτελείται στην CPU οποιαδήποτε χρονική στιγμή επομένως οι ποικίλες διεργασίες οι οποίες είναι ταυτόχρονα ενεργές εκτελούνται ψευδοπαράλληλα, με διαδοχική εναλλαγή του επεξεργαστή μεταξύ τους κάθε λίγη ώρα. Σε ένα παράλληλο σύστημα, όπου υπάρχουν πολλαπλοί επεξεργαστές, η πολυδιεργασία μπορεί να υλοποιηθεί πραγματικά παράλληλα. Σε κάθε περίπτωση, το τμήμα του πυρήνα το οποίο λαμβάνει διάφορες αποφάσεις σχετικά με την ανάθεση των CPU στις διάφορες διεργασίες ονομάζεται χρονοπρογραμματιστής.



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

1.3.2 Διαχείριση μνήμης

Στις σύγχρονες αρχιτεκτονικές η μνήμη είναι οργανωμένη ιεραρχικά. Αρχίζοντας από την ταχύτερη: καταχωρητές, κρυφή μνήμη, κύρια μνήμη και δευτερεύουσα μνήμη (π. χ. σκληροί δίσκοι). Το τμήμα εκείνο του ΛΣ που καλείται διαχειριστής μνήμης συντονίζει τη χρήση των διαφόρων τύπων μνήμης, καταγράφοντας ποια τμήματά τους είναι διαθέσιμα, ποια είναι δεσμευμένα και αναλόγως με τις απαιτήσεις των διεργασιών, εκχωρεί ή απελευθερώνει τμήματα για να τα χρησιμοποιήσουν οι τελευταίες. Αυτή η δραστηριότητα λέγεται διαχείριση εικονικής μνήμης, αφού η συνολική μνήμη που είναι σε θέση να αξιοποιήσουν οι διεργασίες κατά την εκτέλεσή τους μπορεί να υπερβαίνει το μέγεθος της κύριας μνήμης (της φυσικής μνήμης RAM), μέσω της δέσμευσης ενός τμήματος του σκληρού δίσκου από τον πυρήνα το οποίο χρησιμοποιείται από τον τελευταίο σαν επέκταση της κύριας μνήμης. Τα προηγμένα ΛΣ αποφεύγουν, όπου και όταν είναι δυνατό, τη χρήση αυτής της τεχνικής, επειδή η χρήση δευτερεύουσας μνήμης ως κύριας μειώνει την ταχύτητα του συστήματος.

1.3.3 Συστήματα αρχείων

Το τμήμα του ΛΣ που ονομάζεται διαχειριστής συστήματος αρχείων δημιουργεί την αφαιρετική έννοια των αρχείων και των δένδρων καταλόγων. Έτσι ο χρήστης και οι εφαρμογές του έχουν τη δυνατότητα να βλέπουν τη δευτερεύουσα μνήμη σαν ένα σύνολο από αρχεία, τα οποία μπορούν να δημιουργούν, να τροποποιούν, να διαγράφουν, να μετακινούν και να αντιγράφουν.

1.3.4 Δικτύωση

Τα περισσότερα σύγχρονα λειτουργικά συστήματα επιτρέπουν τη σύνδεση ενός υπολογιστή τόσο σε τοπικά δίκτυα όσο και στο Διαδίκτυο, ενσωματώνοντας στον κώδικα τους την υλοποίηση των απαιτούμενων αντίστοιχων πρωτοκόλλων.

1.3.5 Εσωτερική ασφάλεια

Με τον όρο αυτό, γνωστό και ως προστασία μνήμης, εννοούνται οι ενέργειες στις οποίες προβαίνει το ΛΣ προκειμένου να προφυλάξει τους πόρους μιας διεργασίας από τις παρεμβάσεις άλλων διεργασιών που τυχαίνει να εκτελούνται (ψευδο) παράλληλα (δεν πρέπει μια διεργασία να εγγράφει δεδομένα στο τμήμα εκείνο της μνήμης που έχει εκχωρηθεί σε άλλη διεργασία).

1.3.6 Εξωτερική ασφάλεια

Το λειτουργικό σύστημα ενός υπολογιστικού συστήματος, το οποίο υποστηρίζει πολλούς διαφορετικούς χρήστες, εκτελεί ένα ευρύ φάσμα εφαρμογών και είναι συνδεδεμένο στο δίκτυο. Το γεγονός αυτό αυξάνει τις απαιτήσεις ασφάλειας των αντικειμένων του, επειδή συντρέχει κίνδυνος απώλειας, καταστροφής, διαγραφής, τροποποίησης ή διάδοσης τους, χωρίς την απαιτούμενη δικαιοδοσία. Ενδεικτικές ευπάθειες και απειλές αποτελούν η απώλεια συνθηματικών, η μη εξουσιοδοτημένη εκτέλεση λογισμικού με στόχο την σπατάλη των διαθέσιμων πόρων, καθώς και κακόβουλο λογισμικό ή ενέργειες. Οι συνηθέστεροι μηχανισμοί οι οποίοι εν μέρει προσφέρουν ασφάλεια σε ένα λειτουργικό σύστημα είναι, οι



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

βιομετρικοί μέθοδοι, αγνωστικά πρωτόκολλα, μηχανισμοί δημοσίου κλειδιού και σύνθετα συνθηματικά.

1.3.7 Οδηγοί συσκευών

Ένας οδηγός συσκευής είναι το λογισμικό συστήματος που χρησιμοποιεί το ΛΣ και οι διεργασίες των χρηστών κάθε φορά που πρέπει να ανταλλάξουν δεδομένα με τη συγκεκριμένη συσκευή. Συνήθως οι οδηγοί συσκευών περιλαμβάνονται στο λογισμικό που συνοδεύει την περιφερειακή συσκευή κατά την αγορά της. Ωστόσο τα περισσότερα ΛΣ διατίθενται και με ορισμένους προκατασκευασμένους, γενικής χρήσης οδηγούς συσκευών, συνήθως όχι τόσο βελτιστοποιημένους από άποψη ταχύτητας.



Σχήμα 4: Σύγχρονα Λειτουργικά Συστήματα

1.4 Σύγχρονα Λειτουργικά Συστήματα

Τα δημοφιλέστερα λειτουργικά συστήματα έχουν διαμορφωθεί σε δύο μεγάλες οικογένειες: αυτή των Unix-συμβατών και την οικογένεια των Microsoft Windows. Τα Unix-συμβατά Λειτουργικά Συστήματα αποτελούν μια πολυποίκιλη ομάδα, με πολλές κύριες υποκατηγορίες συμπεριλαμβανομένων των BSD και GNU/Linux. Τα Unix ΛΣ τρέχουν σε μια μεγάλη γκάμα από αρχιτεκτονικές υπολογιστών. Χρησιμοποιούνται πολύ σαν συστήματα εξυπηρετητές στις επιχειρήσεις και σε σταθμούς εργασίας σε ακαδημαϊκούς και μηχανολογικούς χώρους εργασίας. Παραλλαγές του Unix που διακινούνται ως Ελεύθερο λογισμικό, όπως το GNU/Linux και BSD αυξάνουν σε δημοτικότητα στο χώρο των σταθμών εργασίας και των προσωπικών υπολογιστών. Το Solaris της Sun είναι ένα τέτοιο πολύμορφο αλλά αληθινό Unix και μπορεί να τρέχει στους σταθμούς εργασίας της Sun αλλά και στον μικρότερο υπολογιστή αρχιτεκτονικής x86. Το ΛΣ Mac OS X της Apple είναι μια παραλλαγή του BSD και έχει αντικαταστήσει τα προηγούμενα (μη-unix) MacOS ΛΣ της Apple σε μια σχετικά μικρή αλλά αφοσιωμένη αγορά, έχοντας γίνει ένα πολύ δημοφιλές Unix.

1.4.1 MS Windows

Η οικογένεια των Microsoft Windows ΛΣ ξεκίνησε σαν ένα επίπεδο γραφικής διασύνδεσης πάνω από το παλιότερο MS-DOS περιβάλλον για τους IBM PC. Οι σύγχρονες εκδόσεις των Windows βασίζονται στον πυρήνα των Windows NT που πρωτοδιαμορφώθηκε στο OS/2, με



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

πιο πρόσφατη έκδοση τα Windows 8 (σε αναμονή των Windows 10). Τα Windows τρέχουν πάνω σε 32- και 64-bit Intel και AMD υπολογιστές. Τα Windows είναι το δημοφιλέστερο ΛΣ προσωπικών υπολογιστών απολαμβάνοντας ένα σχεδόν μονοπώλιο του 90% του παγκόσμιου μεριδίου αγοράς των προσωπικών υπολογιστών. Επίσης χρησιμοποιείται ευρέως και στους εξυπηρετητές υποστηρίζοντας εφαρμογές όπως Web εξυπηρετητές (Web Servers) και εξυπηρετητές βάσεων δεδομένων (DBMS Servers).

1.4.2 Linux

Το Linux περιλαμβάνει χαρακτηριστικά από πολλά γνωστά λειτουργικά συστήματα UNIX όπως SVR4 - AT&T, BSD, AIX-IBM, HP-UX, Solaris - Sun Microsystems. Αναπτύχθηκε αρχικά από τον Linus Torvalds το 1991 ως ένα λειτουργικό σύστημα για IBM συμβατούς προσωπικούς υπολογιστές, βασισμένους στον μικροεπεξεργαστή Intel 80386. Με την πάροδο του χρόνου η συμμετοχή πολλών στην ανάπτυξή του το έκανε διαθέσιμο σε πολλές άλλες αρχιτεκτονικές όπως Alpha, SPARC, Motorola MC680x0, PowerPC, και IBM System/390. Ένα από τα σημαντικότερα πλεονεκτήματά του είναι ότι διανέμεται δωρεάν ο πηγαίος του κώδικας, υπό την GNU Public License και είναι ανοικτός και διαθέσιμος σε οποιονδήποτε (ο κώδικας είναι διαθέσιμος για download από το επίσημο site <http://www.kernel.org>). Για την επιτυχία οποιουδήποτε Linux project είναι απαραίτητος ο GNU C compiler. Ο πυρήνας του Linux προσπαθεί να ενσωματώσει χαρακτηριστικά που υπάρχουν σε άλλους πυρήνες Unix. Οποιοσδήποτε χρησιμοποιεί το Linux, ή δημιουργεί τη δική του διανομή δεν πρέπει να θεωρεί το παραγόμενο προϊόν ιδιοκτησία του. Οι πιο γνωστές διανομές Linux είναι RedHat (www.redhat.com), Debian (www.debian.org/), Mandrake (www.mandrakesoft.com), SuSE (www.suse.com/), Slackware (www.slackware.com/), Ubuntu (www.ubuntu.com).

1.4.3 Mac OS

Το Mac OS είναι ένα λειτουργικό σύστημα με γραφικό περιβάλλον που αναπτύχθηκε από την Apple για τους ηλεκτρονικούς υπολογιστές Macintosh. Το Mac OS, πριν ακόμη η Apple το ονομάσει έτσι, εφαρμόστηκε για πρώτη φορά το 1984 στον πρωτότυπο υπολογιστή Macintosh ως μη διακριτό τμήμα του συστήματος και συνήθως αναφερόταν απλά ως το "λογισμικό συστήματος". Η Apple εσκεμμένα υποβάθμιζε την ύπαρξη του λειτουργικού συστήματος κατά τα πρώτα χρόνια του Macintosh για να δώσει έμφαση στην εύκολη χρήση του υπολογιστή της αλλά και για να διαφοροποιήσει το προϊόν της από άλλα λειτουργικά συστήματα όπως το MS-DOS, το οποίο είχε την φήμη του δύσχρηστου και τεχνικά απαιτητικού συστήματος, και το Unix. Η ευκολία επικεντρωνόταν στο ότι το Mac OS ήταν από τα πρώτα λειτουργικά που χρησιμοποίησαν γραφικό περιβάλλον χρήστη αντί για γραμμή εντολών για την επικοινωνία ανθρώπου και υπολογιστή. Μεγάλο μέρος του αρχικού λογισμικού συστήματος ήταν αποθηκευμένο στη μνήμη ROM καθώς οι πρώτοι Macintosh δεν διέθεταν σκληρό δίσκο, με ενημερώσεις να παρέχονται συνήθως δωρεάν στους εμπόρους από την Apple, μέσω δισκετών. Το 2000 η Apple κυκλοφόρησε τη δέκατη έκδοση του Mac OS, το Mac OS X, ουσιαστικά ένα ριζικά διαφορετικό λειτουργικό σύστημα βασισμένο στη διανομή BSD του Unix και τον μικροπυρήνα Mach. το Mac OS X έχει δεχτεί οκτώ (8) αναβαθμίσεις, που όλες φέρουν ονόματα αιλουροειδών Mac OS X v10.1 "Puma" (Πούμα), Mac OS X v10.2 "Jaguar" (Ιαγουάρος), Mac OS X v10.3 "Panther" (Πάνθηρας), Mac OS X v10.4 "Tiger" (Τίγρης), Mac OS X v10.5 "Leopard" (Λεοπάρδαλη), Mac OS X v10.6 "Snow Leopard"



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

(Λεοπάρδαλη του Χιονιού), Mac OS X v10.7 "Lion" (Λιοντάρι), Mac OS X v10.8 "Mountain Lion" (Βουνίσιο Λιοντάρι), Mac OS X v10.9 "Mavericks". Αν και το Mac OS X δημιουργήθηκε για να λειτουργεί μόνο σε ηλεκτρονικούς υπολογιστές Macintosh, οι οποίοι κατασκευάζονται από την ίδια την Apple, από το 2006 και μετά οι Macintosh χρησιμοποιούν σχεδόν το ίδιο hardware (υλικό) με αυτό των περισσότερων άλλων κατασκευαστών, δίνοντας τη δυνατότητα για μη εξουσιοδοτημένη χρήση του Mac OS X σε μη-Apple υπολογιστές με τη χρήση hacks.

1.4.4 iOS

Το iOS (προηγουμένως iPhone OS) είναι ένα λογισμικό για κινητά το οποίο αναπτύχθηκε και διανέμεται από την Apple. Αρχικά παρουσιάστηκε το 2007 για το iPhone, ενώ υποστηρίζει και άλλες συσκευές της Apple όπως το iPod touch, το iPad και το Apple TV. Αντίθετα από το Windows Phone της Microsoft και το Android της Google, η Apple δεν δίνει την άδεια για την εγκατάσταση του λογισμικού iOS σε συσκευές που δεν είναι κατασκευής Apple.

1.4.5 Android OS

Το Android είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας το οποίο τρέχει τον πυρήνα του λειτουργικού Linux. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού Java, ελέγχοντας την συσκευή μέσω βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV), αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένου Η/Υ και σε άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Οι συσκευές με Android έχουν περισσότερες πωλήσεις από όλες τις συσκευές Windows, iOS και Mac OS X μαζί.

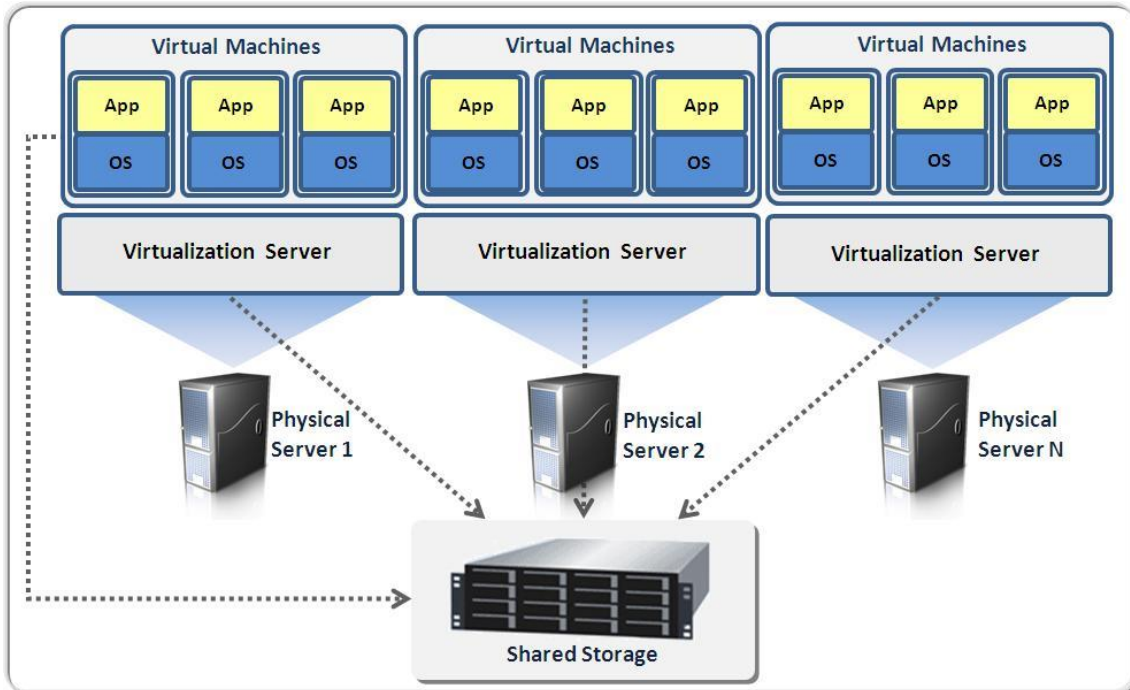
1.5 Virtualization

Στην επιστήμη της πληροφορικής, η εικονικοποίηση (virtualization) είναι ένας ευρύς όρος των υπολογιστικών συστημάτων που αναφέρεται σε έναν μηχανισμό αφαίρεσης, στοχευμένο στην απόκρυψη λεπτομερειών της υλοποίησης και της κατάστασης ορισμένων υπολογιστικών πόρων από πελάτες των πόρων αυτών (π.χ. εφαρμογές, άλλα συστήματα, χρήστες κλπ). Η εν λόγω αφαίρεση μπορεί είτε να αναγκάζει έναν πόρο να συμπεριφέρεται ως πλειάδα πόρων (π.χ. μία συσκευή αποθήκευσης σε διακομιστή τοπικού δικτύου), είτε πολλαπλούς πόρους να συμπεριφέρονται ως ένας (π.χ. συσκευές αποθήκευσης σε καταναμημένα συστήματα). Η εικονικοποίηση δημιουργεί μία εξωτερική διασύνδεση η οποία αποκρύπτει την υποκείμενη υλοποίηση (π.χ. πολυπλέκοντας την πρόσβαση από διαφορετικούς χρήστες). Αυτή η προσέγγιση στην εικονικοποίηση αναφέρεται ως εικονικοποίηση πόρων. Μία άλλη προσέγγιση, ίδιας όμως νοοτροπίας, είναι η εικονικοποίηση πλατφόρμας, όπου η αφαίρεση που επιτελείται προσομοιώνει ολόκληρους υπολογιστές. Το αντίθετο της εικονικοποίησης είναι η διαφάνεια: ένας εικονικός πόρος είναι ορατός, αντιληπτός, αλλά στην πραγματικότητα ανύπαρκτος, ενώ ένας διαφανής πόρος είναι υπαρκτός αλλά αόρατος.



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων



Σχήμα 5: Virtualization

Σε ένα εικονικοποιημένο σύστημα συμμετέχουν τρεις παράγοντες: ο πελάτης μιας υπηρεσίας, ο παροχέας της υπηρεσίας και ένας ενδιαμέσος. Ο πελάτης και ο παροχέας αλληλεπιδρούν μέσω μίας πρότυπης διασύνδεσης, τις κλήσεις προς την οποία όμως αναχαιτίζει ο ενδιαμέσος. Ο τελευταίος επιτελεί την εικονικοποίηση λειτουργώντας ως παροχέας για τον πελάτη και ως πελάτης για τον παροχέα. Ένα παράδειγμα είναι ο μηχανισμός εικονικής μνήμης των σύγχρονων λειτουργικών συστημάτων, όπου ο διαχειριστής εικονικής μνήμης (ο ενδιαμέσος) παρεμβάλλεται μεταξύ ενός πραγματικού χώρου διευθύνσεων (παροχέας) και ενός εικονικού που γίνεται αντιληπτός από κάθε διεργασία (πελάτης). Ο ενδιαμέσος παρέχει την ψευδαισθηση πολλών ισομεγέθων χώρων διευθύνσεων (ένας για κάθε διεργασία), ενώ στην πραγματικότητα υπάρχει μόνο ένας συνολικά (η πραγματική μνήμη). Ο πελάτης και ο παροχέας δεν γνωρίζουν τίποτα για την εικονικοποίηση και τη μεσολάβηση του ενδιαμέσου.

Ιδιαίτερο ενδιαφέρον παρουσιάζει η εικονικοποίηση πλατφόρμας, όπου ένα λογισμικό ελέγχου («επόπτης» ή hypervisor) εκτελούμενο σε πραγματικό υλικό προσομοιώνει ένα υπολογιστικό περιβάλλον, μία «εικονική μηχανή», επάνω από το οποίο μπορεί να τρέξει κάποιο φιλοξενούμενο λογισμικό (συνήθως ένας πλήρης πυρήνας), απομονωμένο από το υπόλοιπο σύστημα. Η θεμελιώδης λογική πίσω από την εικονικοποίηση πλατφόρμας είναι η αρχή πως οποιαδήποτε λειτουργία μπορεί να εκτελεστεί είτε από λογισμικό είτε από εξειδικευμένο υλικό• οι μόνες διαφορές αφορούν την ευελιξία και την απόδοση. Είναι δυνατόν να προσομοιώνονται ταυτόχρονα πολλαπλές εικονικές μηχανές, εντελώς απομονωμένες μεταξύ τους, από το ίδιο λογισμικό ελέγχου. Η εικονικοποίηση πλατφόρμας εμφανίστηκε αρχικά τη δεκαετία του 1960, πριν από την επέλαση των μικροϋπολογιστών, σε μεγάλα, συγκεντρωτικά συστήματα (mainframes), αλλά μετά το 2000 και την αλματώδη



Δημοκρίτειο Πανεπιστήμιο Θράκης

Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

αύξηση των επιδόσεων του υλικού των PC έχει γίνει πλέον κοινή πρακτική. Υπάρχουν πολλά είδη εικονικοποίησης πλατφόρμας, με σημαντικότερα:

1.5.1 Εξομοίωση

Η εικονική μηχανή προσομοιώνει εξολοκλήρου μία αρχιτεκτονική υλικού, πιθανώς διαφορετική από το πραγματικό υποκείμενο υλικό, επιτρέποντας έτσι να εκτελεστεί επάνω της ένα μη τροποποιημένο, φιλοξενούμενο ΛΣ σχεδιασμένο για τον εξομοιούμενο επεξεργαστή (π.χ. QEMU, έκδοση για PowerPC του VirtualPC κλπ). Η εξομοίωση είναι διερμηνεία σε χρόνο εκτέλεσης του κώδικα του φιλοξενούμενου ΛΣ, με έναν κύκλο ανάγνωσης-αποκωδικοποίησης-εκτέλεσης όπου κάθε εντολή που ανήκει στο σύνολο εντολών του επεξεργαστή-πηγή μεταφράζεται σε μία εντολή του συνόλου εντολών του επεξεργαστή-στόχου. Παράλληλα η εικονική μηχανή παρέχει μία αφαίρεση της μνήμης, των συσκευών Εισόδου / Εξόδου κλπ, φροντίζοντας ώστε κάθε μεταφρασμένη εντολή που απευθύνεται σε αυτά τα υποσυστήματα να τροποποιεί μόνο τις αφαιρέσεις / λογικές αναπαραστάσεις τους, οι οποίες κατευθύνονται και υλοποιούνται από το λογισμικό ελέγχου, και όχι το πραγματικό υλικό. Προκειμένου να αυξηθούν οι επιδόσεις είναι δυνατόν να χρησιμοποιηθεί δυναμική μετάφραση αντί για απλή εξομοίωση, όπου οι μεταφρασμένες εντολές αποθηκεύονται σε κρυφή μνήμη και μπορούν να επαναχρησιμοποιηθούν αργότερα χωρίς εκ νέου μετάφραση, ή δυναμική επαναμεταγλώττιση, όπου εκτός της χρήσης κρυφής μνήμης γίνεται και βελτιστοποίηση κρίσιμων τμημάτων του κώδικα (παρόμοια με τη μεταγλώττιση JIT της Java, του .NET και άλλων παρόμοιων πλατφορμών υψηλού επιπέδου).

1.5.2 Πλήρης

Η εικονική μηχανή προσομοιώνει επαρκές τμήμα του πραγματικού υποκείμενου υλικού ώστε να επιτρέπει την εκτέλεση επάνω της ενός μη τροποποιημένου, φιλοξενούμενου ΛΣ σχεδιασμένου για τον ίδιο τύπο επεξεργαστή με την πραγματική CPU (π.χ. VirtualPC, VMware, Win4Lin κλπ). Στην πλήρη εικονικοποίηση δεν χρειάζεται εξομοίωση του συνόλου εντολών του επεξεργαστή και μάλιστα ένα τμήμα του κώδικα του φιλοξενούμενου ΛΣ μπορεί να εκτελείται απευθείας από το υλικό, χωρίς μεσολάβηση του επόπτη, αρκεί να μην επηρεάζει υποσυστήματα εκτός του άμεσου ελέγχου του τελευταίου. Τα κρίσιμα σημεία του φιλοξενούμενου κώδικα ωστόσο, όπως αυτά που προσπαθούν να αποκτήσουν πρόσβαση στο υλικό (π.χ. κλήσεις συστήματος), συλλαμβάνονται από το λογισμικό ελέγχου και προσομοιώνονται, αφού τα αποτελέσματα κάθε λειτουργίας που επιτελείται σε μία εικονική μηχανή δεν επιτρέπεται να τροποποιούν την κατάσταση άλλων εικονικών μηχανών, του επόπτη ή του υλικού. Αν το πραγματικό υλικό βοηθά και επιταχύνει τη λειτουργία του λογισμικού ελέγχου τότε η πλήρης εικονικοποίηση ονομάζεται εγγενής (native). Η βοήθεια αυτή αφορά κυρίως εύκολη διάκριση μεταξύ εντολών που μπορούν να εκτελεστούν απευθείας και εντολών που πρέπει να προσομοιωθούν από το λογισμικό. Όπως και στην εξομοίωση η εικονική μηχανή παρέχει στο φιλοξενούμενο ΛΣ μία αφαίρεση της μνήμης, των συσκευών Εισόδου / Εξόδου κλπ, ενώ η εγγενής εκτέλεση μεγάλου μέρους του κώδικα παρέχει πολύ καλύτερες επιδόσεις σε σχέση με την εξομοίωση.



Δημοκρίτειο Πανεπιστήμιο Θράκης

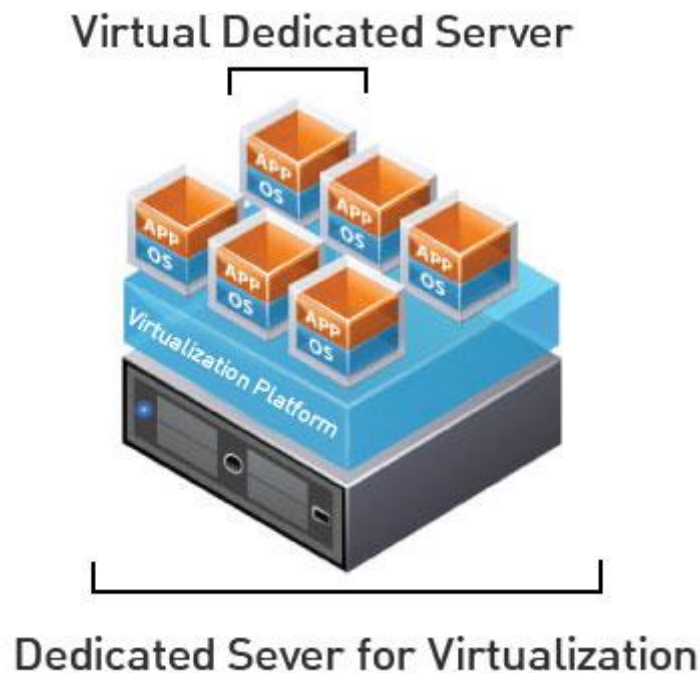
Τμήμα Δασολογίας και Διαχείρισης Περιβάλλοντος και Φυσικών Πόρων

1.5.3 Παραεικονικοποίηση

Η εικονική μηχανή δεν προσομοιώνει επακριβώς το υλικό αλλά παρέχει στις εικονικές μηχανές ένα API, μία προγραμματιστική διασύνδεση, ώστε να επιτρέπει την εκτέλεση επάνω της ενός τροποποιημένου, φιλοξενούμενου ΛΣ σχεδιασμένου για εκτέλεση από τον συγκεκριμένο επόπτη (π.χ. Denali, XEN). Το προαναφερθέν API ονομάζεται διασύνδεση υπερκλήσεων και ένα λειτουργικό σύστημα πρέπει να μεταφερθεί ρητά σε έκδοση κατάλληλη για εκτέλεση από ένα σύστημα παραεικονικοποίησης, ώστε ο φιλοξενούμενος πυρήνας αντί να προσπελαύνει το υλικό άμεσα να εκτελεί υπερκλήσεις και να αναμένει απαντήσεις ή ασύγχρονες ειδοποιήσεις από τον επόπτη. Το όφελος από τη βελτίωση των επιδόσεων και την απλοποίηση της γραφής του επόπτη είναι μεγάλο.

1.5.4 Σύστημα-ξενιστής

Μία άλλου είδους κατηγοριοποίηση των εποπτών εικονικοποίησης είναι σε αυτούς που εκτελούνται ως εφαρμογές πάνω από ένα υποκείμενο λειτουργικό (π.χ. VirtualPC, VMware κλπ) και σε αυτούς που λειτουργούν οι ίδιοι ως λιτά λειτουργικά συστήματα και άρα εκτελούνται απευθείας επάνω από το υποκείμενο υλικό (π.χ. Xen).



Σχήμα 6: DSV & VDS