



Περιεχόμενα

1. Εισαγωγή στις Βασικές Έννοιες του Υλικού (Hardware Concepts)

- Κύριες αρχιτεκτονικές υλικού: Επεξεργαστές (CPU), Μνήμη (RAM), Γραφικά (GPU), Δίσκοι (HDD, SSD).
- Διαύλοι επικοινωνίας (buses), I/O συστήματα, και βασικές αρχές λειτουργίας hardware.
- Κεντρικές έννοιες, όπως η πολυεπεξεργασία (multicore systems) και οι μνήμες cache.
- Εισαγωγή στο virtualization του hardware.

2. Εισαγωγή στα λειτουργικά συστήματα (Operating Systems)

- Λειτουργικά συστήματα
- Το UEFI.
- Boot loader
- Μετά το UEFI.

3. Διαχείριση Διεργασιών και Νημάτων (Processes and Threads)

- Ορισμός και έννοια της διεργασίας (process).
- Δημιουργία, διαχείριση και τερματισμός διεργασιών.
- Νήματα (threads) και σύγχρονος πολυνηματικός προγραμματισμός. Ζητήματα συγχρονισμού.
- Προβλήματα κριτικών περιοχών και παραδείγματα συγχρονισμού.

4. Διαχείριση Μνήμης

- Εικονική μνήμη, σελιδοποίηση (paging), και αντικατάσταση σελίδων.
- Τεχνικές διαχείρισης μνήμης σε λειτουργικά συστήματα.

5. Συστήματα Αρχείων και Αποθήκευσης

- Δομές και λειτουργίες συστήματος αρχείων.
- Διαχείριση χώρου αποθήκευσης, εσωτερικές δομές και εξωτερική αποθήκευση.

6. Διαχείριση Εισόδου/Εξόδου (I/O Management)

- Βασικές αρχές και λειτουργία συστημάτων I/O.
- Οδηγοί συσκευών (device drivers) και I/O συστήματα.

7. Διαχείριση Πόρων και Προγραμματισμός CPU

- Αλγόριθμοι προγραμματισμού διεργασιών και διαχείριση των πόρων του συστήματος.

8. Ασφάλεια και Προστασία (Security and Protection)

- Μέθοδοι προστασίας δεδομένων και δικαιωμάτων πρόσβασης.
- Μηχανισμοί ασφάλειας στα λειτουργικά συστήματα.

9. Εισαγωγή στις Τεχνολογίες Νέφους (Cloud Technologies)

- Τύποι υποδομών νέφους (IaaS, PaaS, SaaS).
- Virtualization και containers (π.χ. Docker, Kubernetes).

10. Εικονικοποίηση και Διαχείριση Εικονικών Μηχανών (Virtualization and VM Management)

- Hypervisors τύπου 1 και 2.
- Εικονικοί πόροι και διαχείριση VMs.

11. Διαχείριση Πόρων στο Νέφος (Cloud Resource Management)

- Δυναμική κατανομή πόρων και προγραμματισμός στο νέφος.
- Παρακολούθηση και διαχείριση απόδοσης νέφους.

12. Κυβερνοασφάλεια στο Νέφος (Cloud Security)

- Κίνδυνοι ασφαλείας στο νέφος και τεχνικές προστασίας.
- Προστασία δεδομένων και μηχανισμοί ελέγχου πρόσβασης.

13. Τάσεις και μέλλον των λειτουργικών συστημάτων

- Κίνδυνοι ασφαλείας στο νέφος και τεχνικές προστασίας.
- Προστασία δεδομένων και μηχανισμοί ελέγχου πρόσβασης.

Διάλεξη 01

Βασικές Έννοιες Υλικού για Λειτουργικά Συστήματα και Νέφη

Στόχοι:

- Κατανόηση των κύριων στοιχείων του υλικού που αλληλεπιδρούν με τα λειτουργικά συστήματα.
- Εισαγωγή σε έννοιες επεξεργαστών, μνήμης, δίσκων, και διαύλων.
- Εισαγωγή στην εικονικοποίηση (virtualization) και τη σχέση της με το υλικό.

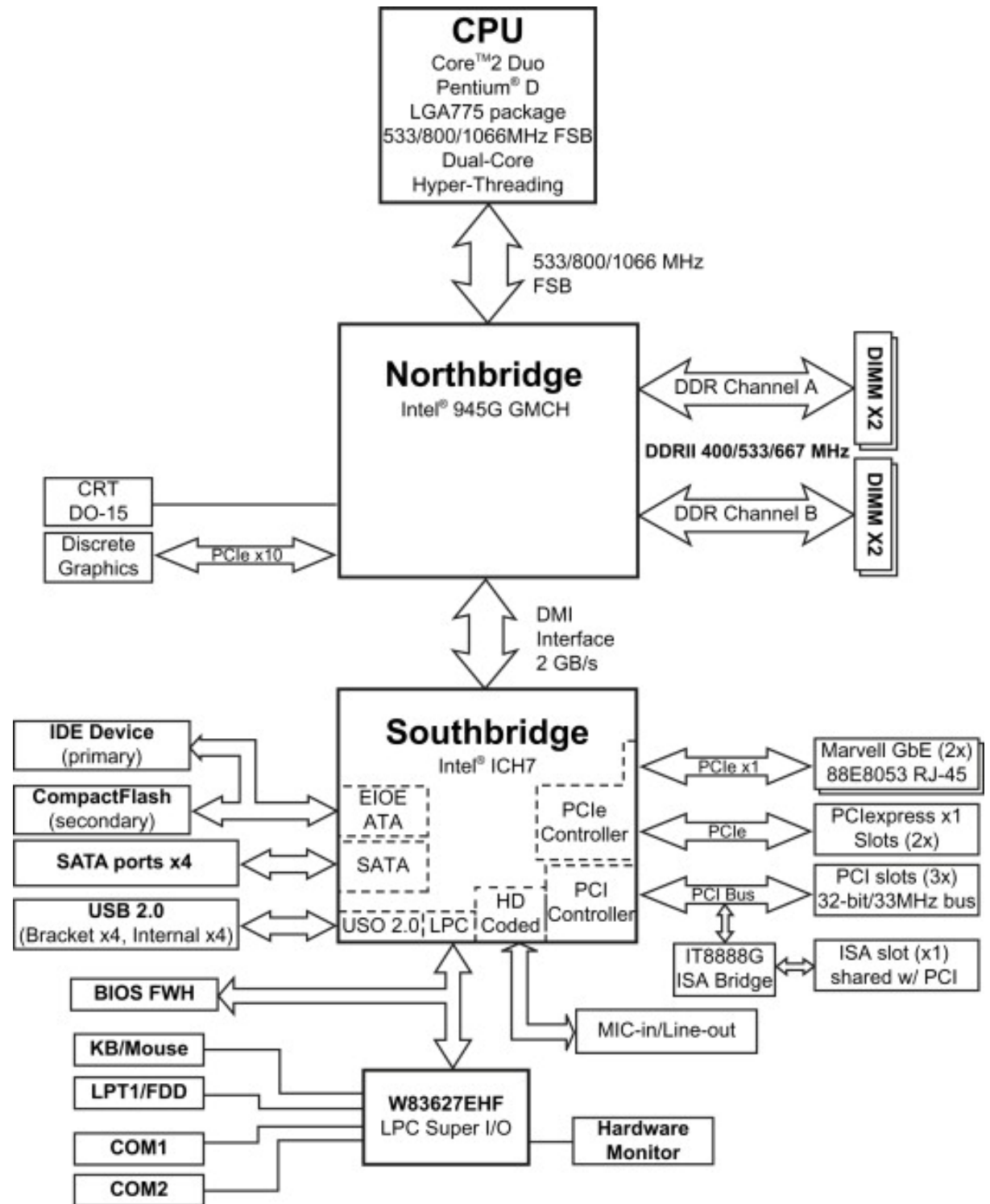
Βασικό Διάγραμμα

Δεξιά εμφανίζεται το βασικό διάγραμμα, ενός υπολογιστικού συστήματος. Ο εγκέφαλος είναι η κεντρική μονάδα επεξεργασίας CPU.

Η λειτουργία οποιουδήποτε λειτουργικού συστήματος εξαρτάται από την αλληλεπίδραση του με το υλικό (hardware). Τα λειτουργικά συστήματα λειτουργούν ως γέφυρα ανάμεσα στις εφαρμογές και τους πόρους του υλικού, διαχειρίζοντας και συντονίζοντας την επικοινωνία και τη χρήση των διαθέσιμων πόρων από διαφορετικά προγράμματα. Για να κατανοήσουμε πώς λειτουργούν τα λειτουργικά συστήματα, είναι σημαντικό να έχουμε μια σαφή εικόνα των βασικών στοιχείων του υλικού και της λειτουργίας τους.

Η αμεσότερη διασύνδεση της κεντρικής μονάδας επεξεργασίας (intel, amd, m) είναι η μνήμη. Μέσω της βόρειας γέφυρας (Northbridge) η κεντρική μονάδα επεξεργασίας διασυνδέεται με τη μνήμη RAM. Επίσης υπάρχει άμεση διασύνδεση με τη μονάδα γραφικών (αριστερά). Επεξεργαστές όπως οι νέοι M1,M2 της Apple έχουν ενσωματωμένες στο ίδιο chip, τόσο τη μνήμη όσο και τη GPU. Έτσι μέσα στη CPU έχουμε ενσωματωμένους πυρήνες γραφικών, πυρήνες κεντρικής επεξεργασίας, όπως επίσης και Gbytes μνήμης.

Σε δεύτερο επίπεδο έχουμε τη νότια γέφυρα (Southbridge Chipset) όπου μέσω αυτής, διάφορες συσκευές έχουν πρόσβαση στον επεξεργαστή, τη μνήμη και το αντίστροφο. Το Chipset διασυνδέει συσκευές δίσκων (SATA, IDE,Flash, USB), πληκτρολόγιο, ποντίκι, σειριακές συσκευές, κάρτες δικτύου και υποδοχές επέκτασης ISA, PCI, PCI express, PCIe2.0 κλπ.



1. Επεξεργαστές (Processors)

Ο επεξεργαστής (CPU) είναι η «καρδιά» κάθε υπολογιστικού συστήματος. Είναι υπεύθυνος για την εκτέλεση των εντολών των προγραμμάτων. Τα κύρια μέρη ενός επεξεργαστή περιλαμβάνουν:

Μονάδα Ελέγχου (Control Unit):

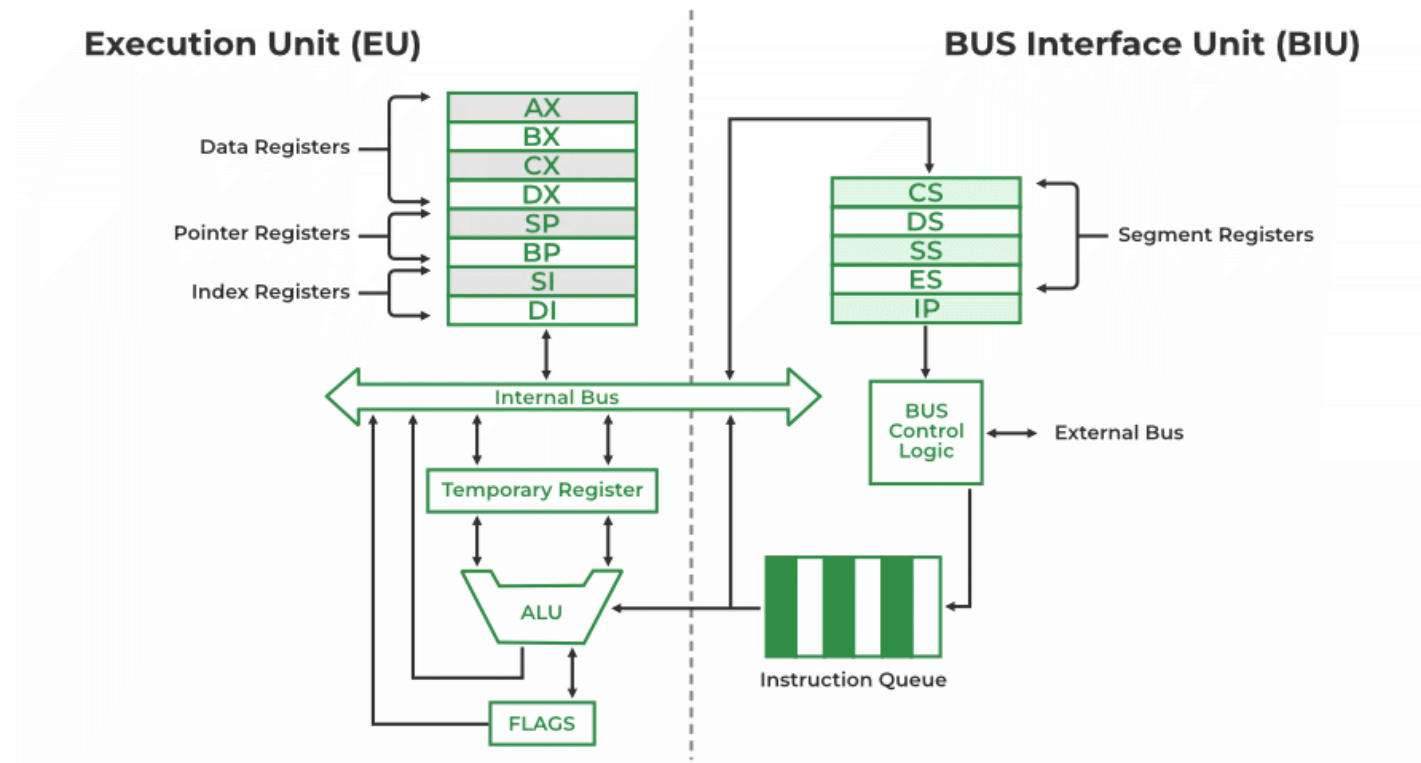
Διαχειρίζεται τη ροή των δεδομένων και των εντολών.

Αριθμητική Λογική Μονάδα (ALU): Εκτελεί αριθμητικές και λογικές πράξεις.

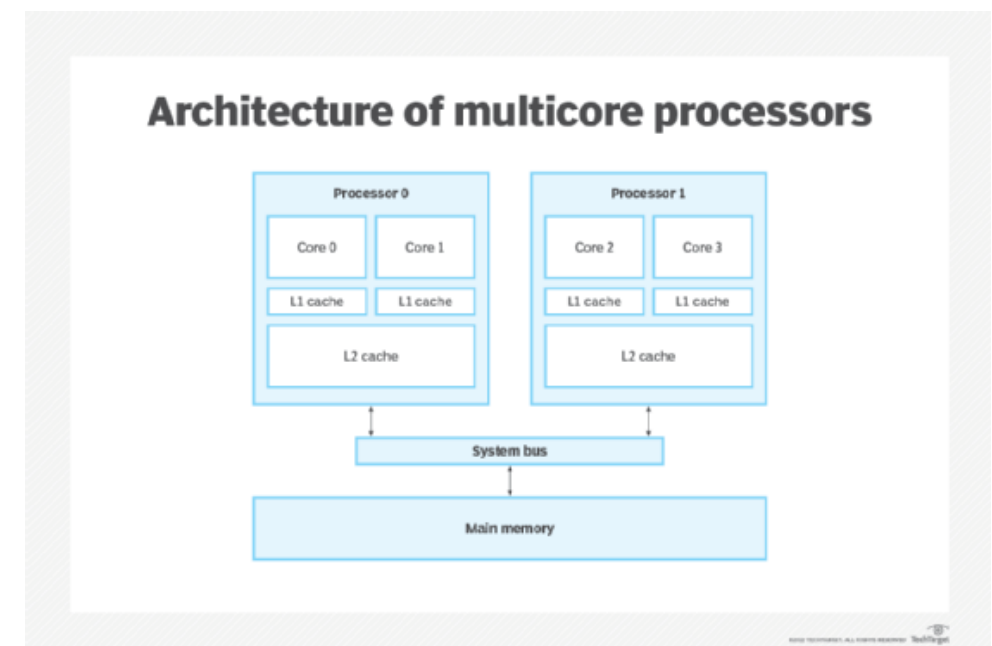
Καταχωρητές (Registers): Μικρές, γρήγορες αποθήκες δεδομένων που χρησιμοποιούνται από την CPU για την εκτέλεση εντολών.

Πολυπύρρηνοι Επεξεργαστές (Multicore Processors)

Οι σύγχρονοι επεξεργαστές περιέχουν πολλούς πυρήνες, που επιτρέπουν την εκτέλεση πολλών διεργασιών ταυτόχρονα (multithreading). Αυτό ενισχύει την απόδοση και την αποδοτικότητα του συστήματος, ειδικά σε εφαρμογές που εκμεταλλεύονται την παράλληλη επεξεργασία.



Διάγραμμα του Intel 8086.



Σύγχρονοι επεξεργαστές

Η σειρά Intel 11th-gen Core έφτασε το μέγιστο σε 10-cores/20-threads, ενώ η νεότερη σειρά 12th-gen φτάνει τα 24 threads, με υβριδικό σχεδιασμό που περιλαμβάνει 8 πυρήνες απόδοσης που υποστηρίζουν multi-threading, συν 8 αποδοτικούς πυρήνες που όχι. Εν τω μεταξύ, η AMD έχει το Zen 3 με 16 πυρήνες και 32 νήματα και αυτοί οι αριθμοί πυρήνων αναμένεται να αυξηθούν και επίσης να συνδυαστούν με

προσεγγίσεις big.LITTLE όπως μόλις έκανε η οικογένεια Core 12ης γενιάς.

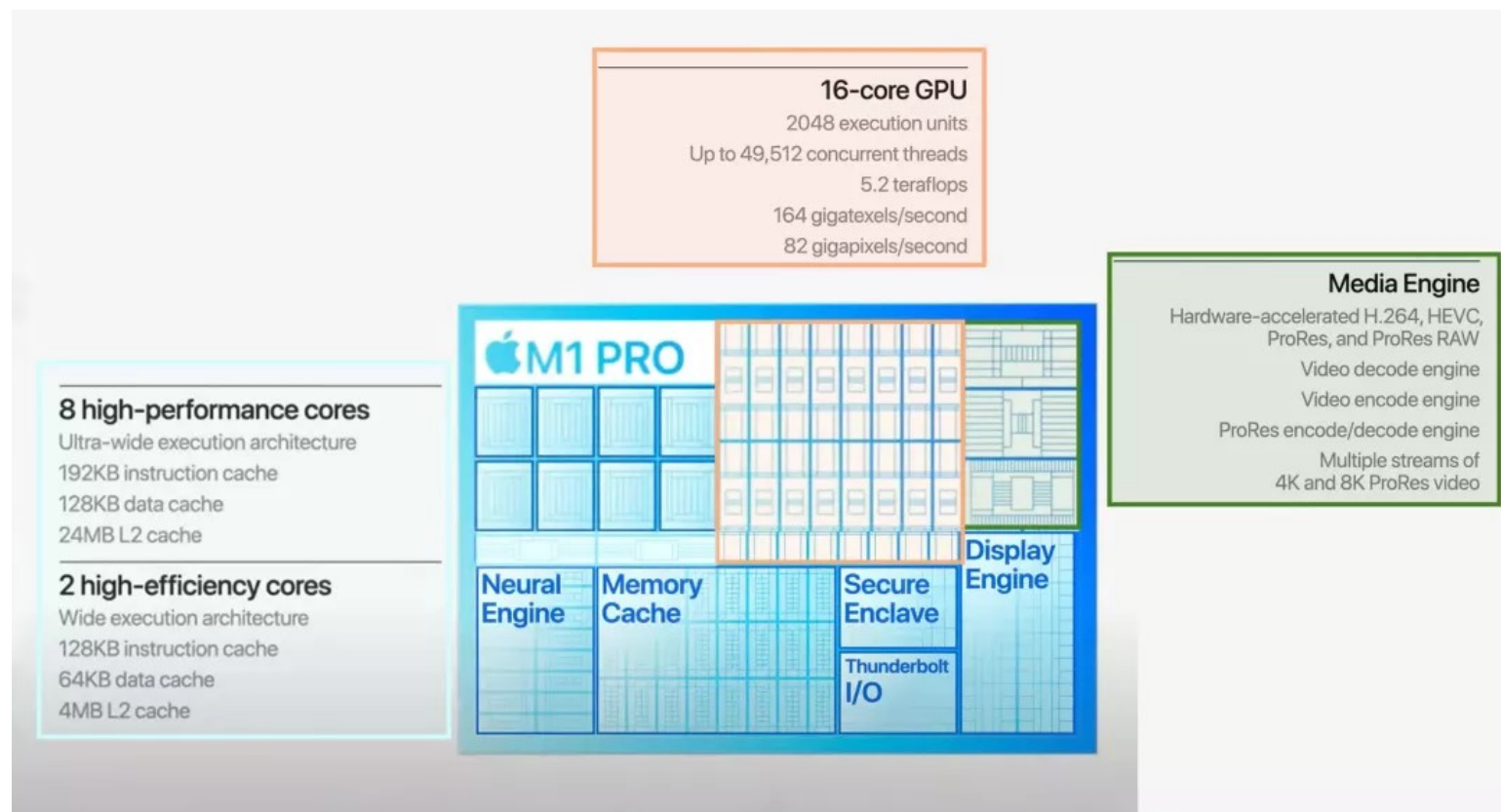
Εκτός από τους βασικούς αριθμούς, και οι δύο εταιρείες έχουν αυξήσει τα μεγέθη της κρυφής μνήμης, τα επίπεδα προσωρινής μνήμης, ενώ πρόσθεσαν και νέες επεκτάσεις ISA και βελτιστοποιήσεις αρχιτεκτονικής.

Επεξεργαστές για φορητούς υπολογιστές και όχι μόνο.

Όπως αποδεικνύεται πλήρως από το τσιπ Apple M1, οι καλά σχεδιασμένες CPU μπορούν να έχουν **τόσο αποδοτικά προφίλ κατανάλωσης ενέργειας, όσο και εξαιρετική απόδοση** και με την εισαγωγή της εγγενούς υποστήριξης **Arm** στα Windows 11, η Qualcomm και η Samsung είναι σίγουρο ότι θα καταβάλουν προσπάθεια να το πετύχουν αυτό.

Η υιοθέτηση αυτών των αποτελεσματικών στρατηγικών σχεδιασμού από τον τομέα **χαμηλής κατανάλωσης ενέργειας και κινητής τηλεφωνίας** δεν συνέβη από τη μια μέρα στην άλλη, αλλά ήταν το αποτέλεσμα της συνεχούς προσπάθειας κατασκευαστών CPU όπως η Intel, η Apple, η Qualcomm και η AMD να προσαρμόσουν τα τσιπ τους για να λειτουργούν σε φορητές συσκευές.

Επίσης η νέα τάση είναι η **ενσωμάτωση πολλών συσκευών στην κεντρική υπολογιστική μονάδα**, όπως, **πυρήνες γραφικών, φυσική μνήμη** κλπ



Η μνήμη RAM (memory)

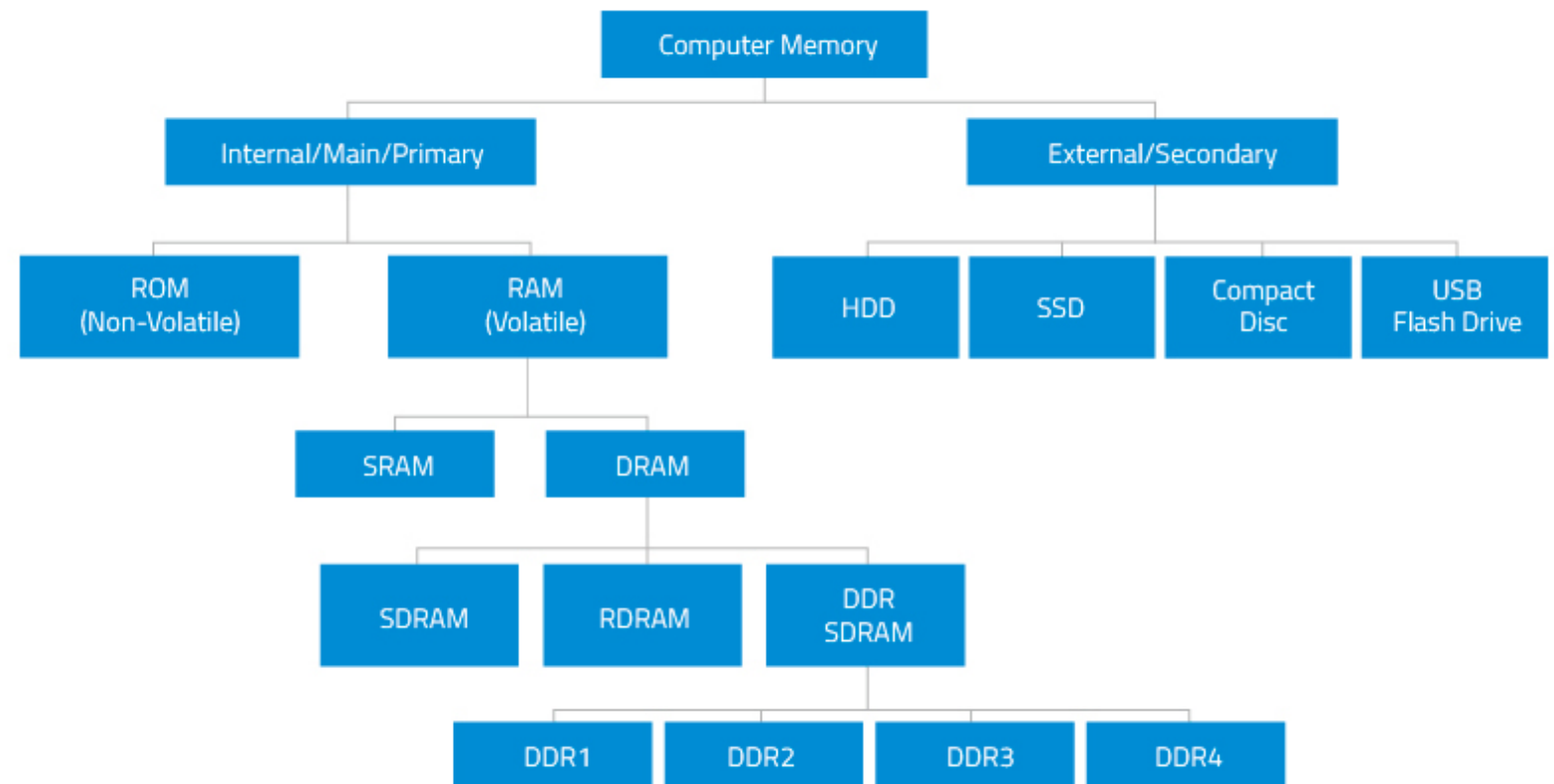
Μνήμη (Memory)

Η μνήμη αποτελεί έναν από τους πιο σημαντικούς πόρους για ένα λειτουργικό σύστημα. Υπάρχουν διαφορετικά επίπεδα μνήμης με διαφορετική ταχύτητα και χωρητικότητα:

Κύρια Μνήμη (RAM): Χρησιμοποιείται για την προσωρινή αποθήκευση δεδομένων και εντολών που χρειάζονται οι διεργασίες. Είναι γρήγορη, αλλά προσωρινή, καθώς τα δεδομένα χάνονται με τον τερματισμό του συστήματος.

Εικονική Μνήμη (Virtual Memory)

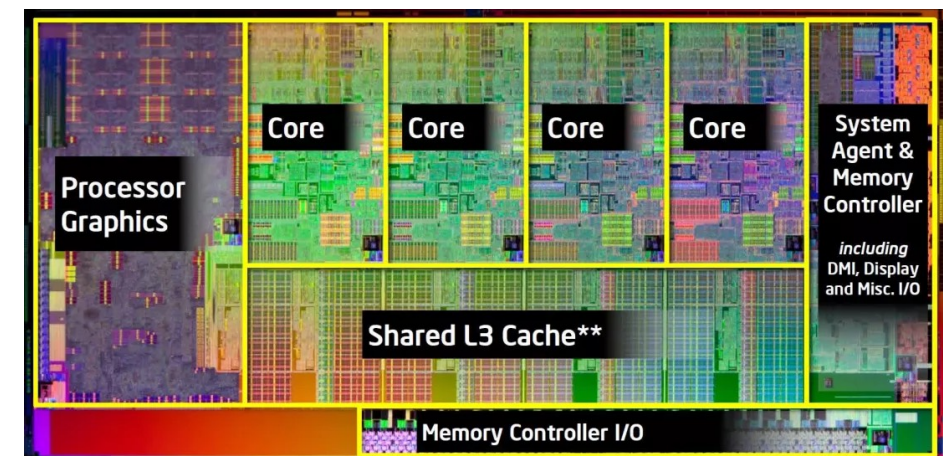
Η εικονική μνήμη είναι μια τεχνική που επιτρέπει στα λειτουργικά συστήματα να χρησιμοποιούν περισσότερο χώρο από αυτόν που υπάρχει στη RAM, αξιοποιώντας τον αποθηκευτικό χώρο του δίσκου. Αυτό βοηθά στη διαχείριση μεγάλων εφαρμογών και στην αποτελεσματικότερη χρήση της μνήμης.



Διάφορα είδη μνήμης

Μνήμη Cache:

Γρηγορότερη από τη RAM, χρησιμοποιείται από την CPU για την αποθήκευση δεδομένων που απαιτούνται άμεσα.



Σύγχρονη & ασύγχρονη

Υπάρχουν δύο κύριοι τύποι DRAM, η σύγχρονη και η ασύγχρονη. Η Asynchronous DRAM είναι ένας παλαιότερος τύπος DRAM που χρησιμοποιείται στους πρώτους προσωπικούς υπολογιστές. Ονομάζεται "ασύγχρονη" επειδή δεν είναι συγχρονισμένη με το εσωτερικό ρολόι του υπολογιστή.

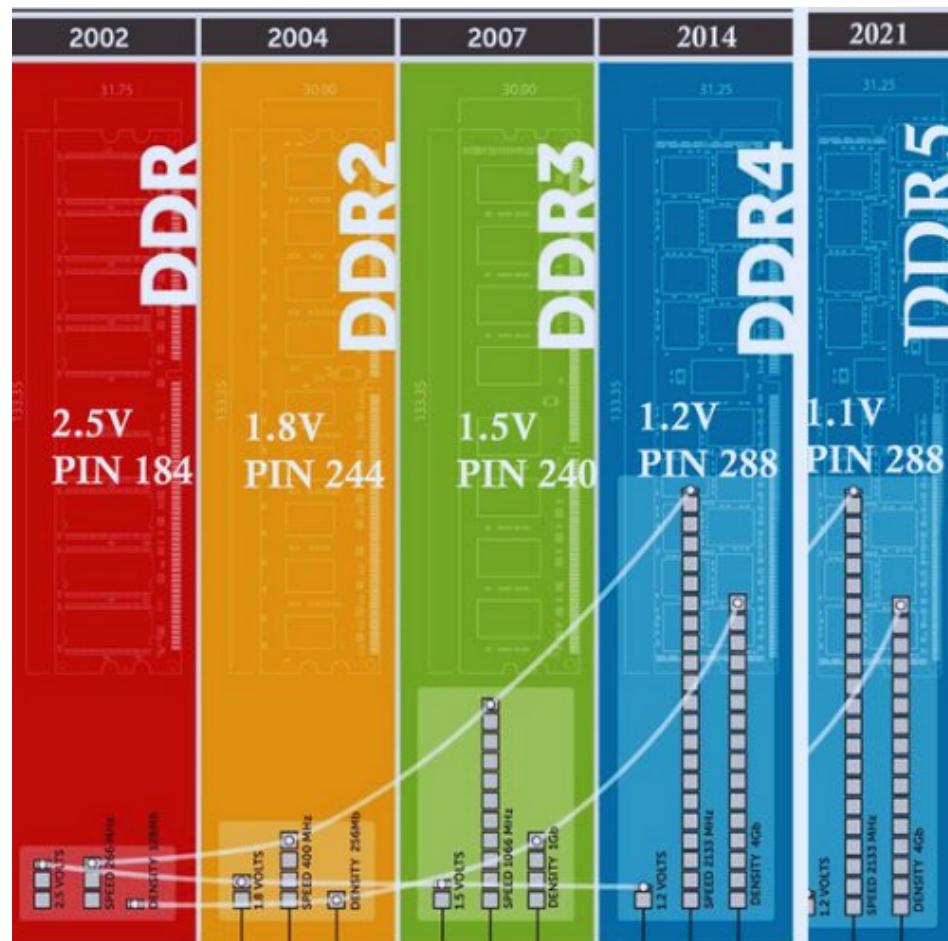
Η σύγχρονη DRAM ή "SDRAM" είναι ένας τύπος DRAM που συγχρονίζεται με το ρολόι του συστήματος. Η SDRAM

λειτουργεί πιο αποτελεσματικά επειδή λειτουργεί σύμφωνα με το συγχρονισμό του ρολογιού. Αυτό επιτρέπει ταχύτερους κωδικοποιητές και ταχύτερες λειτουργίες του υπολογιστή και βελτιώνει την αξιοπιστία. Χρησιμοποιείται πιο συχνά σε σύγχρονους υπολογιστές.

Η ασύγχρονη DRAM δεν χρησιμοποιείται συνήθως σε σύγχρονες εφαρμογές. Αυτό οφείλεται κυρίως στο γεγονός ότι η σύγχρονη DRAM είναι πολύ πιο αποτελεσματική και αξιόπιστη για

σύγχρονους υπολογιστές και έξυπνες συσκευές. Η ασύγχρονη DRAM ήταν μια πρώιμη μορφή DRAM που δεν συγχρονιζόταν με το ρολόι της CPU. Αν και αυτό λειτούργησε καλά εκείνη την εποχή, δεν θα λειτουργούσε αποτελεσματικά με το μεγαλύτερο εύρος ζώνης που απαιτείται για τους σημερινούς υπολογιστές. Από την εφεύρεση της συγχρονισμένης DRAM, η ασύγχρονη DRAM έχει μειωθεί σε χρήση και παραγωγή.

DDR



DDR SDRAM (1ης γενιάς). Το DDR SDRAM (1ης γενιάς), επίσης γνωστό ως Double Data Rate Synchronous Dynamic RAM, εφευρέθηκε το 2002. Αυτό το δημοφιλές νέο είδος SDRAM είναι μια ταχύτερη και πιο αποτελεσματική έκδοση του SDRAM. Αυτό το προϊόν είναι πιο δημοφιλές στους σύγχρονους υπολογιστές, επιτρέποντας μεγαλύτερες ταχύτητες από ποτέ.

DDR2 (2ης γενιάς). Το DDR2 (2η γενιά) εφευρέθηκε το 2004. είναι η επόμενη αναβάθμιση από το DDR SDRAM (1ης Γενιάς) και έφερε μεγαλύτερες ταχύτητες και περισσότερες δυνατότητες από τον προκατόχο του. Το DDR2 είναι ιδανικό για χρήση σε προσωπικούς υπολογιστές και φορητούς υπολογιστές.

DDR3 (3ης γενιάς). Το DDR3 επινοήθηκε το 2007 και χρησιμοποιείται ευρέως σε υπολογιστές παιχνιδιών. Αυτή η DRAM καταναλώνει 30% λιγότερη ενέργεια από τους προκατόχους της και μπορεί να μεταφέρει δεδομένα με διπλάσιο ρυθμό από το DDR2.

DDR4 (4ης γενιάς). Το DDR4 επινοήθηκε το 2014 και είναι το τρέχον πρότυπο για τους σύγχρονους υπολογιστές. Το DDR4 είναι εξαιρετικό για παιχνίδια, προγράμματα μηχανικής, κωδικοποίηση και λογισμικό επεξεργασίας βίντεο. Το DDR4 έχει μεγαλύτερες ταχύτητες, μειωμένη κατανάλωση ενέργειας και βελτιωμένη αξιοπιστία σε σύγκριση με τα DDR3 και DDR2.

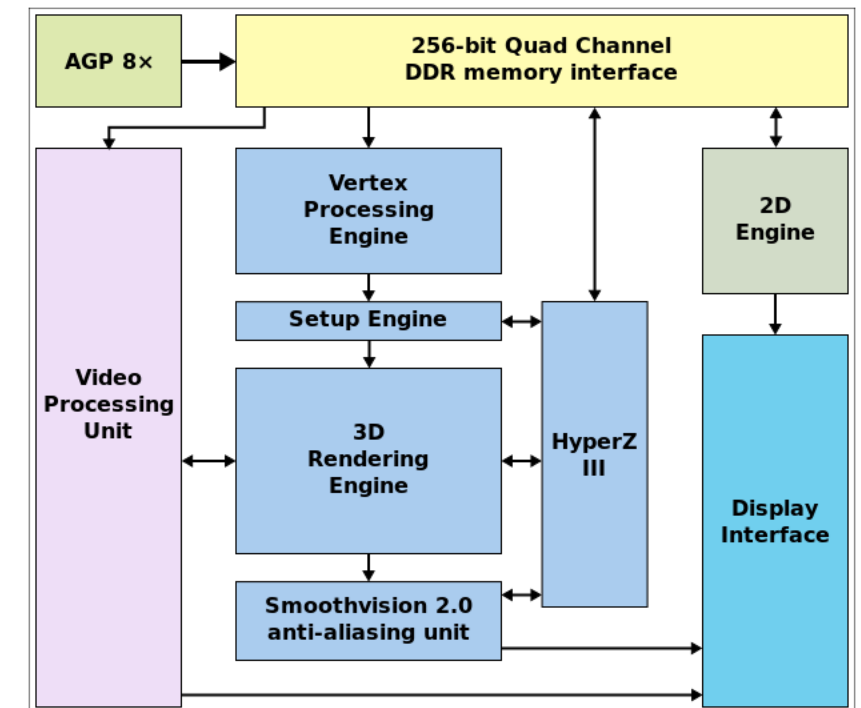
DDR5 (5ης γενιάς). Το DDR5 είναι η τελευταία γενιά DDR, που εφευρέθηκε το 2021. Το DDR5 είναι κατάλληλο για χρήστες που θέλουν να επιτύχουν τις υψηλότερες δυνατές ταχύτητες. Το DDR5 αυξάνει το εύρος ζώνης και μειώνει αποτελεσματικά την καθυστέρηση, ενώ παράλληλα μειώνει την κατανάλωση ενέργειας.

Κάρτες γραφικών

Μια κάρτα γραφικών (ονομάζεται επίσης κάρτα βίντεο , κάρτα οθόνης , προσαρμογέας γραφικών , κάρτα VGA/ VGA , προσαρμογέας βίντεο , προσαρμογέας οθόνης ή καθομιλουμένη GPU) είναι μια κάρτα επέκτασης υπολογιστή που δημιουργεί μια τροφοδοσία εξόδου γραφικών σε μια συσκευή προβολής όπως οθόνη . Οι κάρτες γραφικών ονομάζονται μερικές φορές διακριτές ή αποκλειστικές κάρτες γραφικών για να τονίσουν τη διάκρισή τους σε έναν ενσωματωμένο επεξεργαστή γραφικών στη μητρική πλακέτα ή στην κεντρική μονάδα επεξεργασίας (CPU). Μια μονάδα επεξεργασίας γραφικών (GPU) που εκτελεί τους απαραίτητους υπολογισμούς είναι το κύριο στοιχείο μιας κάρτας γραφικών, αλλά το ακρωνύμιο "GPU" χρησιμοποιείται μερικές φορές επίσης για να αναφέρεται λανθασμένα στην κάρτα γραφικών στο σύνολό της.

Μια **μονάδα επεξεργασίας γραφικών (GPU)**, που περιστασιακά ονομάζεται

επίσης **μονάδα οπτικής επεξεργασίας (VPU)**, είναι ένα εξειδικευμένο ηλεκτρονικό κύκλωμα που έχει σχεδιαστεί για να χειρίζεται γρήγορα και να αλλάζει τη μνήμη για να επιταχύνει τη δημιουργία εικόνων σε ένα buffer που προορίζεται για έξοδο σε μια οθόνη. Λόγω του μεγάλου βαθμού προγραμματιζόμενης υπολογιστικής πολυπλοκότητας για μια τέτοια εργασία, μια σύγχρονη κάρτα γραφικών είναι επίσης ένας υπολογιστής από μόνη της. Η χωρητικότητα μνήμης των περισσότερων σύγχρονων καρτών γραφικών κυμαίνεται από 2 έως και 32 GB από την τελευταία δεκαετία του 2010 και οι εφαρμογές για χρήση γραφικών γίνονται πιο ισχυρές και διαδεδομένες. Δεδομένου ότι η μνήμη βίντεο πρέπει να είναι προσβάσιμη από την GPU και το κύκλωμα οθόνης, χρησιμοποιεί συχνά ειδική μνήμη υψηλής ταχύτητας ή πολλαπλών θυρών, όπως VRAM , WRAM ,



Διάγραμμα της Radeon 9700 κάρτας γραφικών.

SGRAM κ.λπ. Γύρω στο 2003, η μνήμη βίντεο βασιζόταν συνήθως στην τεχνολογία DDR . Κατά τη διάρκεια και μετά από εκείνο το έτος, οι κατασκευαστές κινήθηκαν προς τα DDR2 , GDDR3 , GDDR4 , GDDR5 , GDDR5X και GDDR6 . Ο αποτελεσματικός ρυθμός ρολογιού μνήμης στις σύγχρονες κάρτες είναι γενικά μεταξύ 2 και 15GHz.

Θύρες γραφικών

Συστοιχία γραφικών βίντεο (**VGA**) (DE-15). Γνωστό και ως D-sub , το VGA είναι ένα αναλογικό πρότυπο που υιοθετήθηκε στα τέλη της δεκαετίας του 1980 και σχεδιάστηκε για οθόνες CRT, που ονομάζεται επίσης σύνδεσμος VGA . Σήμερα, η αναλογική διεπαφή VGA χρησιμοποιείται για αναλύσεις βίντεο υψηλής ευκρίνειας, συμπεριλαμβανομένων 1080p και άνω.

Ψηφιακή οπτική διεπαφή (**DVI**). Το Digital Visual Interface είναι ένα ψηφιακό πρότυπο σχεδιασμένο για οθόνες, όπως επίπεδες οθόνες (οθόνες LCD , οθόνες πλάσματος, οθόνες ευρείας τηλεόρασης υψηλής ευκρίνειας) και βιντεοπροβολείς. Υπήρχαν επίσης μερικές σπάνιες οθόνες CRT

υψηλής τεχνολογίας που χρησιμοποιούσαν DVI. Αποφεύγει την παραμόρφωση της εικόνας και τον ηλεκτρικό θόρυβο, που αντιστοιχεί κάθε pixel από τον υπολογιστή σε ένα pixel οθόνης, χρησιμοποιώντας την εγγενή του ανάλυση . Αξίζει να σημειωθεί ότι οι περισσότεροι κατασκευαστές περιλαμβάνουν υποδοχή DVI- I , που επιτρέπει, μέσω απλού προσαρμογέα, τυπική έξοδο σήματος RGB σε μια παλιά οθόνη CRT ή LCD με είσοδο VGA.

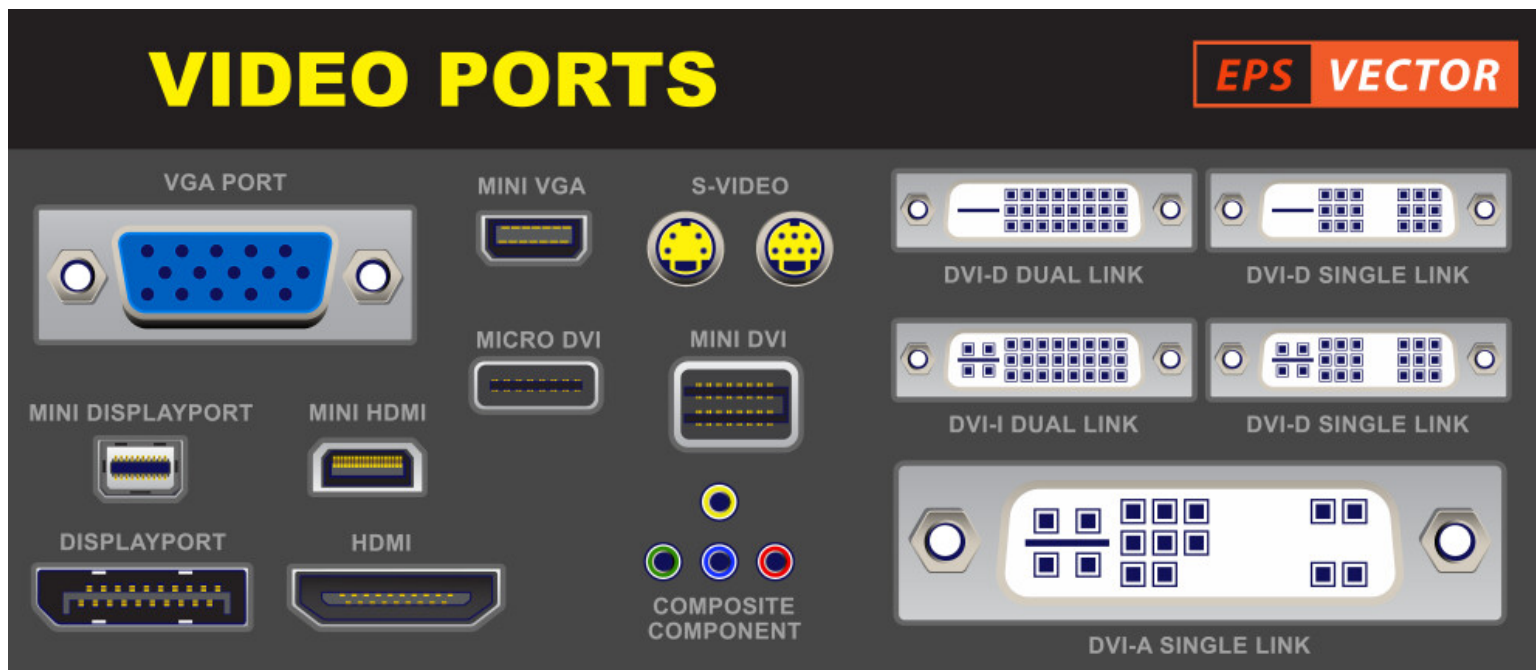
Διασύνδεση πολυμέσων υψηλής ευκρίνειας (**HDMI**).

Το HDMI είναι μια συμπαγής διεπαφή ήχου/εικόνας για τη μεταφορά μη συμπιεσμένων δεδομένων βίντεο και συμπιεσμένων/μη συμπιεσμένων

δεδομένων ψηφιακού ήχου από μια συσκευή συμβατή με HDMI ("η συσκευή πηγής") σε μια συμβατή ψηφιακή συσκευή ήχου, οθόνη υπολογιστή , βιντεοπροβολέα ή ψηφιακή τηλεόραση. Το HDMI είναι μια ψηφιακή αντικατάσταση για τα υπάρχοντα πρότυπα αναλογικού βίντεο.

DP. Το DisplayPort είναι μια διεπαφή ψηφιακής οθόνης που αναπτύχθηκε από την Ένωση Προτύπων ηλεκτρονικών βίντεο (VESA). Η διεπαφή χρησιμοποιείται κυρίως για τη σύνδεση μιας πηγής βίντεο σε μια συσκευή προβολής , όπως μια οθόνη υπολογιστή , αν και μπορεί επίσης να χρησιμοποιηθεί για τη μετάδοση ήχου, USB και άλλων μορφών δεδομένων.

USB-C. Η λειτουργία Alt επιτρέπει στις θύρες USB-C να μεταφέρουν δεδομένα/βίντεο μέσω πολλαπλών διεπαφών, συμπεριλαμβανομένης της διεπαφής βίντεο DisplayPort. Οι θύρες USB-C που μπορούν να μεταφέρουν σήματα DisplayPort ονομάζονται είτε USB-C DisplayPort είτε DP Alt Mode. Επιτρέπουν να συνδέονται πηγές βίντεο (π.χ. υπολογιστές, συσκευές αναπαραγωγής Blu-Ray, κ.λπ.) και συσκευές προβολής (π.χ. τηλεοράσεις, προβολείς κ.λπ., που υποστηρίζουν το DisplayPort , μεταξύ τους μέσω των θυρών USB-C για μετάδοση βίντεο υψηλής ευκρίνειας.



Αναλύσεις οθόνης

Η ανάλυση οθόνης ή οι τρόποι εμφάνισης μιας ψηφιακής τηλεόρασης , οθόνης υπολογιστή ή άλλης συσκευής οθόνης είναι ο αριθμός των διακριτών εικονοστοιχείων σε κάθε διάσταση που μπορούν να εμφανιστούν. Συνήθως αναφέρεται ως πλάτος × ύψος , με τις μονάδες σε εικονοστοιχεία: για παράδειγμα, 1024×768 σημαίνει ότι το πλάτος είναι 1024 εικονοστοιχεία και το ύψος είναι 768 εικονοστοιχεία. Αυτό το παράδειγμα θα λέγεται κανονικά ως "χίλια είκοσι τέσσερα επί επτά εξήντα οκτώ".

Μία χρήση του όρου ανάλυση οθόνης ισχύει για οθόνες σταθερής διάταξης pixel, όπως πάνελ οθόνης πλάσματος (PDP), οθόνες υγρών κρυστάλλων (LCD), προβολείς ψηφιακής επεξεργασίας φωτός (DLP), οθόνες OLED και παρόμοιες τεχνολογίες, και είναι απλά ο φυσικός αριθμός στηλών και σειρών των pixel που δημιουργούν την οθόνη (π.χ. 1920×1080).

Για οθόνες συσκευών όπως τηλέφωνα, τάμπλετ, οθόνες και τηλεοράσεις, η χρήση του όρου ανάλυση οθόνης όπως ορίζεται παραπάνω είναι μια εσφαλμένη ονομασία. Ο όρος ανάλυση οθόνης χρησιμοποιείται συνήθως για να σημαίνει πλήθος εικονοστοιχείων , τον μέγιστο αριθμό

εικονοστοιχείων σε κάθε διάσταση (π.χ. 1920×1080), που δεν λέει τίποτα για την πυκνότητα εικονοστοιχείων της οθόνης στην οποία σχηματίζεται η εικόνα. Η ανάλυση αναφέρεται σωστά στην πυκνότητα των εικονοστοιχείων , τον αριθμό των εικονοστοιχείων ανά μονάδα απόστασης ή περιοχής και όχι τον συνολικό αριθμό των εικονοστοιχείων.

Αναλύσεις:

Τηλεόραση τυπικής ευκρίνειας (**SDTV**):

480i (ψηφιακό πρότυπο συμβατό με NTSC που χρησιμοποιεί δύο πλεγμένα πεδία των 243 γραμμών το καθένα)

576i (ψηφιακό πρότυπο συμβατό με PAL που χρησιμοποιεί δύο πλεγμένα πεδία των 288 γραμμών το καθένα)

Τηλεόραση βελτιωμένης ευκρίνειας (**EDTV**):

480p (προοδευτική σάρωση 720×480)

576p (προοδευτική σάρωση 720×576)

Τηλεόραση υψηλής ευκρίνειας (**HDTV**):

720p (προοδευτική σάρωση 1280×720)

1080i (1920×1080 χωρισμένο σε δύο πλεγμένα πεδία των 540 γραμμών)



Διάφορα μεγέθη οθονών

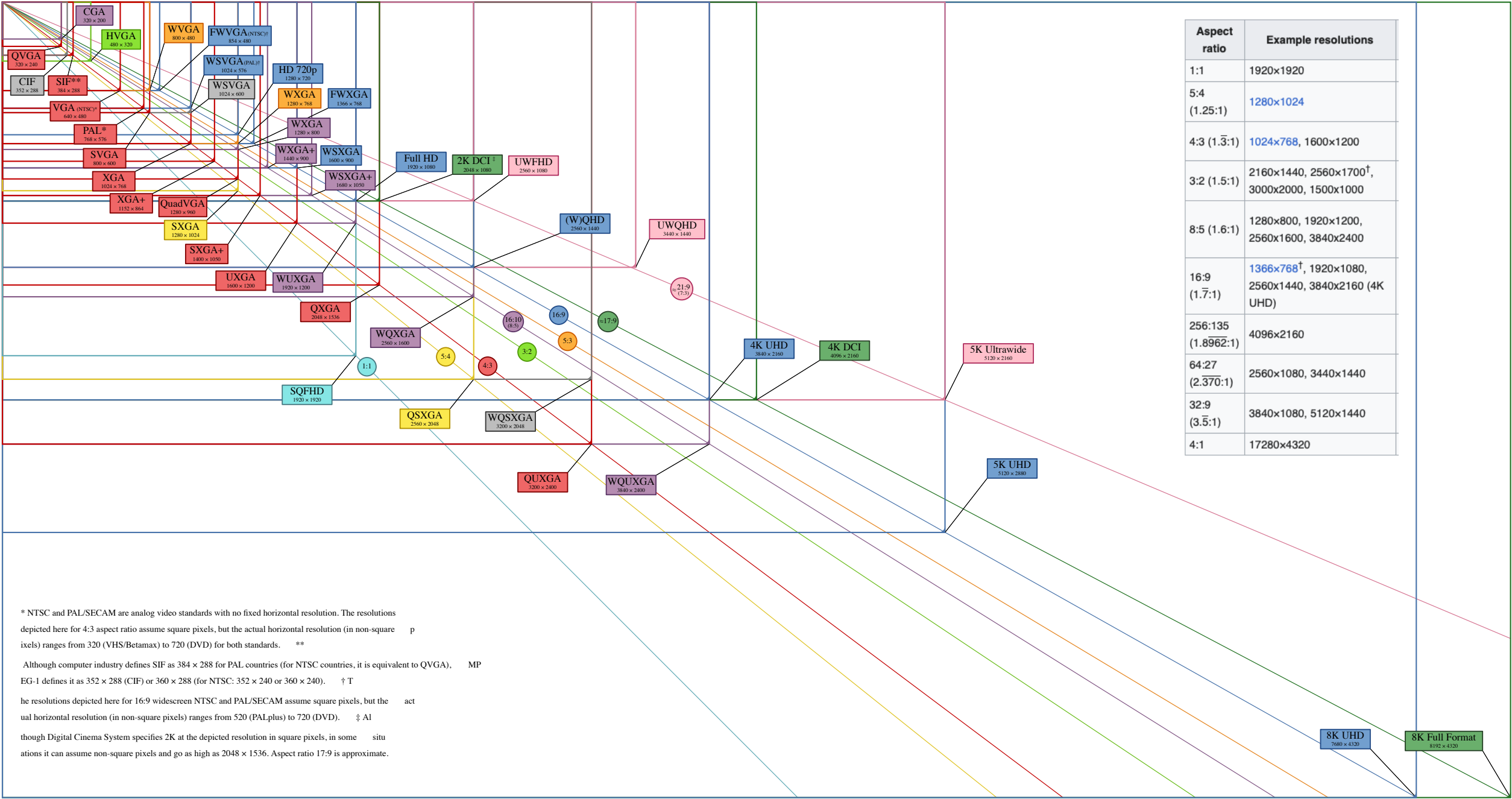
1080p (προοδευτική σάρωση 1920×1080)

Τηλεόραση εξαιρετικά υψηλής ευκρίνειας (**UHDTV**):

4K UHD (προοδευτική σάρωση 3840×2160)

8K UHD (προοδευτική σάρωση 7680×4320)

Αναλύσεις οθόνης



Αναλύσεις βάσει του λόγου (αναλογίας) διαστάσεων οθόνης

Χώρος μνήμης για βίντεο 4K 32bit

Ας υποθέσουμε ότι η ανάλυση της οθόνης είναι 4K (3840 X 2160) και θέλω να προβάλω βίντεο με βάθος χρώματος 32bits. Αν υποθέσουμε ότι θέλω να ανεβάσω στη μνήμη ένα βίντεο 10" σε πλήρη ανάλυση, πόση μνήμη θα χρειαστώ, αν θεωρήσουμε ότι δεν υπάρχει καμία συμπίεση δεδομένων;

Για να υπολογίσουμε τον απαιτούμενο χώρο μνήμης για ένα βίντεο 10 δευτερολέπτων 4K (3840 x 2160) με βάθος χρώματος 32 bits, ας λάβουμε υπόψη τα ακόλουθα:

1. Συνολική ανάλυση:

$3840 \text{ (πλάτος)} \times 2160 \text{ (ύψος)} = 8.294.400 \text{ pixels}$

2. Βάθος χρώματος:

$32 \text{ bits/pixel (4 bytes/pixel)}$

3. Χρόνος:

10 δευτερόλεπτα

4. Συχνότητα καρέ:

Υποθέτουμε τυπική συχνότητα καρέ 30 fps (frames per second)

Υπολογισμός:

Συνολικός αριθμός pixels ανά καρέ: 8.294.400 pixels/frame

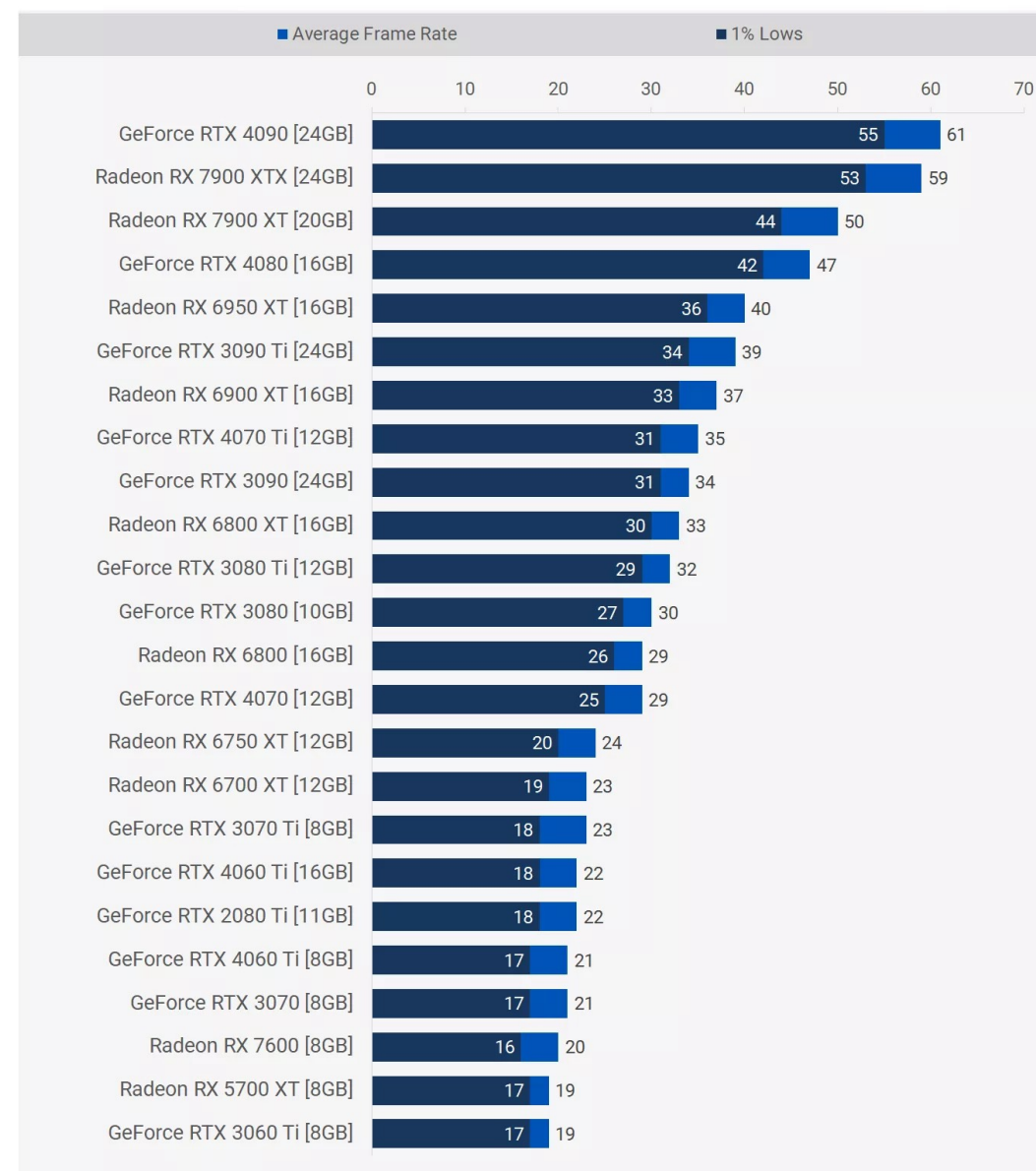
Συνολικός αριθμός bytes ανά καρέ: 8.294.400 pixels/frame * 4 bytes/pixel = 33.177.600 bytes/frame

Συνολικός αριθμός καρέ σε 10 δευτερόλεπτα: 30 frames/second * 10 seconds = 300 frames

Συνολικός απαιτούμενος χώρος μνήμης: 33.177.600 bytes/frame * 300 frames = 9.953.280.000 bytes

Αποτέλεσμα:

Για να αποθηκεύσετε ένα βίντεο 10 δευτερολέπτων 4K 32bit με τυπική συχνότητα καρέ 30 fps, θα χρειαστείτε περίπου 9,95 GB μνήμης.



Συστήματα Αποθήκευσης (Storage Systems)

Υπάρχουν τρεις τύποι αποθήκευσης δεδομένων υπολογιστή που παρατίθενται παρακάτω.

1. Πρωτογενής αποθήκευση

Η κύρια μνήμη, όπως είναι επίσης γνωστή, είναι άμεσα προσβάσιμη από την CPU (Κεντρική Μονάδα Επεξεργασίας) του υπολογιστή. Είναι μια άμεσης πρόσβασης μνήμη και είναι για προσωρινή αποθήκευση. Αυτό σημαίνει ότι η μνήμη διαγράφεται όταν απενεργοποιηθεί ή επανεκκινηθεί η συσκευή του υπολογιστή. Αυτή η συσκευή αποθήκευσης χωράει εσωτερικά στον υπολογιστή. Για παράδειγμα, RAM και Cache Memory (και οι δύο είναι εσωτερική μνήμη υπολογιστή). Παρέχει γρήγορη εξυπηρέτηση όσον αφορά την πρόσβαση σε αρχεία δεδομένων. Είναι μικρή σε μέγεθος.

2. Δευτερεύουσα αποθήκευση

Η δευτερεύουσα συσκευή αποθήκευσης δεν είναι άμεσα προσβάσιμη από την CPU. Παραδείγματα αποτελούν συσκευές αποθήκευσης USB, μονάδα οπτικού δίσκου, δισκέτα, σκληρό δίσκο, SSD, μνήμη flash, κ.λπ. Μια δευτερεύουσα συσκευή αποθήκευσης αποθηκεύει δεδομένα μόνιμα, μέχρι αυτά να αφαιρεθούν από εξωτερικό παράγοντα. Περιλαμβάνει τόσο εσωτερική όσο και εξωτερική μνήμη.

Δεν είναι άμεσης προσπέλασης και έχει μεγαλύτερη χωρητικότητα αποθήκευσης από την κύρια αποθήκευση.

3. Τριτοβάθμια μνήμη

Ο πρωταρχικός στόχος του τριτογενούς επιπέδου αποθήκευσης είναι να παρέχει τεράστια χωρητικότητα αποθήκευσης με φθινό κόστος. Η τριτοβάθμια αποθήκευση αποτελείται από αρχεία



Hard disc



DVD



CD-ROM



Pendrive



Floppy Disk



Memory Card

Συσκευές αποθήκευσης δεδομένων.

δεδομένων μεγάλης χωρητικότητας που μπορεί να φιλοξενήσουν μεγάλο αριθμό αφαιρουμένων μέσων, όπως ταινίες ή οπτικούς δίσκους.

Σκληροί και οπτικοί δίσκοι

Οι μονάδες σκληρού δίσκου (HDD) δεν ήταν κάτι καινούργιο το 1982, αλλά ποτέ ένας σκληρός δίσκος δεν εγκαταστάθηκε στον πρώτο υπολογιστή της IBM. Αντίθετα, ο κόσμος περίμενε το μοντέλο δεύτερης γενιάς eXTension (XT). Το PC XT περιλάμβανε εγκατεστημένο έναν τυπικό σκληρό δίσκο 10MB. Η μονάδα δίσκου για να εγκατασταθεί στο σύστημα, απαιτούσε ένα νέο τροφοδοτικό και μια νέα ενημέρωση του BIOS.

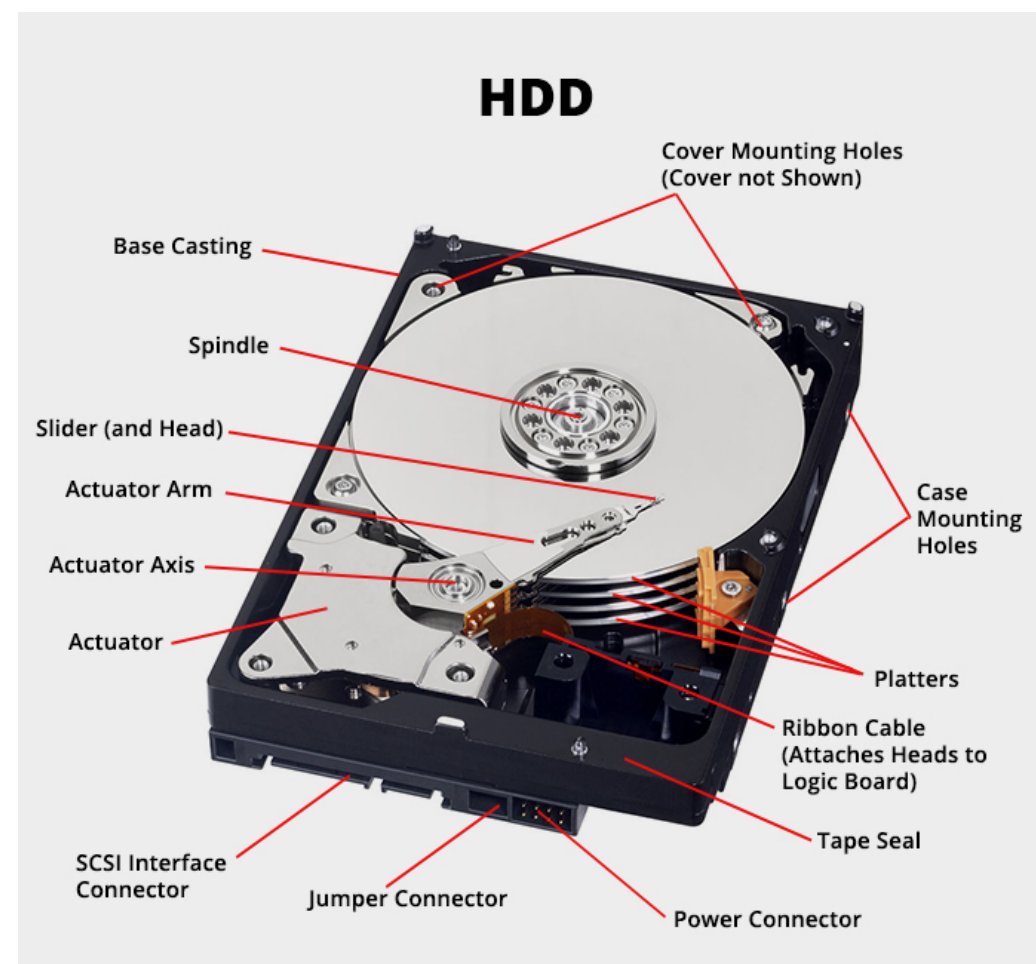
Ο πρώτος σκληρός δίσκος του IBM PC ήταν ο Seagate Technology Model ST-412 . Η διεπαφή μεταξύ αυτού και της μητρικής πλακέτας έγινε το de facto πρότυπο διαύλου μονάδας δίσκου για αρκετά χρόνια.

Σήμερα, μπορούμε να βρούμε εσωτερικούς σκληρούς δίσκους 20 terabyte (TB) στην αγορά, όπως το Seagate Exos X20 Αυτή η εταιρεία από μόνη της έχει στείλει πλήρη χωρητικότητα αποθήκευσης 3 zettabyte σκληρού δίσκου έως το 2021 —που ισοδυναμεί με 150.000.000 σκληρούς δίσκους με 20 TB ο καθένας. Η μνήμη μόνο για ανάγνωση που άλλαξε τον κόσμο. Ο πλήρως οπτικός και ψηφιακός συμπαγής δίσκος γεμάτος δεδομένα χωρούσε έως και 650 MB σε 1,2 mm πολυανθρακικό πλαστικό με ανακλαστική επιφάνεια αλουμινίου. Θα μπορούσαν να διαβαστούν μόνο με λέιζερ. Το CD-ROM έγινε το πρότυπο για τη διανομή λογισμικού και βιντεοπαιχνιδιών στα τέλη της δεκαετίας του 1980 και παρέμεινε μέχρι τη δεκαετία του '90. (Τα CD μουσικής είναι

παρόμοια, αλλά χρησιμοποιούν διαφορετική μορφή , αν και οι μονάδες CD υπολογιστή θα μπορούσαν τελικά να τα διαβάσουν επίσης.)

Το μόνο μειονέκτημα του CD-ROM είναι ότι είναι μνήμη μόνο για ανάγνωση (είναι ακριβώς εκεί στο όνομα). Οι χρήστες δεν μπορούσαν να γράψουν δεδομένα σε αυτό. Ωστόσο, αυτό τα έκανε ιδανικά για τους διανομείς λογισμικού και παιχνιδιών που τους άρεσε ότι ήταν εύκολο να προστατευτεί από αντιγραφή.

Όσο πιο γρήγορα η μονάδα περιστρέφεται το CD-ROM, τόσο πιο γρήγορα γίνεται η πρόσβαση στα δεδομένα. Το βασικό πρότυπο του 1x ήταν περίπου 150 Kb ανά δευτερόλεπτο, αλλά τελικά, οι μονάδες δίσκου έφτασαν το 52x ή ακόμα και το 72x, αλλά με ορισμένες προειδοποιήσεις για τον φυσικό κόσμο. Το εγγράψιμο με συμπαγή δίσκο (CD-R) ονομαζόταν αρχικά CD-Write-Once και χρησιμοποιεί λίγη από την ίδια τεχνολογία με την προηγούμενη μαγνητο-οπτική μονάδα - τη δυνατότητα εγγραφής των δεδομένων σας σε έναν δίσκο μία φορά μόνο για



Αγγίξτε ή κάντε κλικ σε αυτό το δεσμευτικό θέσης και αρχίστε να πληκτρολογείτε για να αντικαταστήσετε τη λεζάντα εικόνας, ή απενεργοποιήστε τη δυνατότητα στα στοιχεία ελέγχου Μορφής.

δημιουργία αντιγράφων ασφαλείας ή διανομή . Θα μπορούσατε να γράψετε σε CD-R σε διαμόρφωση αρχείων ήχου ("Κόκκινη έκδοση") που περιέχει έως και 80 λεπτά μουσικής ή μορφή δεδομένων ("Κίτρινη έκδοση") με 700 MB πληροφοριών και θα λειτουργούσαν σε κανονικά προγράμματα αναπαραγωγής CD ή CD- H ROM οδηγεί τις περισσότερες φορές.

Δίαυλοι επικοινωνίας

Οι δίαυλοι είναι τα μονοπάτια που επιτρέπουν τη μεταφορά δεδομένων μεταξύ των διαφόρων μερών ενός συστήματος:

Δίαυλος Συστήματος (System Bus):

Συνδέει τον επεξεργαστή με τη μνήμη και τα περιφερειακά.

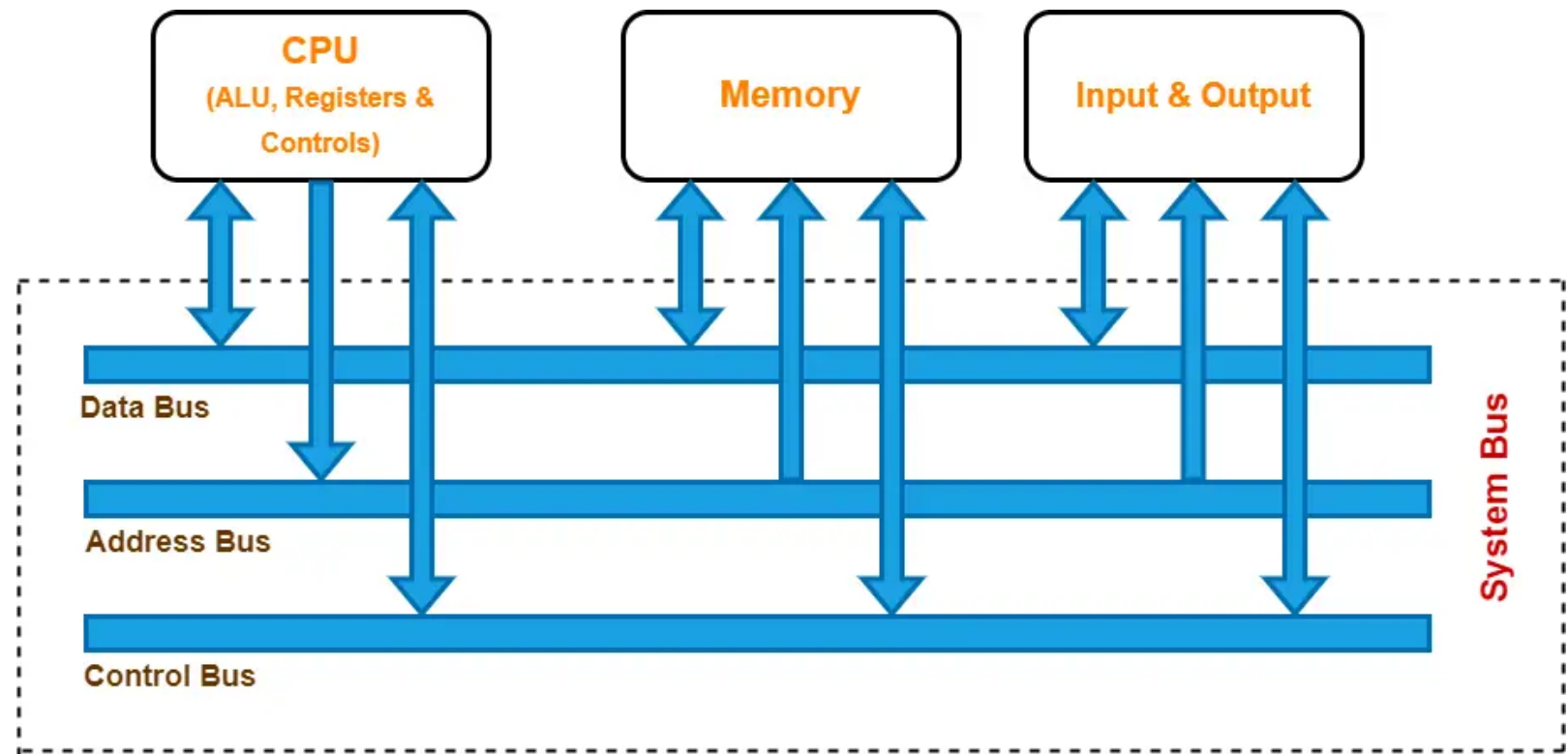
Δίαυλος Διεύθυνσης (Address Bus) και

Δίαυλος Δεδομένων (Data Bus):

Μεταφέρουν διευθύνσεις και δεδομένα αντίστοιχα.

Όταν η CPU επιθυμεί να αποκτήσει πρόσβαση σε μια συγκεκριμένη θέση κύριας μνήμης, στέλνει αυτή τη διεύθυνση στη μνήμη του διαύλου διευθύνσεων. Τα δεδομένα σε αυτή τη θέση επιστρέφονται στη συνέχεια στην CPU στο δίαυλο δεδομένων. Τα σήματα ελέγχου αποστέλλονται κατά μήκος του διαύλου ελέγχου.

Ο δίαυλος ελέγχου είναι δίαυλος διπλής κατεύθυνσης (τα σήματα μπορούν να μεταφερθούν και προς τις δύο κατευθύνσεις). Οι δίαυλοι δεδομένων και διευθύνσεων μοιράζονται όλα τα στοιχεία του συστήματος. Επομένως, πρέπει να



παρέχονται γραμμές ελέγχου για να διασφαλίζεται ότι η πρόσβαση και η χρήση των δεδομένων και των διευθύνσεων από τα διάφορα στοιχεία του συστήματος δεν οδηγεί σε σύγκρουση.

Ο δίαυλος δεδομένων, που συνήθως αποτελείται από 8, 16, 32, 64 ξεχωριστές γραμμές παρέχει μια αμφίδρομη διαδρομή για τη μετακίνηση δεδομένων και οδηγιών μεταξύ των στοιχείων του συστήματος. Το πλάτος του διαύλου δεδομένων είναι βασικός παράγοντας για τον καθορισμό της συνολικής απόδοσης

του συστήματος. Για παράδειγμα, εάν ο δίαυλος δεδομένων έχει πλάτος 16 bit και κάθε εντολή έχει μήκος 32 bit, τότε ο επεξεργαστής πρέπει να έχει πρόσβαση στην κύρια μνήμη δύο φορές μόνο για να ανακτήσει την εντολή.

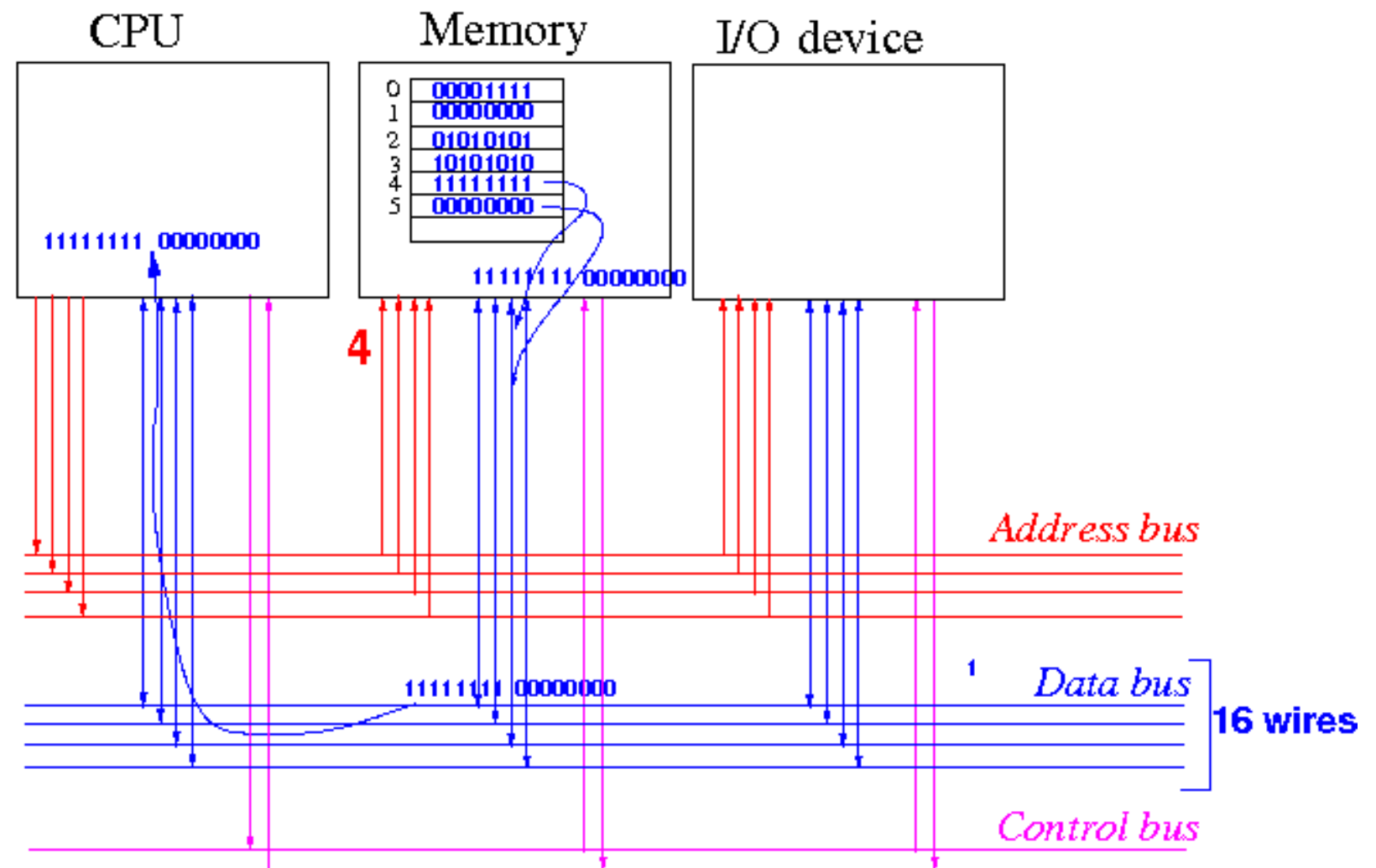
Η ταχύτητα των διαύλων είναι κρίσιμη για την απόδοση του συστήματος, ειδικά σε συστήματα υψηλής απόδοσης όπως αυτά που βρίσκονται στο νέφος.

Δίαυλος διεύθυνσης

Η μνήμη χωρίζεται εσωτερικά σε μονάδες που ονομάζονται λέξεις. Μια λέξη είναι μια ομάδα ψηφίων σταθερού μεγέθους, συνήθως 8, 16, 32 ή 64 bit, τα οποία αντιμετωπίζονται ως μονάδα από τον επεξεργαστή και διαφορετικοί τύποι επεξεργαστών έχουν διαφορετικά μεγέθη λέξεων. Κάθε λέξη στη μνήμη έχει τη δική της συγκεκριμένη διεύθυνση.

Όταν ο επεξεργαστής επιθυμεί να διαβάσει μια λέξη δεδομένων από τη μνήμη, τοποθετεί πρώτα τη διεύθυνση της επιθυμητής λέξης στο **δίαυλο διευθύνσεων**. Το πλάτος του διαύλου διευθύνσεων καθορίζει τη μέγιστη δυνατή χωρητικότητα μνήμης του συστήματος.

Για παράδειγμα, εάν ο δίαυλος διευθύνσεων αποτελούνταν μόνο από 8 γραμμές, τότε η μέγιστη διεύθυνση που θα μπορούσε να μεταδώσει θα ήταν 255 ή 11111111 (σε δυαδικό), δίνοντας μέγιστη χωρητικότητα μνήμης 256 (συμπεριλαμβανομένης της διεύθυνσης 0).



Ένα σύστημα με ένα δίαυλο διευθύνσεων 32 bit μπορεί να απευθύνεται σε 2^{32} (4 294 967 296) θέσεις μνήμης δίνοντας διευθυνσιοδοτούμενο χώρο μνήμης 4 Gb.

Ο δίαυλος διευθύνσεων χρησιμοποιείται επίσης για τη διευθυνσιοδότηση των θυρών εισόδου/εξόδου κατά τις λειτουργίες εισόδου/εξόδου.

Ο επί 3 Δεκαετίες διαύλος PCI

Το PCI Express (Peripheral Component Interconnect Express), επίσημα συντομογραφία PCIe ή PCI-e, είναι ένα πρότυπο διαύλου επέκτασης σειριακών υπολογιστών υψηλής ταχύτητας , σχεδιασμένο για να αντικαταστήσει τα παλαιότερα πρότυπα διαύλου PCI , PCI-X και AGP . Είναι η κοινή διεπαφή μητρικής πλακέτας για κάρτες γραφικών προσωπικών υπολογιστών , κάρτες ήχου , προσαρμογείς κεντρικού υπολογιστή μονάδας σκληρού δίσκου , SSD , Wi-Fi και συνδέσεις υλικού Ethernet. Το PCIe έχει πολλές βελτιώσεις σε σχέση με τα παλαιότερα πρότυπα, όπως υψηλότερη μέγιστη απόδοση διαύλου συστήματος, χαμηλότερο αριθμό ακίδων I/O και μικρότερο φυσικό αποτύπωμα, καλύτερη κλιμάκωση απόδοσης για συσκευές διαύλου, πιο λεπτομερή εντοπισμό σφαλμάτων και μηχανισμό αναφοράς (Advanced Error Reporting, AER), και εγγενής λειτουργία εναλλαγής hot-swap . Πιο πρόσφατες αναθεωρήσεις του προτύπου PCIe παρέχουν υποστήριξη υλικού για εικονικοποίηση I/O .

Η ηλεκτρική διεπαφή PCI Express μετριέται με τον αριθμό των ταυτόχρονων λωρίδων. (Μια λωρίδα είναι μια ενιαία γραμμή αποστολής/λήψης δεδομένων, ανάλογη με έναν "δρόμο μιας λωρίδας" που έχει μία λωρίδα κυκλοφορίας και προς τις δύο κατευθύνσεις.) Η διεπαφή χρησιμοποιείται επίσης σε μια ποικιλία άλλων προτύπων – κυρίως τη “διασύνδεση κάρτας επέκτασης φορητού υπολογιστή” που ονομάζεται ExpressCard . Χρησιμοποιείται επίσης στις διεπαφές αποθήκευσης SATA Express , U.2 (SFF-8639) και M.2 .

Οι προδιαγραφές μορφής διατηρούνται και αναπτύσσονται από την PCI-SIG (PCI Special Interest Group) – έναν όμιλο περισσότερων από 900 εταιρειών που διατηρεί επίσης τις συμβατικές προδιαγραφές PC.

Στο PCIe (Peripheral Component Interconnect Express), οι αριθμοί μετά το "x" (x1, x4, x8, x16, x32, x64 κλπ) υποδεικνύουν τον αριθμό λωρίδων δεδομένων που διατίθενται για την

επικοινωνία μεταξύ μιας συσκευής και του chipset της μητρικής πλακέτας. Ο αριθμός λωρίδων δεδομένων επηρεάζει άμεσα την ταχύτητα μεταφοράς δεδομένων και,



Διάφορες υποδοχές σε **μητρική πλακέτα υπολογιστή** , από πάνω προς τα κάτω:

- PCI Express x4
- PCI Express x16
- PCI Express x1
- PCI Express x16
- **Συμβατικό PCI** (32-bit, 5 V)

Διάφορα είδη διαύλου PCIe

Θύρες Επικοινωνίας: Ιστορία και Χαρακτηριστικά

Οι θύρες επικοινωνίας αποτελούν ζωτικό κομμάτι των υπολογιστών, επιτρέποντας την σύνδεση με περιφερειακές συσκευές και την ανταλλαγή δεδομένων. Ας δούμε μερικές από τις πιο κοινές θύρες και την ιστορία τους:

1. Σειριακή θύρα (RS-232):

Ιστορία: Η σειριακή θύρα, βασισμένη στο πρότυπο RS-232, κυκλοφόρησε στα τέλη της δεκαετίας του 1960 και υπήρξε η κυρίαρχη θύρα σύνδεσης για περιφερειακές συσκευές για δεκαετίες.

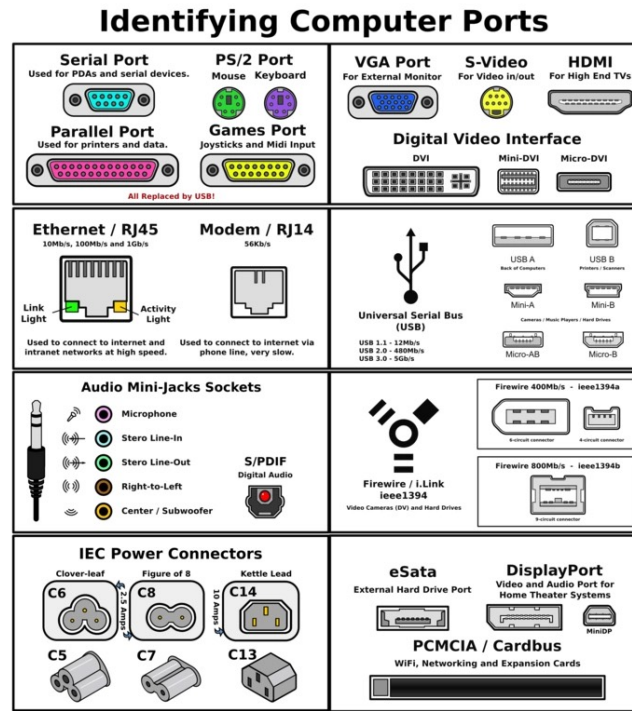
Χαρακτηριστικά: Χρησιμοποιεί αμφίδρομη επικοινωνία, μεταφέροντας δεδομένα ένα bit τη φορά.

Πλεονεκτήματα: Απλή κατασκευή, αξιοπιστία, μεγάλο μήκος καλωδίου.

Μειονεκτήματα: Χαμηλή ταχύτητα (έως 115 kbps), ογκώδης σύνδεση.

Σήμερα: Η χρήση της σειριακής θύρας έχει μειωθεί σημαντικά, **αντικαταστάθηκε από USB** και άλλες ταχύτερες διεπαφές.

2. Παράλληλη θύρα:



Κοινές θύρες επικοινωνίας

Ιστορία: Η παράλληλη θύρα εμφανίστηκε στα τέλη της δεκαετίας του 1980, προσφέροντας σημαντικά υψηλότερες ταχύτητες από τη σειριακή.

Χαρακτηριστικά: Χρησιμοποιεί ταυτόχρονη μεταφορά πολλαπλών bits (8 ή 16) για ταχύτερη επικοινωνία.

Πλεονεκτήματα: Υψηλότερες ταχύτητες (έως 45 Mbps), ιδανική για εκτυπωτές.

Μειονεκτήματα: Περίπλοκη κατασκευή, βαρύτερα καλώδια, μικρότερο μήκος καλωδίου.

Σήμερα: Η παράλληλη θύρα έχει σχεδόν εξαφανιστεί, **αντικαταστάθηκε από USB** και άλλες ταχύτερες και πιο ευέλικτες διεπαφές.

3. USB (Universal Serial Bus):

Ιστορία: Το USB κυκλοφόρησε το 1996 και εξελίχθηκε σε μια από τις πιο δημοφιλείς και ευέλικτες θύρες σύνδεσης.

Χαρακτηριστικά: Υποστηρίζει plug-and-play, hot-swapping, πολλαπλές συσκευές σε μια θύρα, τροφοδοσία συσκευών.

Πλεονεκτήματα: Ταχύτητα (έως 40 Gbps), ευκολία χρήσης, ευελιξία, τροφοδοσία συσκευών.

Μειονεκτήματα: Περίπλοκη κατασκευή, περιορισμένο μήκος καλωδίου (ανάλογα με την έκδοση).

Σήμερα: Το USB κυριαρχεί στις θύρες σύνδεσης, με πολλαπλές εκδόσεις (USB 2.0, 3.0, 3.1, 3.2, 4.0) που προσφέρουν ολοένα αυξανόμενες ταχύτητες.

Εικονικοποίηση (Virtualization)

Η εικονικοποίηση είναι μια τεχνολογία που επιτρέπει τη δημιουργία πολλαπλών εικονικών μηχανών (VMs) πάνω από το ίδιο φυσικό υλικό. Αυτή η τεχνολογία είναι βασική στα σύγχρονα συστήματα νέφους, επιτρέποντας την αποδοτική χρήση των πόρων του υλικού μέσω του διαμοιρασμού τους σε πολλαπλούς χρήστες και εφαρμογές

Συμπεράσματα

Οι έννοιες του υλικού είναι κρίσιμες για την κατανόηση της λειτουργίας των λειτουργικών συστημάτων. Η κατανόηση του πώς λειτουργούν τα συστήματα αποθήκευσης, η μνήμη, οι επεξεργαστές, και η εικονικοποίηση βοηθά στο να κατανοήσουμε πώς τα λειτουργικά συστήματα διαχειρίζονται αυτούς τους πόρους και πώς μπορούν να βελτιστοποιηθούν για τις απαιτήσεις των νέων τεχνολογιών, όπως αυτές του νέφους.



Ιδεατές μονάδες (μηχανές)

Αναφορές αρχιτεκτονικής υπολογιστών

A. Βασικό διάγραμμα

<https://hothardware.com/reviews/intel-core-i7-8700k-and-core-i5-8400-coffee-lake-processor-review?page=2>

<https://www.spiceworks.com/tech/tech-general/articles/what-is-computer-architecture/>

B. Επεξεργαστής

<https://www.geeksforgeeks.org/architecture-of-8086/>

https://www.researchgate.net/figure/Generalized-block-diagram-of-a-multi-core-CPU_fig5_363228853

C. Μνήμη

<https://blog.cloudflare.com/ddr4-memory-organization-and-how-it-affects-memory-bandwidth>

<https://www.systemverilog.io/design/ddr4-basics/>

D. Γραφικά

https://en-m-wikipedia-org.translate.goog/wiki/Graphics_card? x tr sl=en& x tr tl=el& x tr hl=el& x tr pto=wapp

<https://core.vmware.com/resource/exploring-gpu-architecture#section2>

https://en.wikipedia.org/wiki/Display_resolution

E. Μονάδες αποθήκευσης

<https://www.bartleby.com/subject/engineering/computer-science/concepts/storage-devices>

<https://echoglobal.tech/a-brief-history-of-data-storage/>

<https://www.pcmag.com/news/the-evolution-of-pc-storage-media>

<https://www.backblaze.com/blog/history-removable-computer-storage/>

F. Θύρες επικοινωνιών

<https://recompute.co.zw/buying-guides/a-complete-guide-of-every-type-of-computer-port/>

<https://www.electronicshub.org/types-of-computer-ports/>

G. Busses & PCI express

https://en-m-wikipedia-org.translate.goog/wiki/PCI_Express? x tr sl=en& x tr tl=el& x tr hl=el& x tr pto=wapp

<https://pcisig.com/blog/evolution-pci-express-specification-its-sixth-generation-third-decade-and-still-going-strong>

Εισαγωγή στα λειτουργικά συστήματα (Operating Systems)

Στόχοι:

- Να εξηγήσουμε τις βασικές αρχές και τον ρόλο των λειτουργικών συστημάτων.
- Να αναλύσουμε το UEFI, τον boot loader, και τη διαδικασία εκκίνησης του συστήματος.

Λειτουργικά Συστήματα

- Linux Distro (eg UBUNTU, Fedora, CentOS etc)
- Mac OS
- Windows
- Unix
- FreeBSD
- Android
- iOS
- Windows mobile
- HarmonyOS (Huawei)



UEFI

Unified Extensible Firmware Interface (UEFI) is a modern computer software that starts up devices, ensuring compatibility, security, and advanced features.

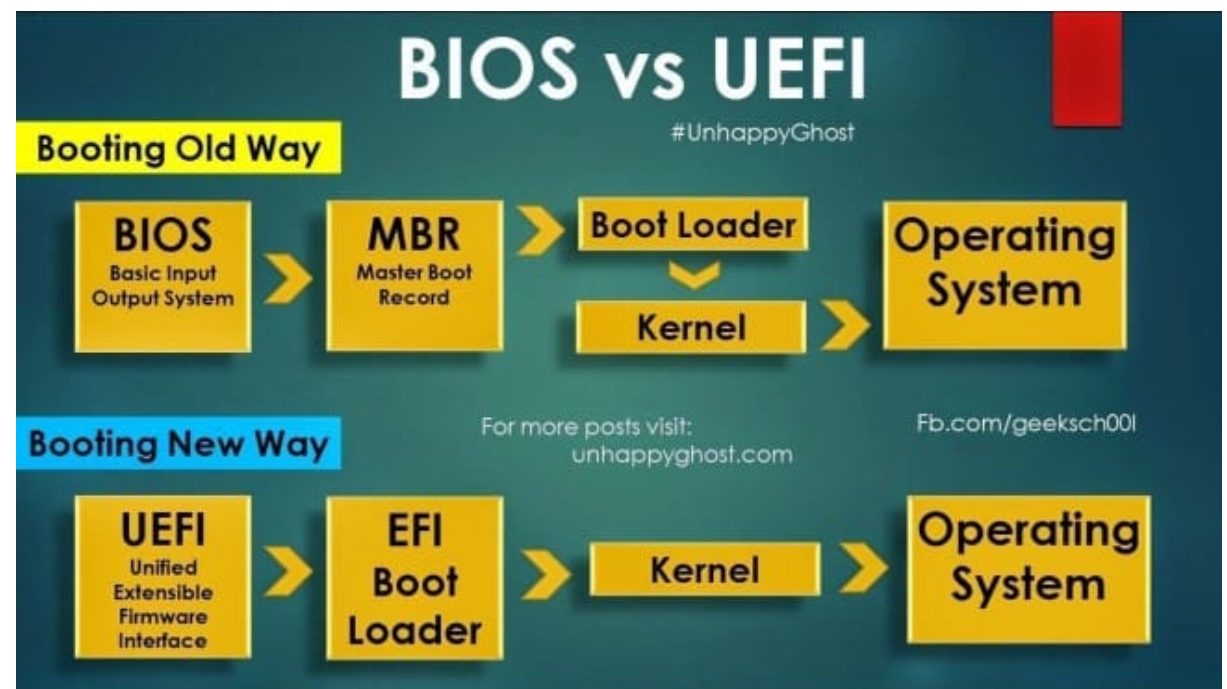
Το Unified Extensible Firmware Interface (UEFI) είναι μια σύγχρονη διεπαφή υλικολογισμικού που αντικαθιστά το παραδοσιακό BIOS (Basic Input/Output System) σε υπολογιστές. Χρησιμεύει ως βασικός σύνδεσμος μεταξύ του λειτουργικού συστήματος και των στοιχείων υλικού ενός υπολογιστή, διευκολύνοντας τη διαδικασία προετοιμασίας και παρέχοντας έναν τυποποιημένο τρόπο αλληλεπίδρασης του λειτουργικού συστήματος με το UEFI του συστήματος.

Το UEFI προσφέρει πολλά πλεονεκτήματα σε σχέση με το παλαιότερο BIOS, συμπεριλαμβανομένης της υποστήριξης για μεγαλύτερες συσκευές αποθήκευσης, ταχύτερους χρόνους εκκίνησης, βελτιωμένες δυνατότητες ασφαλείας και μια πιο φιλική προς το χρήστη διεπαφή. Ένα από τα κύρια κίνητρα πίσω από την ανάπτυξη του UEFI ήταν να ξεπεραστούν οι περιορισμοί του BIOS και να παρέχουμε μια πιο ευέλικτη και επεκτάσιμη πλατφόρμα για την ανάπτυξη hardware & software.

Για να κατανοήσετε καλύτερα το UEFI, εξετάστε ένα παράδειγμα που περιλαμβάνει τη διαδικασία εκκίνησης ενός υπολογιστή. Σε ένα σύστημα με UEFI, η διαδικασία ξεκινά όταν ο υπολογιστής είναι ενεργοποιημένος. Το UEFI αναλαμβάνει τον έλεγχο του συστήματος και ξεκινά το Power-On Self-Test (POST),

μια διαγνωστική διαδικασία που ελέγχει την ακεραιότητα των στοιχείων hardware. Μόλις ολοκληρωθεί με επιτυχία το POST, το UEFI αναζητά μια συσκευή με δυνατότητα εκκίνησης.

Σε ένα σύστημα UEFI, το σύστημα αλληλεπιδρά με τον διαχειριστή εκκίνησης UEFI, δηλαδή ένα στοιχείο λογισμικού που είναι υπεύθυνο για τη φόρτωση του λειτουργικού συστήματος. Ο διαχειριστής εκκίνησης συμβουλεύεται τις μεταβλητές εκκίνησης UEFI, οι οποίες περιέχουν πληροφορίες σχετικά με τις διαθέσιμες επιλογές εκκίνησης και τις προτεραιότητές τους. Αυτές οι μεταβλητές αποθηκεύονται σε μη πτητική μνήμη και μπορούν να τροποποιηθούν από τον χρήστη ή τον διαχειριστή του συστήματος για να προσαρμόσουν τη διαμόρφωση εκκίνησης.



UEFI εκκίνηση συστήματος

Ας υποθέσουμε ότι ο υπολογιστής έχει ένα λειτουργικό σύστημα εγκατεστημένο σε έναν SSD (Solid State Drive). Το UEFI, με τη βοήθεια του διαχειριστή εκκίνησης, εντοπίζει τον loader εκκίνησης UEFI που είναι αποθηκευμένος στο διαμέρισμα συστήματος EFI (ESP) του SSD. Ο loader εκκίνησης UEFI είναι ένα μικρό πρόγραμμα που κατανοεί το σύστημα αρχείων και ξέρει πώς να φορτώνει τον πυρήνα του λειτουργικού συστήματος στη μνήμη.

Μόλις ο loader εκκίνησης UEFI φορτώσει τον πυρήνα του λειτουργικού συστήματος, ο έλεγχος μεταφέρεται στο λειτουργικό σύστημα και παρουσιάζεται η διεπαφή χρήστη. Σε όλη αυτή τη διαδικασία, το UEFI διευκολύνει την επικοινωνία μεταξύ του υλικού και του λογισμικού, εξασφαλίζοντας μια ομαλή και αποτελεσματική εμπειρία εκκίνησης.

Έτσι, το UEFI είναι μια κρίσιμη διεπαφή hard-software που παρέχει ένα τυποποιημένο και επεκτάσιμο πλαίσιο για την ανάπτυξη hardware & software, επιτρέποντας βελτιωμένη απόδοση, υποστήριξη για μεγαλύτερες συσκευές αποθήκευσης και βελτιωμένα χαρακτηριστικά ασφαλείας. Η μετάβαση από το BIOS στο UEFI αντιπροσωπεύει μια σημαντική πρόοδο στην εξέλιξη του hardware & software.

Το UEFI είναι ζωτικής σημασίας για τους σύγχρονους υπολογιστές, προσφέροντας μια τυποποιημένη και ευέλικτη πλατφόρμα hardware & software που βελτιώνει σημαντικά την απόδοση, την ασφάλεια και τη διαχειρισιμότητα του συστήματος. Μια συναρπαστική περίπτωση χρήσης που καταδεικνύει τη σημασία του UEFI είναι στο πλαίσιο της ασφαλούς εκκίνησης σε εταιρικά περιβάλλοντα.

Εκεί η λειτουργία ασφαλούς εκκίνησης του UEFI παίζει βασικό ρόλο, καθώς αποτρέπει την εκτέλεση μη εξουσιοδοτημένου ή μη υπογεγραμμένου hardware & software, προγραμμάτων οδήγησης και λειτουργικών συστημάτων κατά τη διαδικασία εκκίνησης. Αυτό διασφαλίζει ότι μπορούν να εκτελούνται μόνο επαληθευμένα και αξιόπιστα στοιχεία, μειώνοντας τον κίνδυνο κακόβουλου λογισμικού (όπως bootkits και rootkits) και μη εξουσιοδοτημένου κώδικα που διακυβεύει το σύστημα.

```
[kousekip@ako-kaede-mirai]-(11:33am:--08/12) [~] r efivar --list
8be4df61-93ca-11d2-aa0d-00e098032b8c-dbxDefault
8be4df61-93ca-11d2-aa0d-00e098032b8c-dbDefault
8be4df61-93ca-11d2-aa0d-00e098032b8c-KEKDefault
8be4df61-93ca-11d2-aa0d-00e098032b8c-PKDefault
07a66697-d400-4903-b3da-67a61d2b7058-Tcg2ConfigInfo
aa1305b9-01f3-4afb-920e-c9b979a852fd-SecureBootData
1f2d63e1-febd-4dc7-9cc5-ba2b1cef9c5b-FeData
3a997502-647a-4c82-998e-52ef9486a247-AmdSetup
5bce4c83-6a97-444b-63b4-672c014742ff-IrsiInfo
146b234d-4052-4e07-b326-11220f8e1fe8-lBoot0002
146b234d-4052-4e07-b326-11220f8e1fe8-lBoot0001
8be4df61-93ca-11d2-aa0d-00e098032b8c-BootCurrent
74b00bd9-805a-4d61-b51f-43268123d113-RTL-pwrovr
8be4df61-93ca-11d2-aa0d-00e098032b8c-ErrOutDev
8be4df61-93ca-11d2-aa0d-00e098032b8c-ConInDev
59d1c24f-50f1-401a-b101-f33e0daed443-ConOutCandidateDev
8be4df61-93ca-11d2-aa0d-00e098032b8c-ConOutDev
79941ecd-ed36-49d0-8124-e4c31ac75cd4-AmdAcpiVar
59d1c24f-50f1-401a-b101-f33e0daed443-PlugInVgaHandles
59d1c24f-50f1-401a-b101-f33e0daed443-ConInCandidateDev
8be4df61-93ca-11d2-aa0d-00e098032b8c-OsIndicationsSupported
6acce65d-da35-4b39-b64b-5ed927a7dc7e-LvarSmiReadyFlag
8be4df61-93ca-11d2-aa0d-00e098032b8c-PlatformLangCodes
8be4df61-93ca-11d2-aa0d-00e098032b8c-LangCodes
8be4df61-93ca-11d2-aa0d-00e098032b8c-VendorKeys
59d1c24f-50f1-401a-b101-f33e0daed443-certdbv
8be4df61-93ca-11d2-aa0d-00e098032b8c-SignatureSupport
8be4df61-93ca-11d2-aa0d-00e098032b8c-BootOrder
8be4df61-93ca-11d2-aa0d-00e098032b8c-SecureBoot
a04a27f4-df00-4d42-b552-39511302113d-Setup
59d1c24f-50f1-401a-b101-f33e0daed443-SecureBootEnforce
8be4df61-93ca-11d2-aa0d-00e098032b8c-OsIndications
59d1c24f-50f1-401a-b101-f33e0daed443-PhysicalBootOrder
d719b2cb-3d3a-4596-a3bc-dad00e67656f-db
8be4df61-93ca-11d2-aa0d-00e098032b8c-SetupMode
8be4df61-93ca-11d2-aa0d-00e098032b8c-PK
d719b2cb-3d3a-4596-a3bc-dad00e67656f-dbx
8be4df61-93ca-11d2-aa0d-00e098032b8c-KEK
59d1c24f-50f1-401a-b101-f33e0daed443-CustomSecurity
8be4df61-93ca-11d2-aa0d-00e098032b8c-Boot0003
8be4df61-93ca-11d2-aa0d-00e098032b8c-Boot0002
8be4df61-93ca-11d2-aa0d-00e098032b8c-Boot0001
```

Παράδειγμα μεταβλητών αρχικοποίησης συστήματος (Linux)

BOOT LOADER

Πριν από τον πυρήνα (kernel) ενός λειτουργικού συστήματος, εκτελείται το bootloader.

Σε συστήματα με UEFI (Unified Extensible Firmware Interface), η σειρά εκκίνησης πριν από τον πυρήνα (kernel) είναι η εξής:

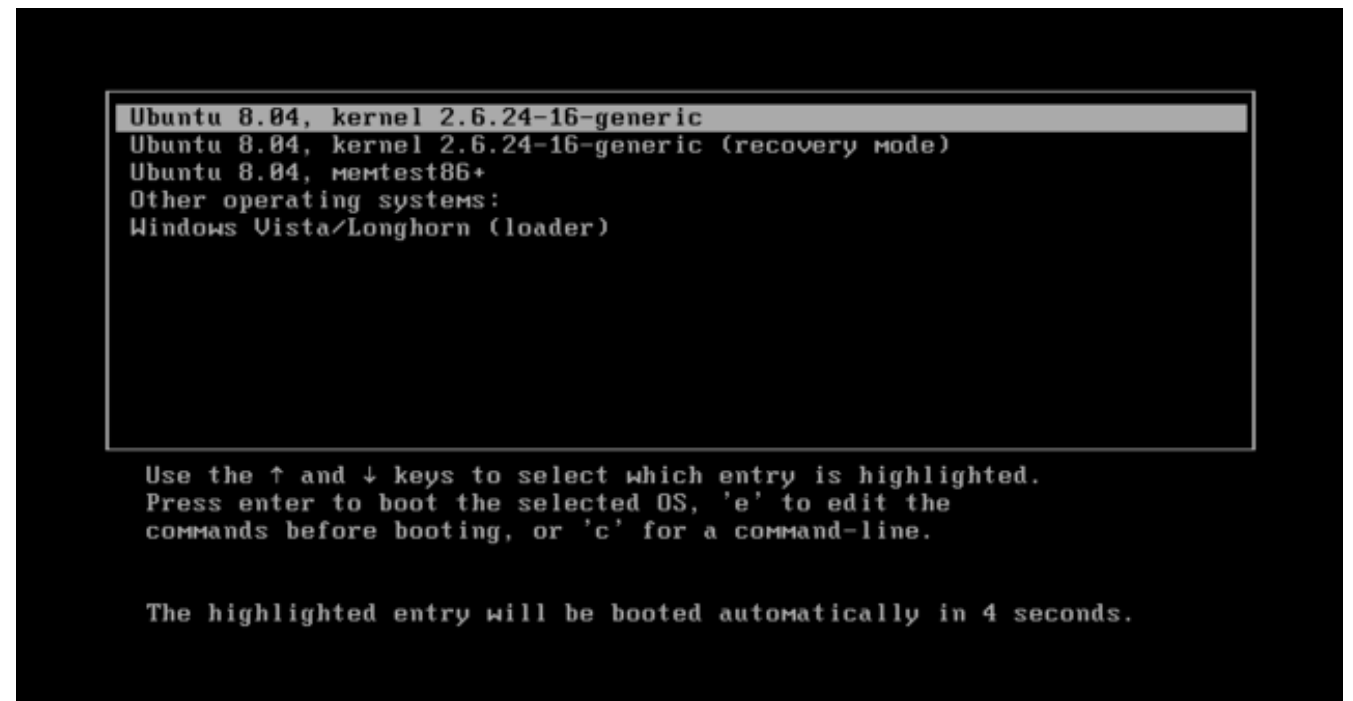
POST (Power-On Self Test): Το UEFI εκτελεί τον POST, ο οποίος ελέγχει την υλική ακεραιότητα του συστήματος (CPU, μνήμη, κάρτα γραφικών, αποθηκευτικός χώρος, κτλ.).

Αναζήτηση bootloader: Το UEFI ψάχνει για ένα εκτελέσιμο πρόγραμμα bootloader σε συμβατούς δίσκους, όπως SSD ή HDD.

Φόρτωση bootloader: Το UEFI εντοπίζει και φορτώνει τον bootloader στη μνήμη RAM.

Εκτέλεση bootloader: Ο bootloader εμφανίζει ένα μενού με διαθέσιμες επιλογές εκκίνησης (λειτουργικά συστήματα, διαγνωστικά εργαλεία, ρυθμίσεις UEFI).

Ο χρήστης επιλέγει το λειτουργικό σύστημα που θέλει να εκκινήσει.



Multi OS Boot Loader

Φόρτωση

πυρήνα: Ο bootloader φορτώνει τον πυρήνα του επιλεγμένου λειτουργικού συστήματος στη μνήμη RAM.

Έλεγχος πυρήνα: Ο bootloader ελέγχει την ακεραιότητα του πυρήνα και βεβαιώνεται ότι είναι συμβατός με το σύστημα.

Εκκίνηση πυρήνα: Ο bootloader παραδίδει τον έλεγχο στον πυρήνα, ο οποίος αναλαμβάνει την εκτέλεση των

λειτουργιών του λειτουργικού συστήματος.

Ο **GRUB** μπορεί να είναι μια καλή επιλογή για bootloader σε συστήματα UEFI.

Ο GRUB υποστηρίζει μια μεγάλη ποικιλία λειτουργικών συστημάτων, συμπεριλαμβανομένων Linux, Windows και macOS.

Επίσης υποστηρίζει LVM (Logical Volume Management), EFI bootstrap και άλλες προηγμένες λειτουργίες.

Μετά το UEFI

Αφού το UEFI ολοκληρώσει την εκκίνηση του υλικού και φορτώσει το λειτουργικό σύστημα (ΛΣ) στη μνήμη, το ΛΣ αναλαμβάνει τον έλεγχο του υπολογιστή σας, λειτουργώντας ως ο διευθυντής ορχήστρας για όλες τις διεργασίες και τα προγράμματα.

Βασικές Αρμοδιότητες του ΛΣ:

Διαχείριση Πόρων: Κατανέμει πόρους όπως μνήμη, επεξεργαστή και αποθηκευτικό χώρο στα προγράμματα που εκτελούνται.

Εκτέλεση Προγραμμάτων: Φορτώνει προγράμματα στη μνήμη, εκτελεί τις εντολές τους και διαχειρίζεται την εκτέλεσή τους.

Παροχή Διεπαφής Χρήστη: Προσφέρει ένα περιβάλλον (όπως γραφική επιφάνεια εργασίας ή γραμμή εντολών) για αλληλεπίδραση με τον υπολογιστή.



Διάφορα είδη λειτουργικών συστημάτων

Διαχείριση Αρχείων: Οργανώνει, αποθηκεύει και ανακτά αρχεία και φακέλους στον σκληρό δίσκο ή άλλα μέσα αποθήκευσης.

Επικοινωνία με Συσκευές: Ελέγχει και συντονίζει την επικοινωνία με περιφερειακές συσκευές όπως εκτυπωτές, ποντίκια, πληκτρολόγια και δίκτυα.

Παροχή Ασφάλειας: Προστατεύει τον υπολογιστή από ιούς, κακόβουλο λογισμικό και μη εξουσιοδοτημένη πρόσβαση.

Υποστήριξη Δικτύου: Επιτρέπει την σύνδεση με το διαδίκτυο και την χρήση δικτυακών υπηρεσιών.

Λειτουργικά συστήματα

Ας δούμε τα βασικά χαρακτηριστικά και τις δυνατότητες των πέντε δημοφιλέστερων λειτουργικών συστημάτων:

1. Ubuntu:

Ελεύθερο και Open-Source: Βασισμένο στο Linux, προσφέρεται δωρεάν και με πλήρη πρόσβαση στον κώδικα.

Ευελιξία: Διαθέτει πολλές διανομές με διαφορετικά περιβάλλοντα γραφικής επιφάνειας και εφαρμογές.

Ιδανικό για: Χρήστες που θέλουν ευελιξία, προγραμματιστές, λάτρεις του open-source.

Μειονεκτήματα: Μπορεί να έχει καμπύλη εκμάθησης για νέους χρήστες, η υποστήριξη ορισμένων συσκευών ενδέχεται να είναι περιορισμένη.

2. macOS:

Αναπτύχθηκε από την Apple: Σχεδιασμένο αποκλειστικά για συσκευές Apple (MacBooks, iMacs).

Εύχρηστο: Φιλικό προς τον χρήστη περιβάλλον, ομαλή ενσωμάτωση με άλλα προϊόντα Apple.

Ιδανικό για: Χρήστες Apple που αναζητούν απλότητα και ομαλή εμπειρία.

Μειονεκτήματα: Κλειστό σύστημα, υψηλότερο κόστος συσκευών, περιορισμένη συμβατότητα με άλλα συστήματα.

3. Windows:

Το πιο δημοφιλές: Χρησιμοποιείται από την πλειοψηφία των υπολογιστών παγκοσμίως.

Ευρεία συμβατότητα: Υποστηρίζει μεγάλο εύρος λογισμικού και περιφερειακών συσκευών.

Ιδανικό για: Χρήστες που θέλουν ευκολία χρήσης, μεγάλη ποικιλία λογισμικού, συμβατότητα.

Μειονεκτήματα: Κλειστό σύστημα, συχνές ενημερώσεις, πιθανά προβλήματα ασφάλειας.

4. **Android:** Λειτουργικό Συστήματος Κινητών: Δημιουργήθηκε από την Google, κυριαρχεί στις smartphones και tablets.

Ανοιχτό: Μεγάλη ποικιλία εφαρμογών και προσαρμογών.

Ιδανικό για: Χρήστες smartphones και tablets που αναζητούν ευελιξία και μεγάλη ποικιλία εφαρμογών.

Μειονεκτήματα: Κίνδυνος κακόβουλου λογισμικού, ασυμβατότητες της αγοράς (πολλές εκδόσεις), θέματα ασφάλειας.

5. iOS:

Αναπτύχθηκε από την Apple: Σχεδιασμένο αποκλειστικά για συσκευές Apple (iPhone, iPad).

Κλειστό σύστημα: Ασφαλές, ομαλή ενσωμάτωση με άλλα προϊόντα Apple.

Ιδανικό για: Χρήστες Apple που αναζητούν ασφάλεια, απλότητα και ομαλή εμπειρία.

Μειονεκτήματα: Υψηλότερο κόστος συσκευών, περιορισμένη ευελιξία, συμβατότητα μόνο με προϊόντα Apple.

Χαρακτηριστικό	Ubuntu	macOS	Windows	Android	iOS
Κωδικός:	Open-Source	Κλειστός	Κλειστός	Open-Source	Κλειστός
Πλατφόρμα:	Linux	macOS	Windows	Android	iOS
Υποστήριξη:	PC, Servers	MacBooks, iMacs	PC, Servers	Smartphones, Tablets	iPhones, iPads
Περιβάλλον:	GNOME, KDE, Xfce, LXDE	macOS	Windows 11, Windows 10	Android UI	iOS UI
Εκδόσεις:	Ubuntu, Kubuntu, Xubuntu, Lubuntu, Mint	macOS Monterey, Ventura	Windows 11, Windows 10, 8.1, 7	Android 13, 12, 11, 10	iOS 16, 15, 14, 13
Διαχείριση:	Πακέτα λογισμικού (apt, dpkg)	App Store	Microsoft Store	Google Play Store	App Store
Προγραμματισμός:	Python, Java, C++, C#	Swift, Objective-C	C#, C++, Java	Java, Kotlin	Swift
Ασφάλεια:	Θεωρείται ασφαλές, με ενημερώσεις κώδικα	Υψηλή ασφάλεια, sandboxed apps	Υψηλή ασφάλεια, Windows Defender	Θεωρείται ασφαλές, Google Play Protect	Υψηλή ασφάλεια, sandboxed apps
Κόστος:	Δωρεάν	Υψηλό	Χαμηλό έως υψηλό	Χαμηλό έως υψηλό	Υψηλό
Ευελιξία:	Υψηλή, πολλές διανομές και επιλογές	Μέτρια	Μέτρια	Υψηλή	Μέτρια

Χαρακτηριστικά λειτουργικών συστημάτων

Διαχείριση Διεργασιών και Νημάτων (Processes and Threads)

Στόχοι:

- Κατανόηση της έννοιας και της δομής των διεργασιών.
- Εξήγηση του τρόπου διαχείρισης των διεργασιών από το λειτουργικό σύστημα.
- Κατανόηση της διαχείρισης νημάτων και του πολυνηματικού προγραμματισμού.
- Ανάλυση ζητημάτων συγχρονισμού και προβλημάτων κριτικών περιοχών.
- Κατανόηση των προβλημάτων deadlock και των μεθόδων για την αποφυγή τους.

Η διαχείριση διεργασιών είναι μια από τις κρίσιμες έννοιες που χρησιμοποιούνται στα λειτουργικά συστήματα και περιλαμβάνει δραστηριότητες που σχετίζονται με τη διαχείριση και τον έλεγχο των διαδικασιών.

Η μνήμη διεργασιών χωρίζεται σε τέσσερις ενότητες:

Η μνήμη προγράμματος αποθηκεύει τον μεταγλωττισμένο κώδικα προγράμματος.

Η ενότητα δεδομένων αποθηκεύει καθολικές και στατικές μεταβλητές.

Η ενότητα heap διαχειρίζεται τη δυναμική εκχώρηση μνήμης μέσα στο πρόγραμμα. Είναι το τμήμα της μνήμης όπου βρίσκεται η δυναμικά εκχωρημένη μνήμη, π.χ. εκχώρηση μνήμης μέσω νέου ή malloc και εκχώρηση μνήμης μέσω διαγραφής ή δωρεάν κ.λπ.

Η ενότητα στοίβας αποθηκεύει τοπικές μεταβλητές που ορίζονται μέσα στο πρόγραμμα.

Μπλοκ ελέγχου διεργασίας (**PCB**)

Ένα πρόγραμμα που εκτελείται ως διεργασία καθορίζεται μοναδικά από διάφορες παραμέτρους. Αυτές οι παράμετροι αποθηκεύονται σε ένα μπλοκ ελέγχου διεργασίας (PCB), δηλαδή μια δομή δεδομένων που περιέχει τις ακόλουθες πληροφορίες:

Αναγνωριστικό διεργασίας: Μοναδικό αναγνωριστικό που σχετίζεται με κάθε διεργασία.

Κατάσταση διεργασίας: Μια διεργασία μπορεί να βρίσκεται σε κατάσταση έναρξης, έτοιμη, εκτελούμενη, αναμονής ή τερματισμού.

Process Control Block (PCB)

Process Identifier
Process State
CPU-Scheduling information
Memory-Management information
CPU registers and Program Counter
Accounting Information
I/O Status Information
.
.
.

Μπλόκ ελέγχου διεργασίας

Πληροφορίες προγραμματισμού CPU: Σχετίζονται με το επίπεδο προτεραιότητας σε σχέση με τις άλλες διεργασίες και δείκτες στις ουρές προγραμματισμού.

Καταχωρητές CPU και μετρητής προγράμματος: Πρέπει να αποθηκευτούν και να αποκατασταθούν κατά την εναλλαγή διεργασιών εντός και εκτός της CPU.

Διεργασίες σε λειτουργικά συστήματα Α Μέρος

Η δημιουργία διεργασιών είναι η διαδικασία μέσω της οποίας το λειτουργικό σύστημα προσθέτει νέες διεργασίες για εκτέλεση. Οι νέες διεργασίες δημιουργούνται μέσω του λειτουργικού συστήματος, συχνά από άλλες διεργασίες μέσω συστημάτων κλήσεων, όπως η κλήση `fork()` σε συστήματα Unix..

Οι διεργασίες στο λειτουργικό σύστημα μπορούν να βρίσκονται σε οποιαδήποτε από τις πέντε καταστάσεις: έναρξη, έτοιμη, εκτέλεση, αναμονή και τερματισμός.

Ας κατανοήσουμε αυτές τις καταστάσεις και τη μετάβαση μιας διαδικασίας από μια κατάσταση σε μια άλλη κατάσταση:

Null to Start: Όταν δημιουργείται μια νέα διαδικασία, μια διαδικασία μεταβαίνει από την κατάσταση NULL στην κατάσταση έναρξης.

Start to Ready: Το λειτουργικό σύστημα μετακινεί μια διεργασία από την κατάσταση έναρξης στην κατάσταση ετοιμότητας όταν είναι έτοιμο να εκτελέσει μια πρόσθετη διεργασία. Σε αυτό το στάδιο, η διαδικασία έχει όλους τους διαθέσιμους πόρους που χρειάζεται για να τρέξει και περιμένει να λάβει χρόνο CPU για την εκτέλεσή της.

Έτοιμο για εκτέλεση: Το λειτουργικό σύστημα επιλέγει μία από τις διεργασίες από την κατάσταση ετοιμότητας χρησιμοποιώντας τον προγραμματιστή ή τον αποστολέα διεργασιών. Μετά από αυτό, η CPU ξεκινά να εκτελεί την επιλεγμένη διαδικασία σε κατάσταση λειτουργίας.

Εκτέλεση έως τερματισμό: Η εκτελούμενη διαδικασία τερματίζεται από το λειτουργικό σύστημα όταν ολοκληρωθεί.

Εκτέλεση σε ετοιμότητα: Ο προγραμματιστής ορίζει ένα συγκεκριμένο χρονικό όριο για την εκτέλεση οποιασδήποτε ενεργής διαδικασίας. Εάν η τρέχουσα διαδικασία απαιτεί περισσότερο χρόνο από αυτόν που καθορίζεται από τον χρονοπρογραμματιστή, επιστρέφει στην κατάσταση ετοιμότητας. Ο πιο κρίσιμος λόγος για αυτή τη μετάβαση είναι ότι έχει φτάσει στο μέγιστο επιτρεπόμενο χρονικό διάστημα για αδιάλειπτη εκτέλεση. Σχεδόν όλα τα λειτουργικά συστήματα πολλαπλού προγραμματισμού επιβάλλουν αυτόν τον χρονικό περιορισμό.

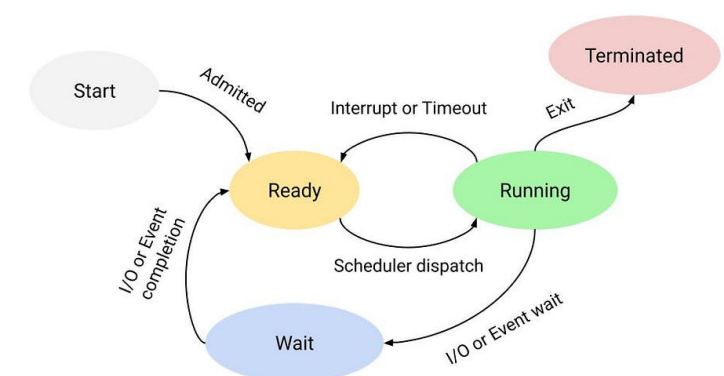
Εκτέλεση για αναμονή: Εάν μια διεργασία απαιτεί κάτι για το οποίο πρέπει να περιμένει, τοποθετείται σε κατάσταση αναμονής. Τώρα, η διαδικασία δεν μπορεί να εκτελεστεί επειδή περιμένει να γίνει

διαθέσιμος κάποιος πόρος ή να συμβεί κάποιο συμβάν. Για παράδειγμα, η διαδικασία μπορεί να περιμένει για είσοδο πληκτρολογίου, αίτημα πρόσβασης στο δίσκο, μηνύματα μεταξύ διεργασιών, τερματισμό ενός χρονοδιακόπτη ή ολοκλήρωση μιας θυγατρικής διαδικασίας.

Wait to Ready: Μια διεργασία σε κατάσταση αναμονής μεταφέρεται στην κατάσταση ετοιμότητας όταν ολοκληρωθεί το συμβάν για το οποίο αναμένει.

Το λειτουργικό σύστημα έχει τρεις τύπους προγραμματιστών διεργασιών:

Μακροπρόθεσμος ή προγραμματιστής εργασιών: Η δουλειά αυτού του χρονοπρογραμματιστή είναι να φέρει νέες διεργασίες στην κατάσταση Έτοιμη. Καθορίζει ποια διεργασία έχει εκχωρηθεί στην CPU για επεξεργασία, επιλέγει διεργασίες από την ουρά και τις φορτώνει στη μνήμη για εκτέλεση.



Εκτέλεση και καταστάσεις διεργασίας

Διεργασίες σε λειτουργικά συστήματα Β Μέρος

Μακροπρόθεσμος..... Ο πρωταρχικός στόχος του προγραμματιστή εργασιών είναι να παρέχει έναν ισορροπημένο συνδυασμό εργασιών (όπως δέσμευση εισόδου/εξόδου και δέσμευση επεξεργαστή) και να ελέγχει τον βαθμό πολυπρογραμματισμού.

Βραχυπρόθεσμος ή προγραμματιστής CPU: Είναι υπεύθυνος για την επιλογή μιας διαδικασίας από την κατάσταση ετοιμότητας και τον προγραμματισμό της στην κατάσταση εκτέλεσης. Είναι επίσης γνωστοί ως Dispatchers .

Ο προγραμματιστής CPU είναι υπεύθυνος για τη διασφάλιση της απουσίας ασυμμετρίας λόγω διεργασιών υψηλού χρόνου έκρηξης.

Εκτελείται συχνά και αλλάζει γρήγορα μια διεργασία από την κατάσταση εκτέλεσης στην κατάσταση ετοιμότητας.

Μεσοπρόθεσμος Προγραμματιστής:

Είναι υπεύθυνος για την εναλλαγή διεργασιών όταν μια συγκεκριμένη διεργασία εκτελεί μια λειτουργία I/O. Εάν μια διεργασία που εκτελείται κάνει ένα αίτημα εισόδου/εξόδου, μπορεί να ανασταλεί. Η διαδικασία που έχει ανασταλεί μεταφέρεται σε δευτερεύουσα αποθήκευση για να αφαιρεθεί από τη

μνήμη και να δημιουργηθεί χώρος για άλλες διεργασίες. Αυτό είναι γνωστό ως μεταγωγή.

Προγραμματισμός CPU στο λειτουργικό σύστημα

Τώρα ο στόχος μας θα ήταν να κατανοήσουμε την έννοια του Προγραμματισμού CPU και γιατί τον χρειαζόμαστε.

Τόσο ο χρόνος I/O όσο και ο χρόνος CPU χρησιμοποιείται σε μια τυπική διαδικασία. Ο χρόνος αναμονής για I/O σε ένα παλιό λειτουργικό σύστημα όπως το MS-DOS είναι χαμένος και η CPU είναι ελεύθερη κατά τη διάρκεια αυτής της περιόδου. Στον πολυπρογραμματισμό σύγχρονων λειτουργικών συστημάτων, μια διεργασία μπορεί να χρησιμοποιεί CPU ενώ μια άλλη περιμένει για I/O.

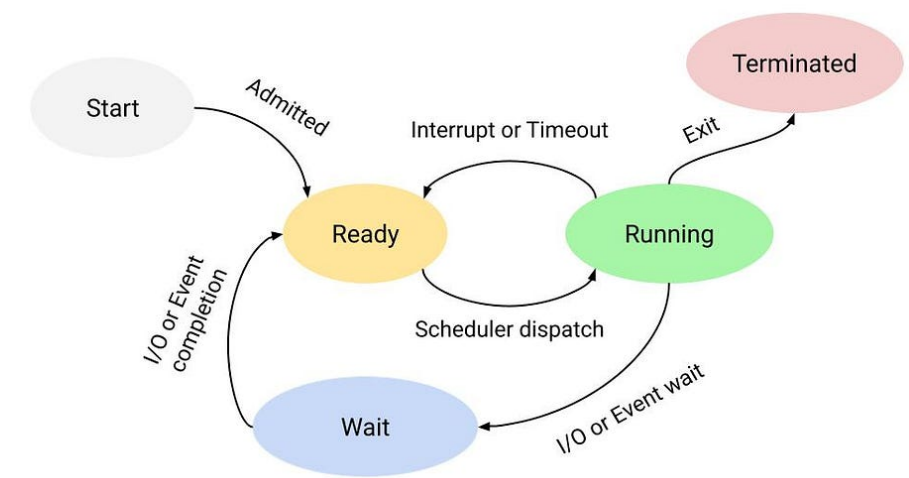
Ο προγραμματισμός της CPU καθορίζει ποια διεργασία θα χρησιμοποιεί αποκλειστικά CPU ενώ μια άλλη είναι σε παύση. Ο στόχος είναι να διασφαλιστεί ότι κάθε φορά που η CPU είναι σε αδράνεια, το OS επιλέγει τουλάχιστον ένα από τα προγράμματα στην έτοιμη ουρά για εκτέλεση.

Τύποι προγραμματισμού CPU

Υπάρχουν δύο κύριοι τύποι προγραμματισμού CPU:

Προληπτικός προγραμματισμός: Σε αυτόν τον προγραμματισμό, οι διεργασίες εκχωρούνται με τις προτεραιότητές τους και οι τρέχουσες διεργασίες μπορούν να διακοπουν από το λειτουργικό σύστημα, έτσι ώστε να μπορεί να εκτελεστεί μια άλλη διαδικασία με υψηλότερη προτεραιότητα.

Μη προληπτικός προγραμματισμός: Σε αυτόν τον προγραμματισμό, μια διαδικασία που εκτελείται δεν μπορεί να διακοπεί μέχρι να ολοκληρώσει την εκτέλεσή της ή να παραιτηθεί οικειοθελώς από τον έλεγχο της CPU. Έτσι, σε αυτήν την προσέγγιση, μόλις αρχίσει να εκτελείται μια διεργασία, συνεχίζει να εκτελείται μέχρι να ολοκληρωθεί ή να μπλοκάρει (π.χ., αναμονή για I/O).



Εκτέλεση και καταστάσεις διεργασίας

Διαχείριση Νημάτων (Threads) & ζητήματα συγχρονισμού

Τα **νήματα (threads)** είναι ελαφριές διεργασίες που μπορούν να εκτελούνται παράλληλα εντός μιας διεργασίας. Ενώ μια διεργασία έχει τη δική της ανεξάρτητη μνήμη, τα νήματα μοιράζονται πόρους της διεργασίας, όπως η μνήμη και οι ανοικτές συνδέσεις (file descriptors). Τα νήματα επιτρέπουν μεγαλύτερη αποδοτικότητα, ειδικά σε συστήματα πολλαπλών πυρήνων (multicore).

Κλασικό Μοντέλο Νημάτων:

- Κάθε διεργασία μπορεί να έχει πολλά νήματα εκτέλεσης.
- Τα νήματα εντός της ίδιας διεργασίας μοιράζονται τους ίδιους πόρους.

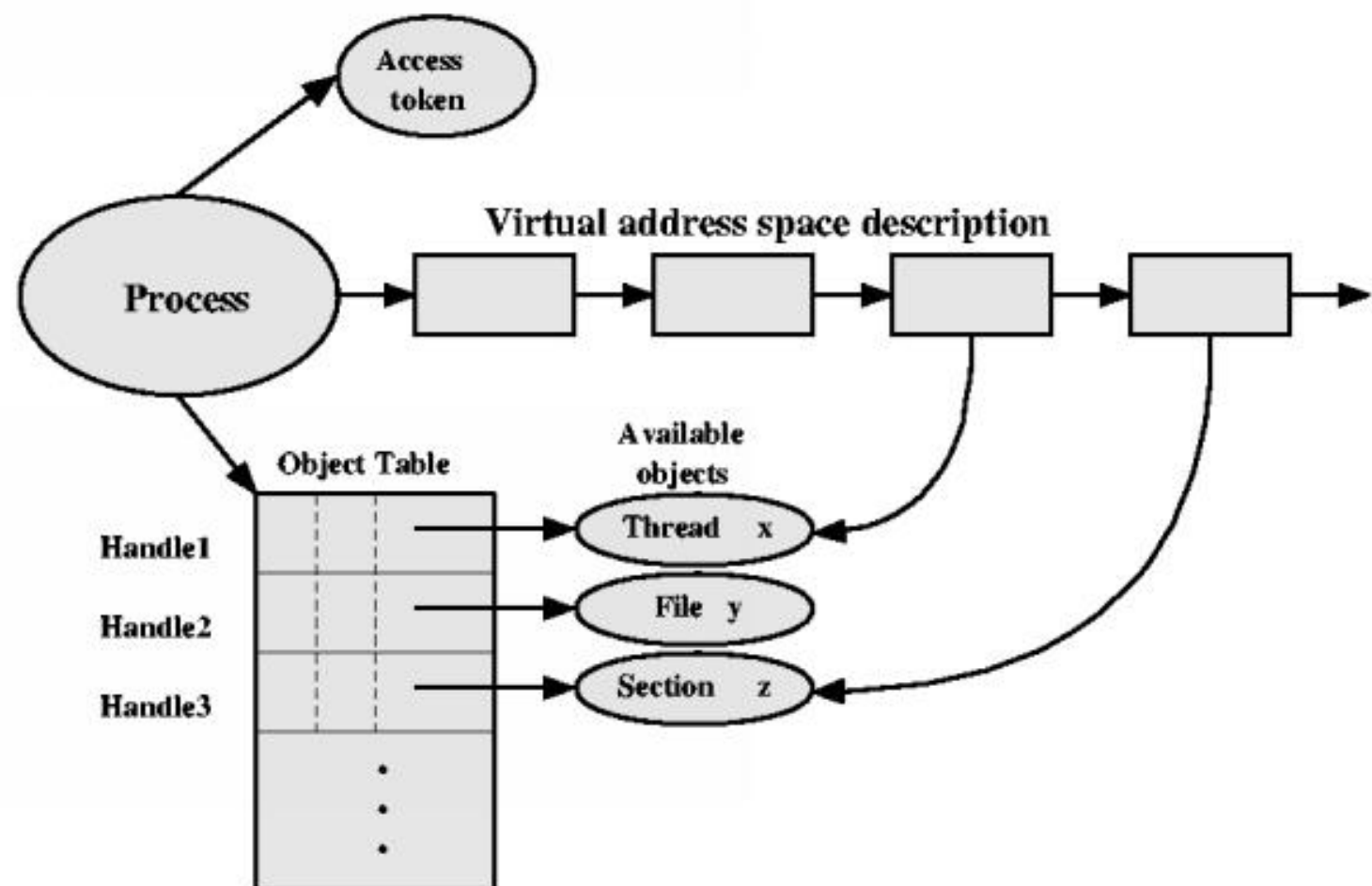
Υλοποίηση Νημάτων:

Υπάρχουν δύο κύριοι τρόποι υλοποίησης των νημάτων:

- **Νήματα Χρήστη (User-Level Threads):** Εκτελούνται πάνω από το λειτουργικό σύστημα.
- **Νήματα Kernel (Kernel-Level Threads):** Εκτελούνται απευθείας από τον πυρήνα του λειτουργικού συστήματος.

Πλεονεκτήματα Χρήσης Νημάτων:

- Καλύτερη απόδοση μέσω πολυνηματικής εκτέλεσης (multithreading).
- Καλύτερη διαχείριση I/O και βελτιωμένη απόδοση για εφαρμογές που περιμένουν δεδομένα.



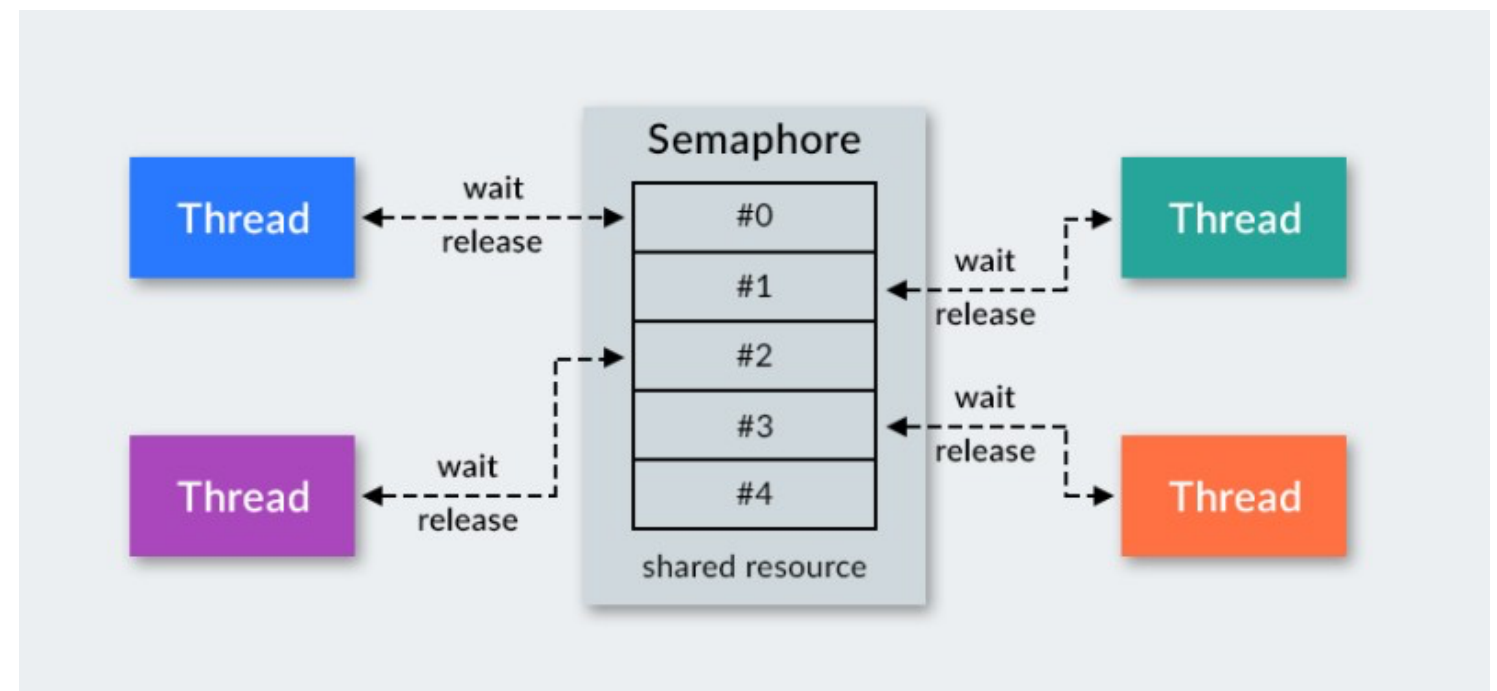
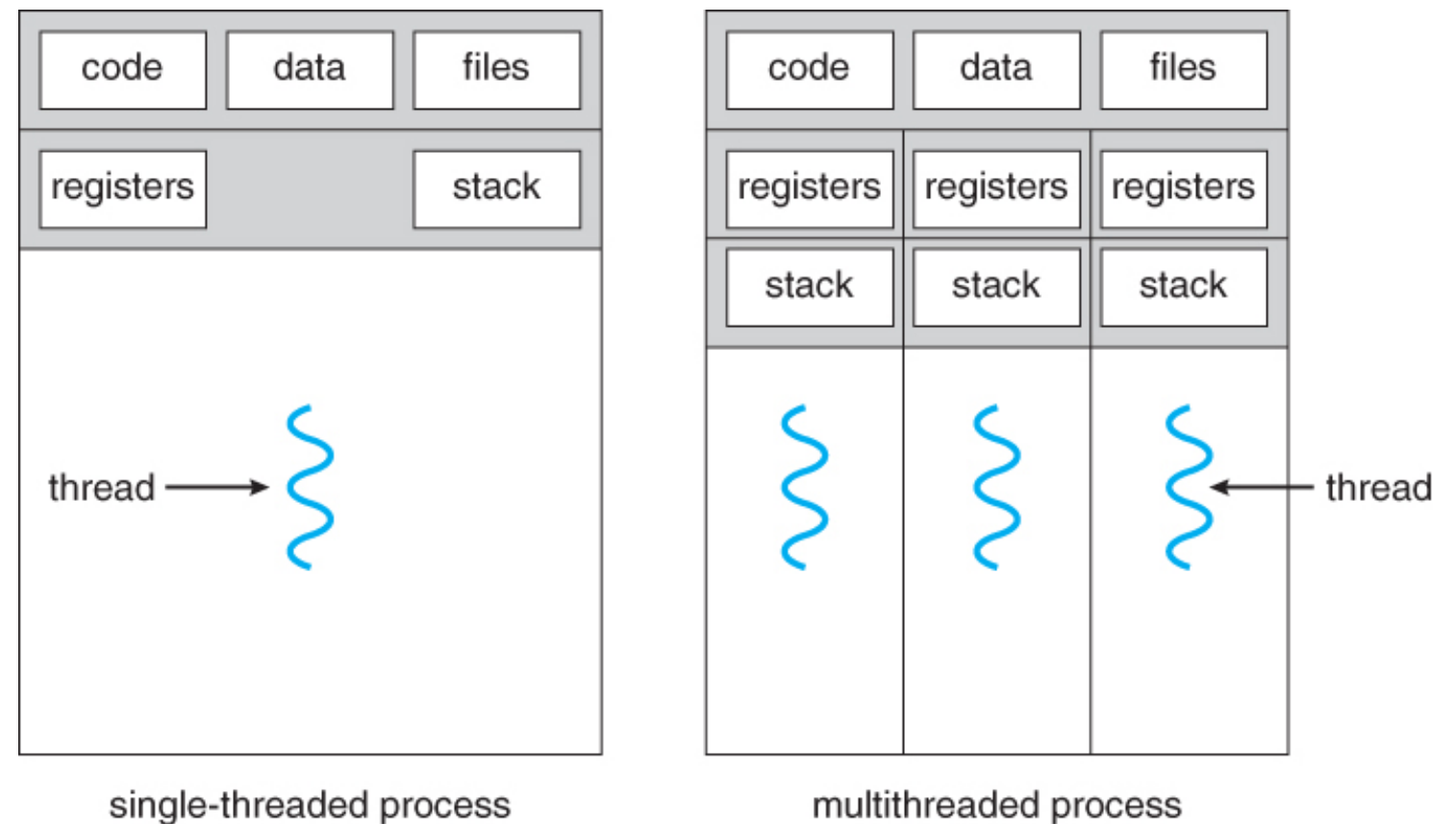
Διαχείριση Νημάτων (Threads) & Ζητήματα συγχρονισμού

Ζητήματα Συγχρονισμού Νημάτων:

Ο συγχρονισμός είναι κρίσιμος όταν πολλά νήματα προσπαθούν να εκτελούν ενέργειες σε κοινόχρηστους πόρους. Οι κριτικές περιοχές είναι οι τμήματα κώδικα όπου τα νήματα αλληλεπιδρούν με κοινόχρηστους πόρους, και αν δεν υπάρχει σωστός συγχρονισμός, μπορεί να προκύψουν προβλήματα όπως τα **race conditions**.

Μηχανισμοί Συγχρονισμού:

1. **Semaphores:** Είναι μεταβλητές που χρησιμοποιούνται για τον έλεγχο της πρόσβασης στους πόρους. Χρησιμοποιούνται για να εξασφαλίσουν ότι ένα μόνο νήμα μπορεί να εκτελείται σε μια κρίσιμη περιοχή κάθε φορά.
2. **Mutexes (Mutual Exclusion):** Μοναδικά κλειδιά που εμποδίζουν πολλαπλά νήματα να εισέλθουν ταυτόχρονα σε μια κρίσιμη περιοχή.
3. **Monitors:** Παρέχουν έναν πιο δομημένο τρόπο συγχρονισμού.



Προβλήματα Deadlock και Ζητήματα Συγχρονισμού

Ένα **deadlock** συμβαίνει όταν δύο ή περισσότερες διεργασίες ή νήματα περιμένουν ο ένας τον άλλο να ελευθερώσει έναν πόρο, με αποτέλεσμα καμία διεργασία να μην μπορεί να προχωρήσει.

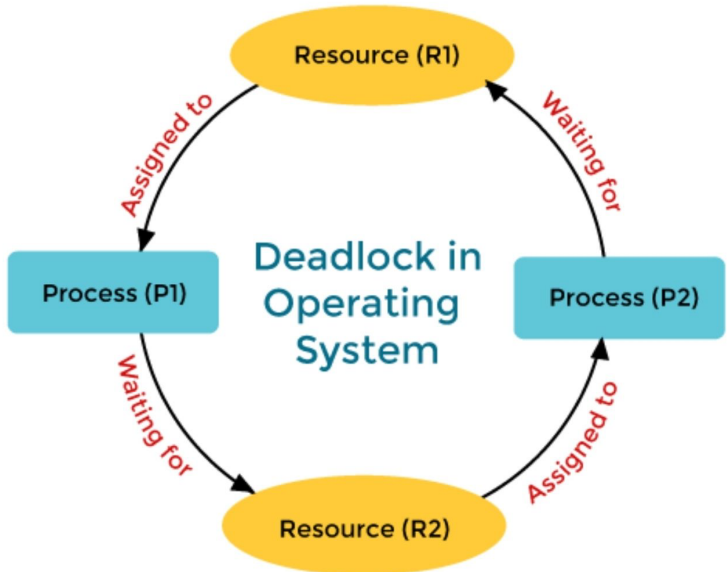
Συνθήκες Deadlock:

- Αμοιβαίος Αποκλεισμός (Mutual Exclusion):** Ένας πόρος μπορεί να χρησιμοποιηθεί μόνο από μία διεργασία κάθε φορά.
- Κατακράτηση και Αναμονή (Hold and Wait):** Μια διεργασία κατέχει έναν πόρο και περιμένει άλλον.
- Καμία Προαίρεση (No Preemption):** Ένας πόρος δεν μπορεί να αφαιρεθεί από τη διεργασία που τον κατέχει.
- Κυκλική Αναμονή (Circular Wait):** Υπάρχει μια κυκλική αλυσίδα διεργασιών, όπου η καθεμία περιμένει έναν πόρο που κατέχει η επόμενη.

Στρατηγικές Αποφυγής Deadlock:

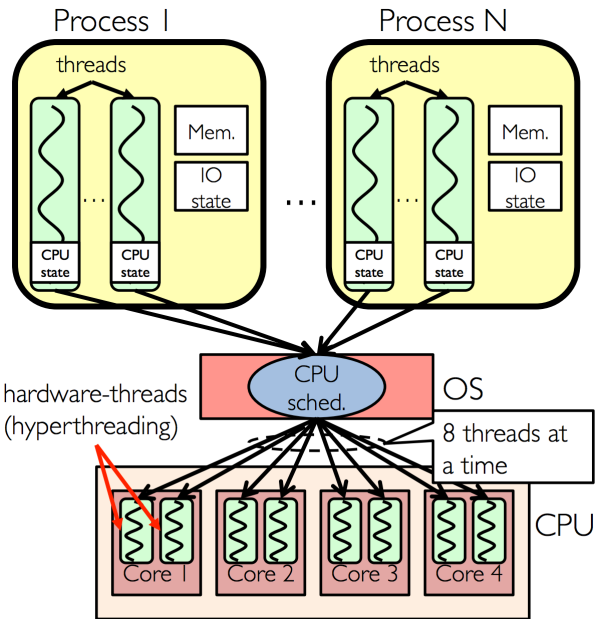
- Αποφυγή:** Χρήση αλγορίθμων όπως ο **Αλγόριθμος του Τραπεζίτη** (Banker's Algorithm).

- Ανίχνευση και Ανάκτηση:** Το σύστημα ανιχνεύει deadlocks και ανακτά πόρους τερματίζοντας διεργασίες.
- Πρόληψη:** Εξασφάλιση ότι τουλάχιστον μία από τις συνθήκες του deadlock δεν ισχύει.



Πολυνηματικός Προγραμματισμός και Πολυπύρνα Συστήματα

Ο πολυνηματικός προγραμματισμός (multithreading) επιτρέπει τη βελτίωση της απόδοσης των εφαρμογών σε συστήματα με πολλούς πυρήνες. Στα συστήματα αυτά, τα νήματα μπορούν να διαμοιράζονται μεταξύ πολλαπλών πυρήνων, επιτρέποντας την εκτέλεση πολλαπλών εργασιών ταυτόχρονα.



Διάλεξη 04

Διαχείριση Μνήμης

Στόχοι:

- Κατανόηση της βασικής έννοιας της μνήμης και του ρόλου της στη λειτουργία του υπολογιστή.
- Ανάλυση της διαχείρισης φυσικής και εικονικής μνήμης.
- Κατανόηση της σελιδοποίησης και της διαχείρισης της μνήμης σε πολυπρογραμματισμένα συστήματα.
- Εξέταση των αλγορίθμων αντικατάστασης σελίδων και των τεχνικών κατανομής μνήμης.
- Ανάλυση της διαχείρισης μνήμης σε διαφορετικά λειτουργικά συστήματα.

Η διαχείριση μνήμης είναι μια ζωτική και πολύπλοκη εργασία του λειτουργικού συστήματος. Επιτρέπει την εκτέλεση πολλών διεργασιών ταυτόχρονα χωρίς διακοπές.

Η γνώση του τρόπου λειτουργίας της διαχείρισης μνήμης στα ΛΣ είναι ζωτικής σημασίας για τη σταθερότητα του συστήματος και τη βελτίωση της απόδοσης του συστήματος.

Οι διευθύνσεις μνήμης είναι ζωτικής σημασίας για τη διαχείριση της μνήμης στα ΛΣ. Μια διεύθυνση μνήμης είναι ένα μοναδικό αναγνωριστικό για μια συγκεκριμένη θέση μνήμης ή αποθήκευσης. Οι διευθύνσεις βοηθούν στην εύρεση και πρόσβαση σε πληροφορίες που είναι αποθηκευμένες στη μνήμη.

Η διαχείριση μνήμης παρακολουθεί κάθε θέση μνήμης, χαρτογραφεί διευθύνσεις και διαχειρίζεται το χώρο διευθύνσεων της μνήμης. Τα διαφορετικά περιβάλλοντα απαιτούν διαφορετικούς τρόπους αναφοράς στις θέσεις διευθύνσεων μνήμης.

Οι δύο κύριοι τύποι διευθύνσεων μνήμης επεξηγούνται στις παρακάτω ενότητες.

Κάθε τύπος έχει διαφορετικό ρόλο στη διαχείριση της μνήμης και εξυπηρετεί διαφορετικό σκοπό.

Φυσικές Διευθύνσεις

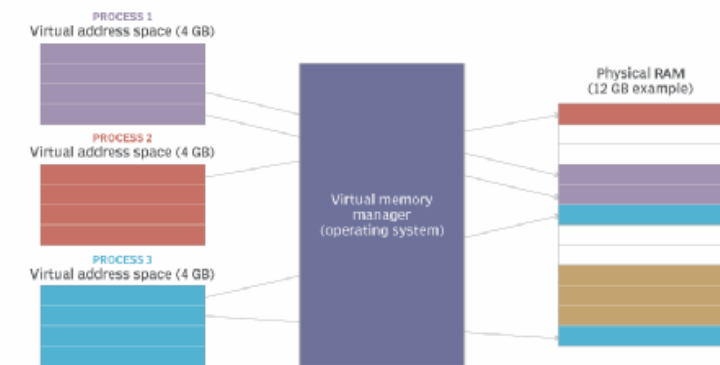
Μια φυσική διεύθυνση είναι ένα αριθμητικό αναγνωριστικό που δείχνει μια θέση φυσικής μνήμης. Η διεύθυνση αντιπροσωπεύει την πραγματική θέση των δεδομένων στο υλικό και είναι ζωτικής σημασίας για τη διαχείριση της μνήμης χαμηλού επιπέδου.

Τα στοιχεία υλικού όπως η CPU ή ο ελεγκτής μνήμης χρησιμοποιούν φυσικές διευθύνσεις. Οι διευθύνσεις είναι μοναδικές και έχουν σταθερές τοποθεσίες, επιτρέποντας στο υλικό να εντοπίζει γρήγορα τυχόν δεδομένα. Οι φυσικές διευθύνσεις δεν είναι διαθέσιμες στα προγράμματα χρηστών.

Εικονικές Διευθύνσεις

Μια εικονική διεύθυνση είναι μια διεύθυνση που δημιουργείται από πρόγραμμα που αντιπροσωπεύει μια αφαίρεση φυσικής μνήμης. Κάθε διεργασία χρησιμοποιεί τον χώρο

The physical RAM after the OS organizes 3 processes



Διεργασία διαχείρισης μνήμης

διευθύνσεων εικονικής μνήμης ως αποκλειστική μνήμη.

Οι εικονικές διευθύνσεις δεν ταιριάζουν με τις θέσεις φυσικής μνήμης. Τα προγράμματα διαβάζουν και δημιουργούν εικονικές διευθύνσεις, χωρίς να γνωρίζουν τον φυσικό χώρο διευθύνσεων. Η κύρια μονάδα μνήμης (MMU) αντιστοιχίζει εικονικές σε φυσικές διευθύνσεις για να εξασφαλίσει έγκυρη πρόσβαση στη μνήμη.

Μονάδα Διαχείρισης Μνήμης (MMU)

Η μονάδα διαχείρισης μνήμης (MMU) βρίσκεται στην Κεντρική Μονάδα Επεξεργασίας. Ανήκει δηλαδή στο υλικό του υπολογιστή και όχι στο λογισμικό και είναι υπεύθυνη για την αντιστοίχιση και την μετάφραση των εικονικών διευθύνσεων σε φυσικές. Άλλες δευτερεύουσες λειτουργίες της είναι η προστασία μνήμης, ο έλεγχος της κρυφής μνήμης και η διαχείριση του διαύλου (bus).

Τεχνικές διαχείρισης μνήμης:

Στατική καταχώρηση: Ολόκληρη η μνήμη εκχωρείται σε ένα πρόγραμμα πριν από την εκτέλεσή του.

Δυναμική καταχώρηση: Η μνήμη εκχωρείται σε τμήματα κατά τη διάρκεια της εκτέλεσης των προγραμμάτων, ανάλογα με τις ανάγκες τους.

Σελιδοποίηση: Η μνήμη και η εικονική μνήμη διαιρούνται σε σελίδες σταθερού μεγέθους.

Τεμαχισμός: Η μνήμη διαιρείται σε τμήματα μεταβλητού μεγέθους, προσαρμοσμένα στις ανάγκες κάθε προγράμματος.

Στόχοι διαχείρισης μνήμης:

Αποτελεσματικότητα: Να διασφαλίζεται η μέγιστη δυνατή χρήση της διαθέσιμης μνήμης, ώστε να εκτελούνται όσο το δυνατόν περισσότερα προγράμματα ταυτόχρονα.

Αποδοτικότητα: Να ελαχιστοποιείται ο χρόνος πρόσβασης στη μνήμη, τόσο στη φυσική όσο και στην εικονική, για βέλτιστη απόδοση του συστήματος.

Ακεραιότητα: Να διασφαλίζεται η απομόνωση των προγραμμάτων μεταξύ τους, αποτρέποντας την πρόσβαση ενός προγράμματος στα δεδομένα ή τον κώδικα ενός άλλου.

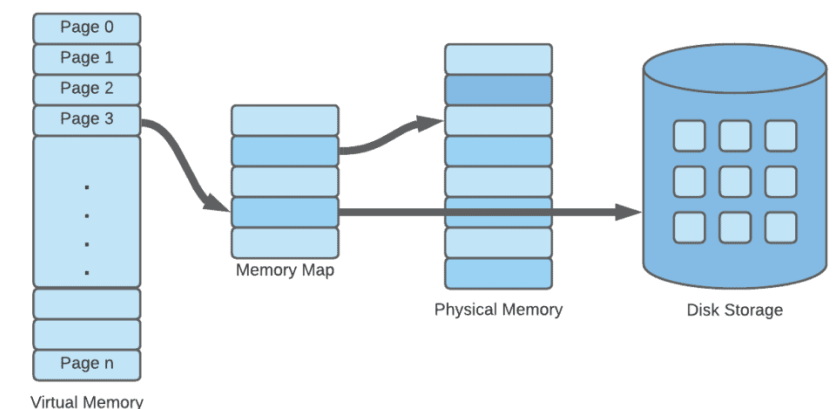
Προστασία: Να προστατεύεται η μνήμη από κακόβουλο λογισμικό ή σφάλματα προγραμματισμού που θα μπορούσαν να οδηγήσουν σε καταστροφή δεδομένων ή αστάθεια του συστήματος.

Πίνακες σελίδων

Η λειτουργία της MMU υλοποιείται με την χρήση των πινάκων σελίδων (page tables).

Σε μία απλή υλοποίηση, η χαρτογράφηση των εικονικών διευθύνσεων σε φυσικές διευθύνσεις μπορεί να συνοψιστεί ως εξής: Η εισερχόμενη εικονική διεύθυνση διαιρείται σε έναν

αριθμό εικονικής σελίδας (bit υψηλής τάξης) και μια σχετική διεύθυνση (bit χαμηλής τάξης). Ο αριθμός της εικονικής σελίδας χρησιμοποιείται ως αριθμοδείκτης θέσης στον πίνακα σελίδων προκειμένου να εντοπίζεται η καταχώριση της συγκεκριμένης εικονικής σελίδας. Από την καταχώριση του πίνακα σελίδων υπολογίζεται ο αριθμός του πλαισίου σελίδας (αν υπάρχει). Ο αριθμός πλαισίου σελίδας προσαρτάται στα αριστερά της σχετικής διεύθυνσης (πιο σημαντικά bit) αντικαθιστώντας τον αριθμό εικονικής σελίδας, ώστε να σχηματίσει μια φυσική διεύθυνση που μπορεί πλέον να σταλεί στη μνήμη. Από μαθηματική σκοπιά, ο πίνακας σελίδων είναι μια συνάρτηση, με όρισμα τον αριθμό εικονικής σελίδας και αποτέλεσμα τον αριθμό φυσικού πλαισίου.



Διαχείριση μνήμης

Δομή Μνήμης

Δομή Μνήμης:

- **Μνήμη RAM (Random Access Memory):** Προσωρινή μνήμη υψηλής ταχύτητας, όπου τα δεδομένα αποθηκεύονται προσωρινά για ταχύτερη πρόσβαση από τον επεξεργαστή.
- **Cache Memory:** Ακόμη πιο γρήγορη από τη RAM, χρησιμοποιείται από τον επεξεργαστή για την αποθήκευση συχνά χρησιμοποιούμενων δεδομένων και εντολών.
- **Αποθηκευτικά Μέσα (Storage Devices):** Όπως οι σκληροί δίσκοι (HDD), οι μονάδες στερεάς κατάστασης (SSD), που χρησιμοποιούνται για μόνιμη αποθήκευση δεδομένων.

Εικονική Μνήμη

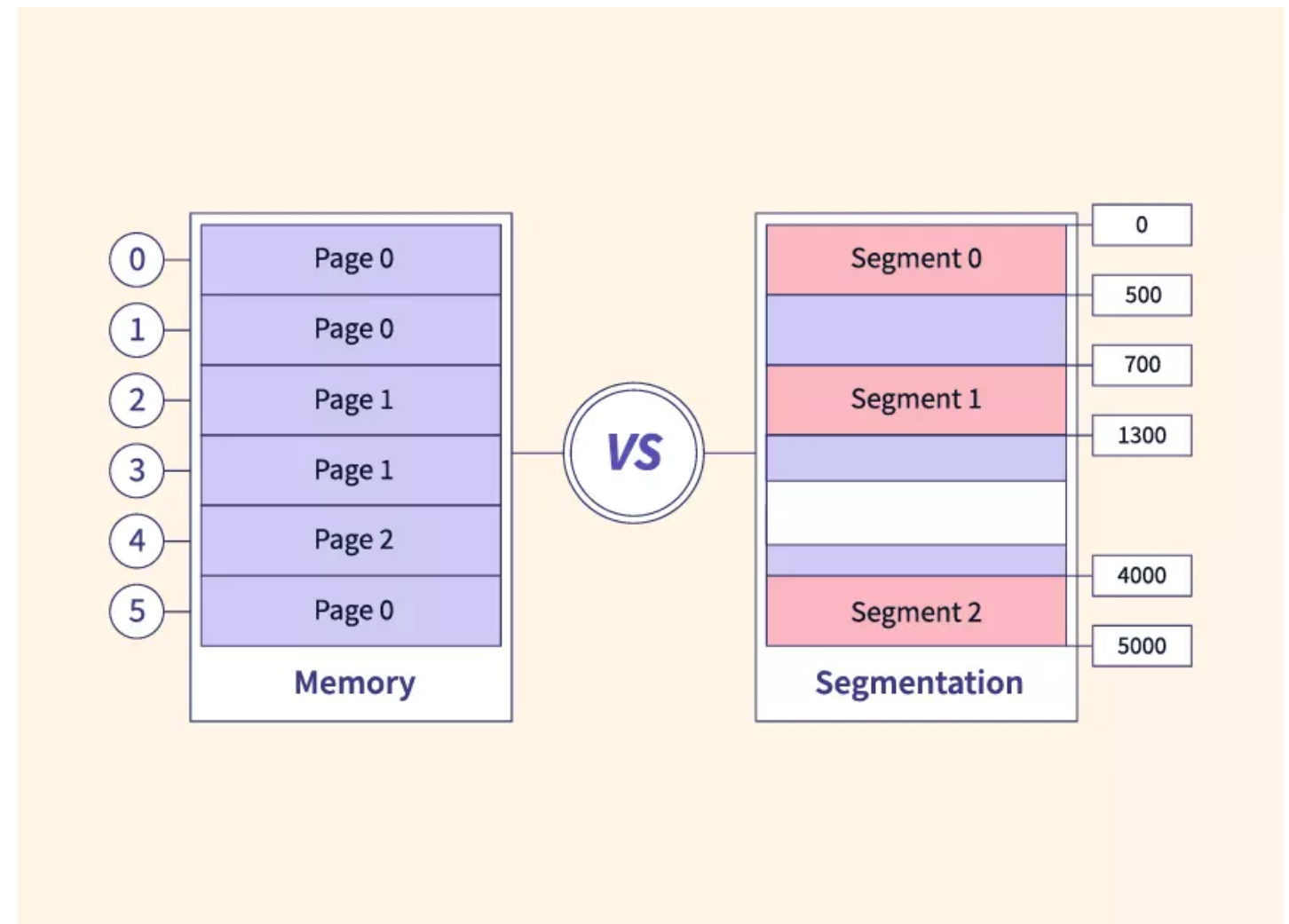
Η **εικονική μνήμη** επιτρέπει σε ένα σύστημα να εκτελεί προγράμματα που απαιτούν περισσότερο χώρο μνήμης από αυτόν που προσφέρει η φυσική μνήμη. Η βασική ιδέα είναι ότι το λειτουργικό σύστημα διαχωρίζει τη λογική από τη φυσική διεύθυνση, επιτρέποντας στα προγράμματα να «νομίζουν» ότι έχουν άπλετο χώρο μνήμης.

Πώς λειτουργεί η Εικονική Μνήμη:

1. **Σελιδοποίηση (Paging):** Χωρίζει τη μνήμη σε μικρά κομμάτια, γνωστά ως σελίδες. Αυτές οι σελίδες μπορούν να τοποθετηθούν οπουδήποτε στη φυσική μνήμη.

2. Κατάτμηση (Segmentation):

Χωρίζει τη μνήμη σε μεγαλύτερα, λογικά τμήματα που αντιστοιχούν στις ανάγκες του προγράμματος.



Σελιδοποίηση ή/και κατάτμηση μνήμης

Δομή Μνήμης

Σελιδοποίηση (Paging)

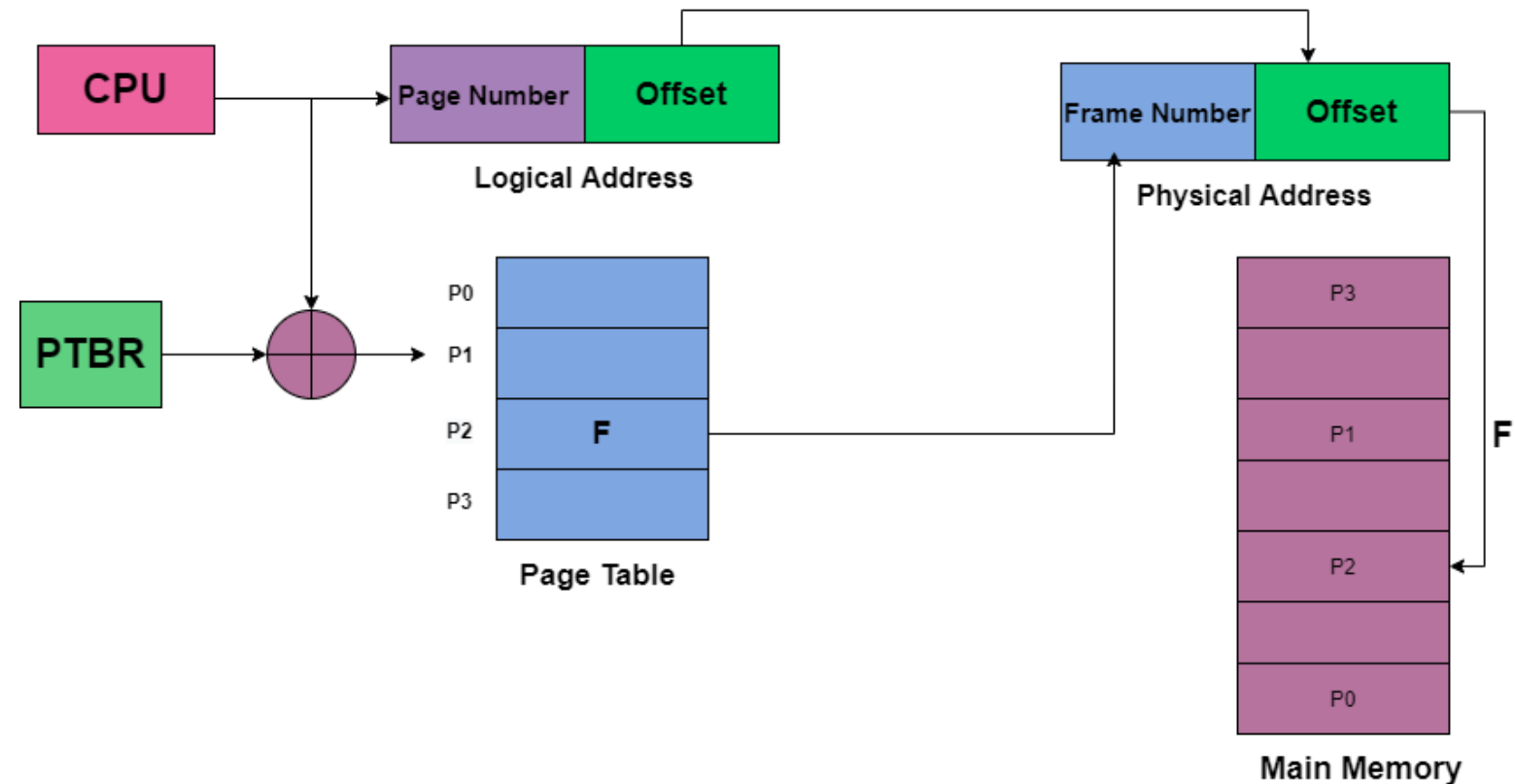
Η **σελιδοποίηση** είναι η πιο δημοφιλής μέθοδος διαχείρισης μνήμης σε σύγχρονα λειτουργικά συστήματα. Η μνήμη χωρίζεται σε μικρές μονάδες, που ονομάζονται σελίδες, και κάθε σελίδα μπορεί να βρίσκεται σε διαφορετική θέση στη φυσική μνήμη.

Πλεονεκτήματα της Σελιδοποίησης:

- Δεν χρειάζεται συνεχόμενη κατανομή μνήμης, κάτι που διευκολύνει τη χρήση της.
- Η διαχείριση της μνήμης γίνεται πιο αποδοτική, καθώς χρησιμοποιούνται μόνο οι απαιτούμενες σελίδες.

Μηχανισμοί Σελιδοποίησης:

- **Πίνακες Σελίδων (Page Tables):** Χρησιμοποιούνται για να αποθηκεύουν τις αντιστοιχίες μεταξύ εικονικών και φυσικών διευθύνσεων.
- **MMU (Memory Management Unit):** Μονάδα που διαχειρίζεται την αντιστοίχιση των διευθύνσεων.



Σελιδοποίηση μνήμης (λιγότερο χρόνο)

Πολιτικές Αντικατάστασης Σελίδων:

Όταν η μνήμη γεμίζει, το λειτουργικό σύστημα πρέπει να αντικαταστήσει σελίδες για να κάνει χώρο για νέες. Υπάρχουν διάφοροι αλγόριθμοι για την αντικατάσταση σελίδων:

1. LRU (Least Recently Used):

Αντικαθιστά την πιο παλιά σελίδα, δηλαδή αυτή που έχει χρησιμοποιηθεί για την τελευταία φορά πριν από όλες τις άλλες.

2. FIFO (First In, First Out):

Αντικαθιστά την παλαιότερη σελίδα που εισήλθε στο σύστημα.

3. Optimal Page Replacement:

Αντικαθιστά τη σελίδα που δεν θα χρησιμοποιηθεί για το μεγαλύτερο χρονικό διάστημα στο μέλλον.

Δομή Μνήμης

Κατάτμηση (Segmentation)

Η **κατάτμηση** είναι μια εναλλακτική τεχνική διαχείρισης μνήμης που χωρίζει τη μνήμη σε μεγαλύτερα, λογικά τμήματα. Κάθε τμήμα αντιστοιχεί σε έναν συγκεκριμένο τύπο δεδομένων ή λειτουργία του προγράμματος (π.χ., κώδικας, δεδομένα).

Πλεονεκτήματα Κατάτμησης:

- Παρέχει μεγαλύτερη ευελιξία σε σχέση με τη σελιδοποίηση, καθώς μπορεί να προσαρμόζεται στις ανάγκες του προγράμματος.
- Δίνει τη δυνατότητα για πιο αποτελεσματική κατανομή της μνήμης για προγράμματα που έχουν διαφοροποιημένες ανάγκες.

Μειονεκτήματα Κατάτμησης:

- Η ανάγκη για συνεχόμενη μνήμη μπορεί να οδηγήσει σε θραύσματα μνήμης (fragmentation).

Αντικατάσταση Σελίδων (Page Replacement)

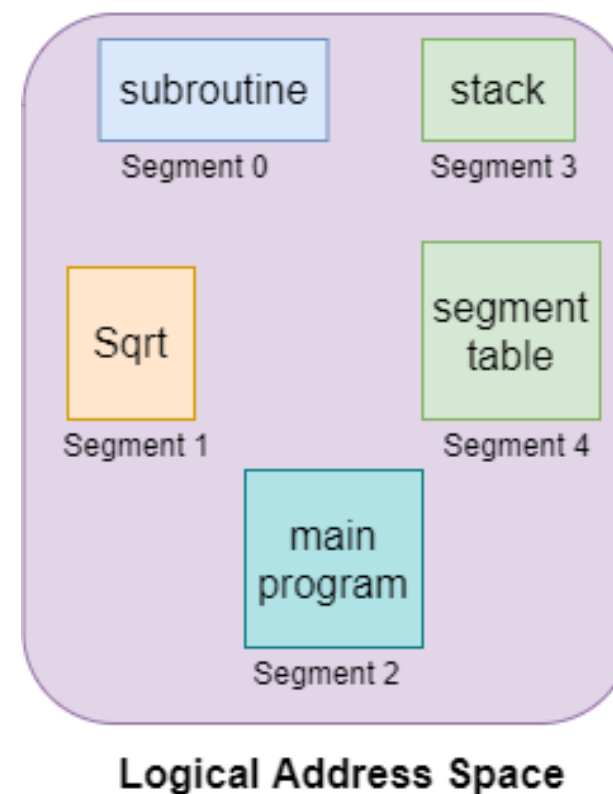
Ένα από τα πιο σημαντικά ζητήματα στη διαχείριση της μνήμης είναι πώς να επιλέγονται οι σελίδες που θα αντικατασταθούν όταν η μνήμη γεμίζει. Υπάρχουν πολλοί αλγόριθμοι

αντικατάστασης σελίδων, όπως αναφέρθηκαν παραπάνω.

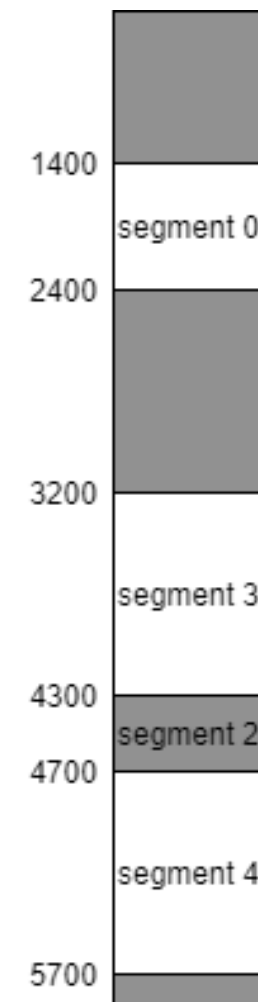
Προκλήσεις της Αντικατάστασης:

- **Thrashing:** Συμβαίνει όταν το σύστημα ξοδεύει περισσότερο χρόνο για να μεταφέρει σελίδες εντός και εκτός της μνήμης παρά για την

εκτέλεση του προγράμματος. Αυτό μπορεί να συμβεί όταν η μνήμη είναι ανεπαρκής για την εκτέλεση πολλών προγραμμάτων ταυτόχρονα.



Segment Table	
	limit
	base
0	1000
1	400
2	400
3	1100



Κατάτμηση μνήμης (λιγότερο χώρο)

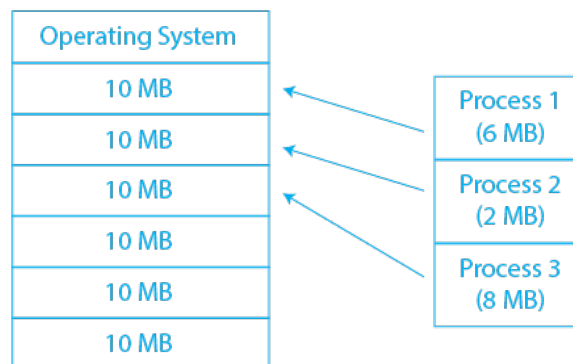
Κατανομή Μνήμης

Πολιτικές Κατανομής Μνήμης

Για τη βελτίωση της αποδοτικότητας στη χρήση της μνήμης, τα λειτουργικά συστήματα χρησιμοποιούν διαφορετικές πολιτικές κατανομής μνήμης.

Τύποι Πολιτικών Κατανομής:

1. **First-Fit Allocation:** Η πρώτη διαθέσιμη περιοχή μνήμης χρησιμοποιείται για την εκτέλεση της διεργασίας.
2. **Best-Fit Allocation:** Η μικρότερη δυνατή περιοχή μνήμης που μπορεί να χωρέσει τη διεργασία επιλέγεται, μειώνοντας τον κατακερματισμό.
3. **Worst-Fit Allocation:** Η μεγαλύτερη διαθέσιμη περιοχή μνήμης χρησιμοποιείται, ώστε να μειωθεί η πιθανότητα μικρών περιοχών που δεν μπορούν να χρησιμοποιηθούν.

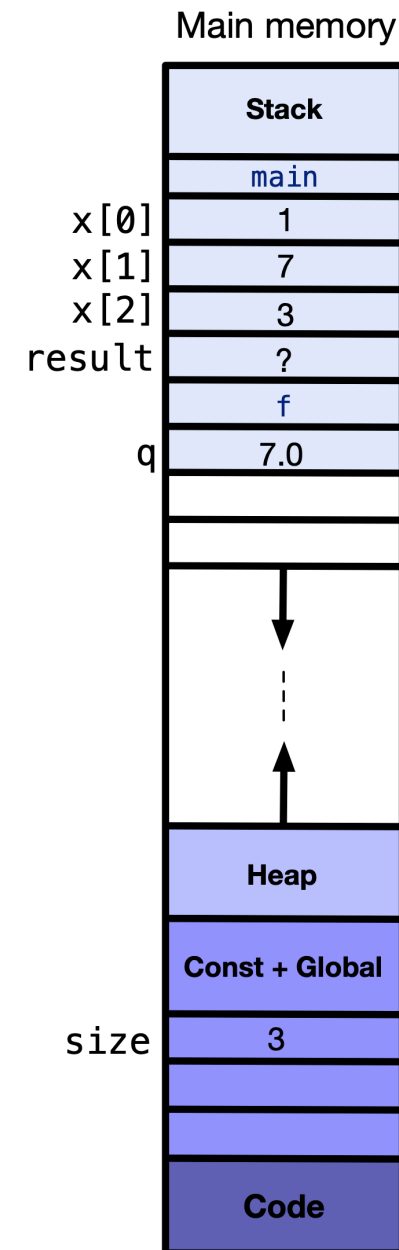


Διαχείριση Μνήμης σε Πολυπρογραμματισμένα Συστήματα

Στα **πολυπρογραμματισμένα συστήματα**, όπου πολλά προγράμματα εκτελούνται ταυτόχρονα, η αποδοτική διαχείριση της μνήμης είναι ακόμη πιο κρίσιμη. Το λειτουργικό σύστημα πρέπει να διασφαλίσει ότι οι διεργασίες δεν έρχονται σε σύγκρουση για τους διαθέσιμους πόρους μνήμης.

Προσεγγίσεις:

- **Fixed Partitioning:** Κάθε διεργασία έχει προκαθορισμένο χώρο μνήμης.
- **Dynamic Partitioning:** Ο χώρος μνήμης που κατανέμεται στις διεργασίες μπορεί να μεταβάλλεται δυναμικά, ανάλογα με τις ανάγκες του συστήματος.



```
1 #include <stdio.h>
2 double f(int[]);
3
4 int main(void) {
5     int x[3] = {1, 7, 3};
6     double result = f(x);
7     return 0;
8 }
9 double f(int list[]) {
10     double q = 7.0;
11     return q;
12 }
```

Κατανομή μνήμης

Διαχείριση Μνήμης σε γνωστά λειτουργικά συστήματα

Διαχείριση Μνήμης σε Διαφορετικά Λειτουργικά Συστήματα

Διάφορα λειτουργικά συστήματα εφαρμόζουν διαφορετικές στρατηγικές διαχείρισης μνήμης, ανάλογα με τις ανάγκες τους.

1. Linux:

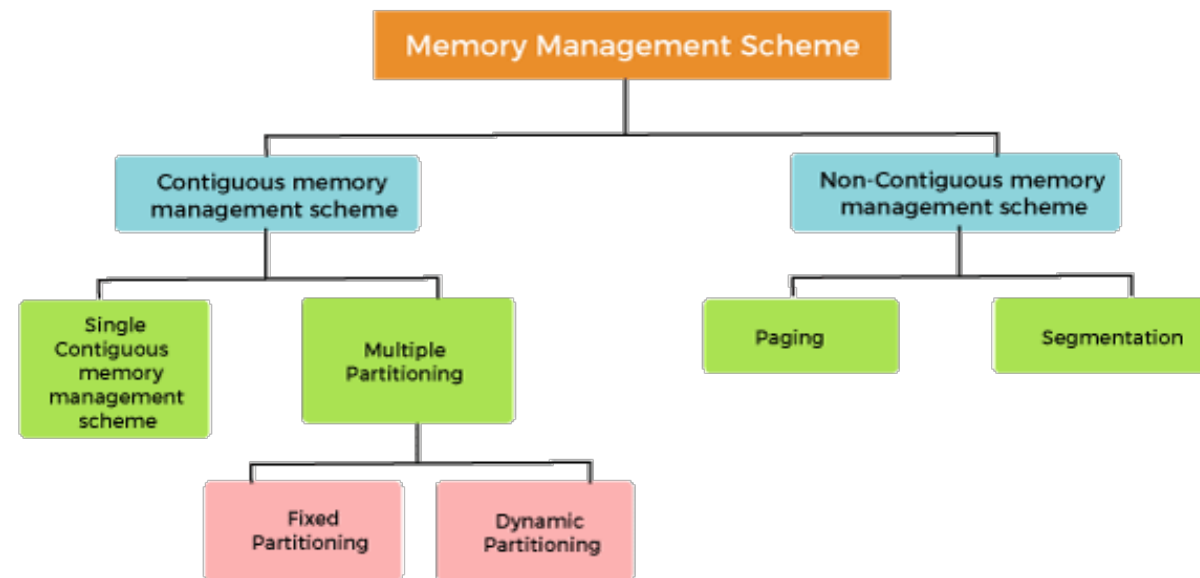
Το Linux χρησιμοποιεί κυρίως **σελιδοποίηση**, αλλά έχει επίσης υποστήριξη για κατάτμηση σε μερικές περιπτώσεις. Η διαχείριση της μνήμης γίνεται μέσω του **virtual memory manager** που διαχειρίζεται τις σελίδες και τους πίνακες σελίδων.

2. Windows:

Τα Windows χρησιμοποιούν συνδυασμό **σελιδοποίησης** και **κατάτμησης**. Το λειτουργικό σύστημα διαχειρίζεται τα διαμερίσματα μνήμης για εφαρμογές και συστήματα με δυναμικό τρόπο.

3. macOS:

Το macOS χρησιμοποιεί **σελιδοποίηση** για τη διαχείριση της μνήμης, με αυξημένη έμφαση στη διαχείριση των **cache** και τη βελτιστοποίηση της απόδοσης για συσκευές με περιορισμένους πόρους μνήμης, όπως τα laptops.



Classification of memory management schemes

Διαχείριση μνήμης

Διάλεξη 05

Συστήματα Αρχείων και Αποθήκευσης (File Systems and Storage)

Στόχοι:

- Κατανόηση των βασικών δομών των συστημάτων αρχείων.
- Ανάλυση των λειτουργιών των συστημάτων αρχείων και των καταλόγων.
- Διαχείριση του χώρου αποθήκευσης και της ακεραιότητας των δεδομένων.
- Βελτιστοποίηση της απόδοσης των συστημάτων αρχείων.
- Παραδείγματα εφαρμογής συστημάτων αρχείων σε διαφορετικά λειτουργικά συστήματα.

Εισαγωγή στα Συστήματα Αρχείων

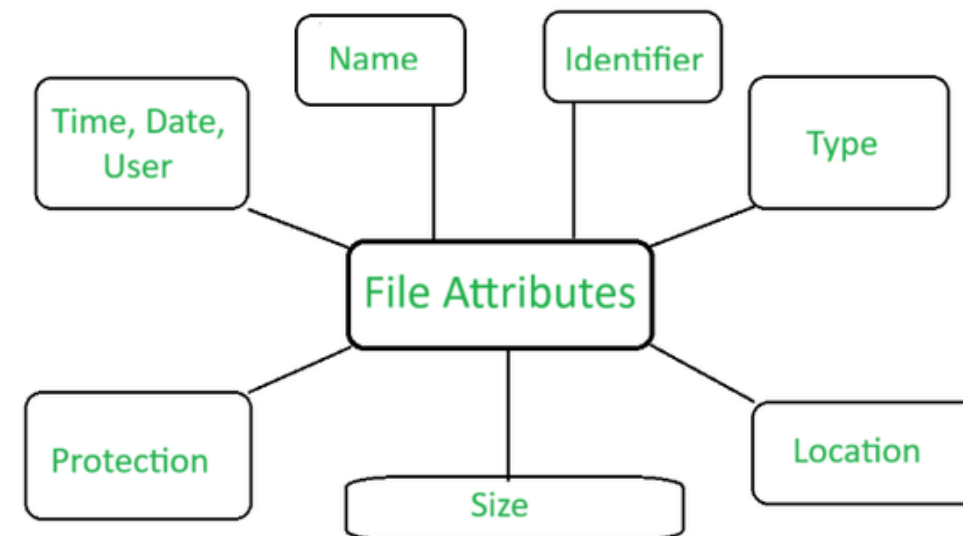
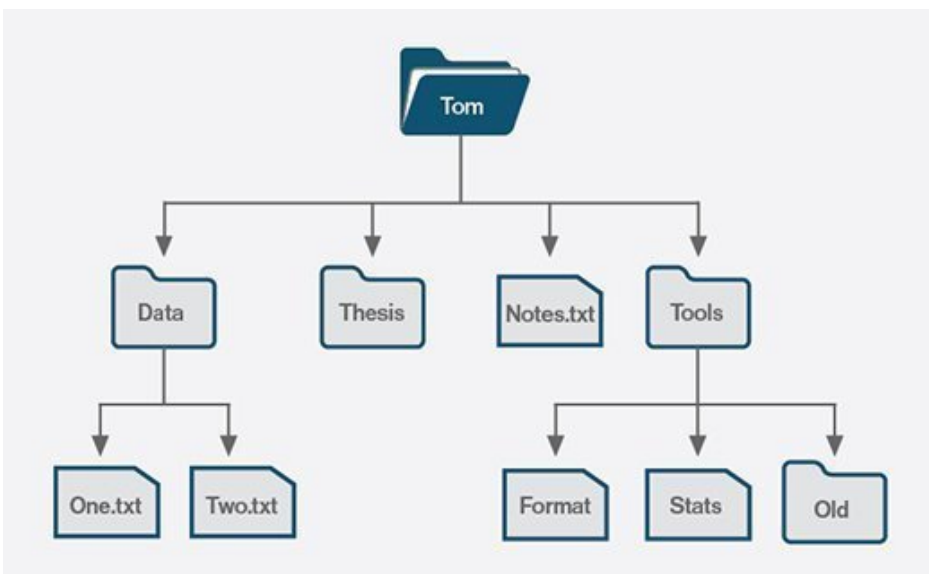
Τα συστήματα αρχείων (file systems)

αποτελούν την ουσία της διαχείρισης των δεδομένων και των αρχείων σε έναν υπολογιστή. Ο ρόλος τους είναι να αποθηκεύουν, να οργανώνουν και να ανακτούν δεδομένα από τις συσκευές αποθήκευσης, όπως σκληρούς δίσκους, μονάδες SSD και άλλες συσκευές αποθήκευσης. Η βασική λειτουργία ενός συστήματος αρχείων είναι να μεταφράζει τις λογικές διευθύνσεις που χρησιμοποιούν τα προγράμματα στις φυσικές διευθύνσεις των συσκευών αποθήκευσης.

Δομή του Συστήματος Αρχείων:

Η δομή ενός συστήματος αρχείων χωρίζεται σε δύο βασικές περιοχές:

Δεδομένα Χρήστη: Περιλαμβάνει τα αρχεία και τους καταλόγους που αποθηκεύονται από τον χρήστη.



File attributes

Μετα-δεδομένα (Metadata):

Πληροφορίες που περιγράφουν τα αρχεία, όπως το μέγεθος, η θέση αποθήκευσης και τα δικαιώματα πρόσβασης.

Τα συστήματα αρχείων μπορούν να διαφέρουν σημαντικά σε λειτουργίες, μορφές και αποδοτικότητα ανάλογα με το λειτουργικό σύστημα και τις απαιτήσεις του χρήστη.

Λειτουργίες Συστήματος Αρχείων

Τα συστήματα αρχείων εκτελούν πολλές βασικές λειτουργίες, συμπεριλαμβανομένης της δημιουργίας, διαχείρισης και ανάκτησης αρχείων. Ας δούμε τις κύριες λειτουργίες που σχετίζονται με τα αρχεία και τους καταλόγους.

Δημιουργία και Διαχείριση Αρχείων

Τα αρχεία είναι οι κύριες οντότητες που αποθηκεύονται σε ένα σύστημα αρχείων. Ένα αρχείο περιέχει δεδομένα, τα οποία μπορεί να είναι οποιουδήποτε τύπου, όπως κείμενο, εικόνες, ή κώδικας. Κάθε αρχείο έχει τα εξής χαρακτηριστικά:

Όνομα αρχείου: Αντιπροσωπεύει την ταυτότητα του αρχείου.

Δομή: Τα αρχεία μπορούν να είναι απλά (flat) ή πιο σύνθετα, περιέχοντας δομές δεδομένων.

Δικαιώματα πρόσβασης: Καθορίζουν ποιος μπορεί να διαβάσει, να γράψει ή να εκτελέσει το αρχείο.

Η δημιουργία αρχείων περιλαμβάνει τη δέσμευση χώρου αποθήκευσης και την καταγραφή των απαραίτητων μετα-δεδομένων. Κάθε φορά που ένα αρχείο δημιουργείται, το σύστημα αρχείων πρέπει να βρει διαθέσιμο χώρο στη συσκευή

αποθήκευσης και να ενημερώσει τις δομές μετα-δεδομένων.

Κατανομή Χώρου Αρχείων

Η κατανομή του χώρου αποθήκευσης είναι μία από τις βασικότερες λειτουργίες που εκτελεί το σύστημα αρχείων. Υπάρχουν διάφορες στρατηγικές κατανομής:

Συνεχής Κατανομή (Contiguous Allocation):

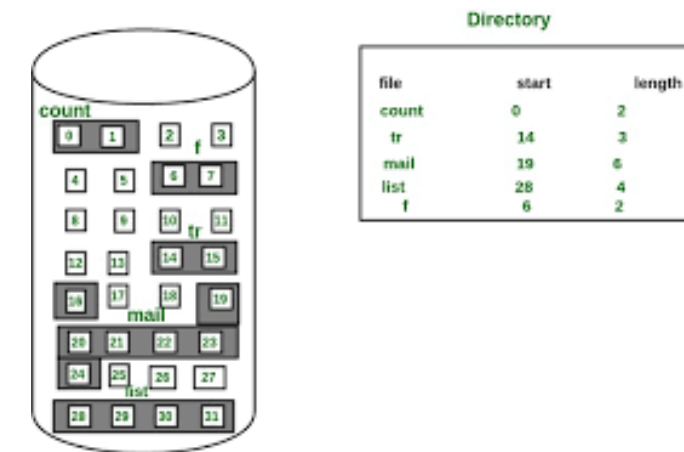
Κάθε αρχείο καταλαμβάνει μια συνεχόμενη περιοχή στο δίσκο. Είναι απλή στη χρήση, αλλά μπορεί να οδηγήσει σε θραύσματα μνήμης (fragmentation).

Κατακερματισμένη Κατανομή (Linked Allocation):

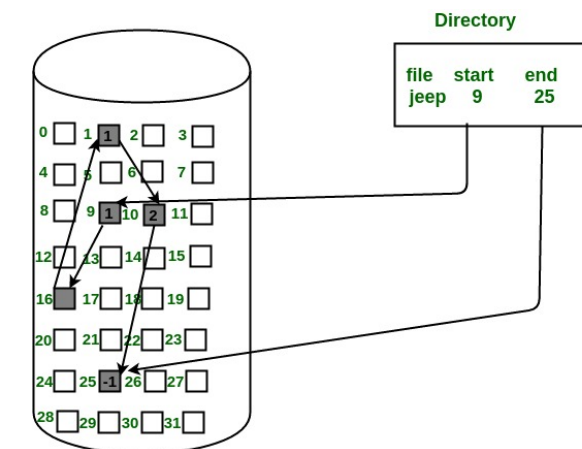
Τα αρχεία κατακερματίζονται σε μπλοκ, τα οποία συνδέονται με δείκτες. Αυτό επιλύει το πρόβλημα της θραύσης, αλλά μπορεί να οδηγήσει σε χαμηλή απόδοση λόγω της ανάγκης για πολλές προσπελάσεις.

Ευρετηριασμένη Κατανομή (Indexed Allocation):

Χρησιμοποιεί έναν πίνακα ευρετηρίου για κάθε αρχείο, όπου κάθε δείκτης αντιστοιχεί σε ένα μπλοκ του δίσκου. Είναι πιο αποδοτική για μεγάλα αρχεία.

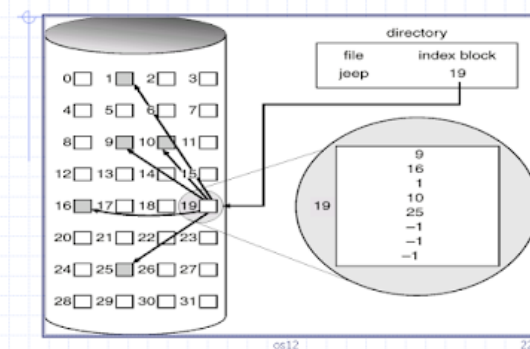


Συνεχής Κατανομή (Contiguous Allocation)



Κατακερματισμένη Κατανομή (Linked Allocation)

Example of Indexed Allocation



Συνεχής Κατανομή (Contiguous Allocation)

Διαχείριση Καταλόγων

Οι **κατάλογοι (directories)** είναι δομές που οργανώνουν τα αρχεία σε μια ιεραρχία. Κάθε κατάλογος περιέχει καταχωρίσεις για τα αρχεία και υποκαταλόγους του, οι οποίες περιλαμβάνουν το όνομα του αρχείου και μια αναφορά στη θέση αποθήκευσής του. Η διαχείριση καταλόγων περιλαμβάνει:

Δημιουργία καταλόγων: Οργανώνει τα αρχεία σε λογικές ομάδες.

Πλοήγηση: Επιτρέπει στους χρήστες να βρίσκουν και να διαχειρίζονται αρχεία.

Προστασία: Ελέγχει ποιος μπορεί να έχει πρόσβαση στους καταλόγους και τα αρχεία που περιέχονται σε αυτούς.

Δομές Καταλόγων:

Υπάρχουν διάφορες δομές καταλόγων που χρησιμοποιούνται από τα συστήματα αρχείων:

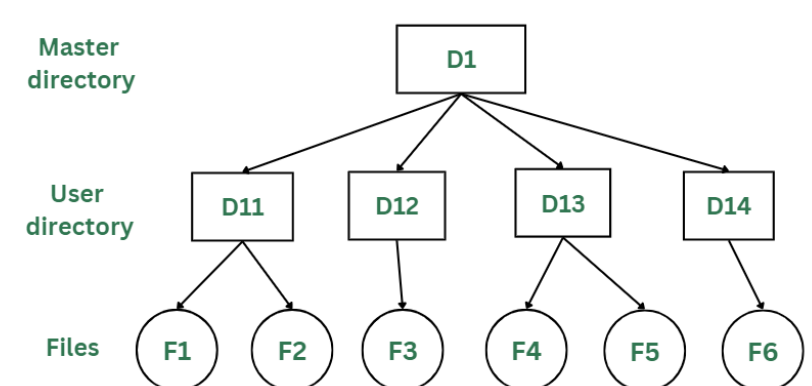
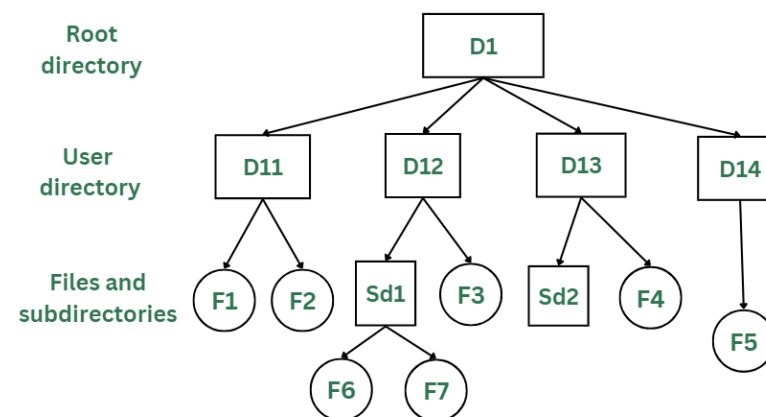
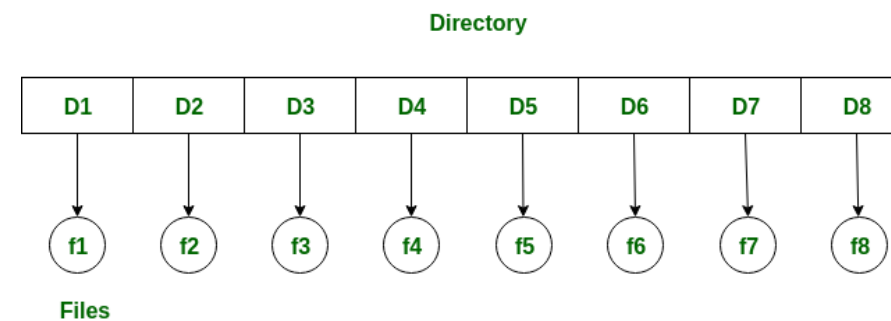
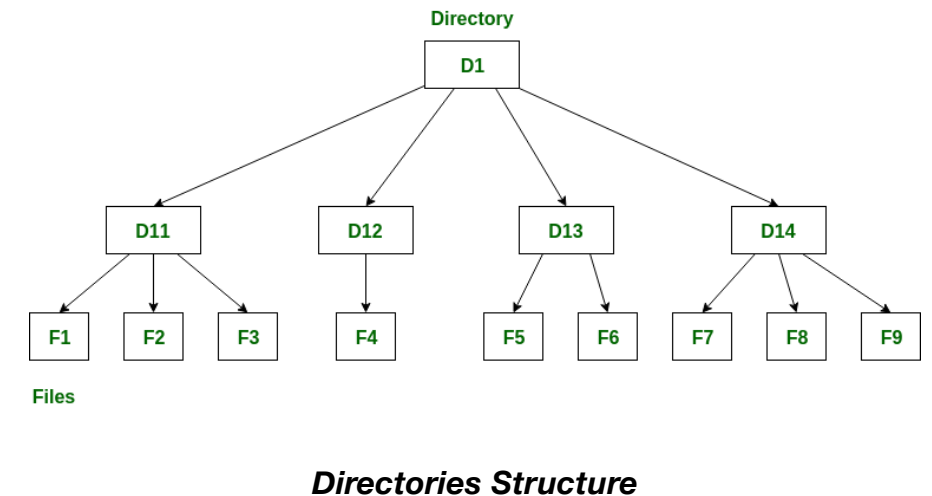
Μονοεπίπεδοι κατάλογοι (Single-level Directory): Όλα τα αρχεία τοποθετούνται σε ένα επίπεδο, χωρίς διαχωρισμό. Παρέχουν απλότητα, αλλά είναι δύσκολο να οργανωθούν μεγάλα συστήματα αρχείων.

Ιεραρχικοί κατάλογοι (Hierarchical Directory): Τα αρχεία οργανώνονται σε μια

ιεραρχική δομή με καταλόγους και υποκαταλόγους. Αυτή είναι η πιο κοινή δομή στα σύγχρονα λειτουργικά συστήματα.

Συστήματα Διπλών Καταλόγων (Two-level Directory Systems):

Χρησιμοποιούν ξεχωριστούς καταλόγους για κάθε χρήστη ή εφαρμογή, προσφέροντας μεγαλύτερη οργάνωση και ευκολία στη διαχείριση.



Διαχείριση Χώρου Αποθήκευσης

Η διαχείριση του χώρου αποθήκευσης

είναι κρίσιμη για την αποδοτική χρήση των συσκευών αποθήκευσης. Το σύστημα αρχείων πρέπει να παρακολουθεί τον διαθέσιμο χώρο και να τον κατανέμει σε νέα αρχεία. Υπάρχουν δύο κύριες μέθοδοι παρακολούθησης του χώρου αποθήκευσης:

Bitmaps

Τα **bitmaps** είναι πίνακες δυαδικών ψηφίων, όπου κάθε ψηφίο αντιπροσωπεύει ένα μπλοκ του δίσκου. Ένα 0 υποδεικνύει ότι το μπλοκ είναι ελεύθερο, ενώ ένα 1 δείχνει ότι είναι κατειλημμένο. Τα bitmaps προσφέρουν μια απλή και αποδοτική μέθοδο για την παρακολούθηση του χώρου.

Συνδεδεμένοι Κατάλογοι (Linked Lists)

Η μέθοδος αυτή χρησιμοποιεί συνδέσμους μεταξύ των ελεύθερων μπλοκ, δημιουργώντας μια λίστα με τα διαθέσιμα τμήματα του δίσκου. Αυτή η μέθοδος είναι πιο αποδοτική για συστήματα με μεγάλη ποικιλία σε μέγεθος αρχείων.

Free-Space Management

- File system maintains **free-space list** to track available blocks/clusters
- Bit vector or bit map** (n blocks)

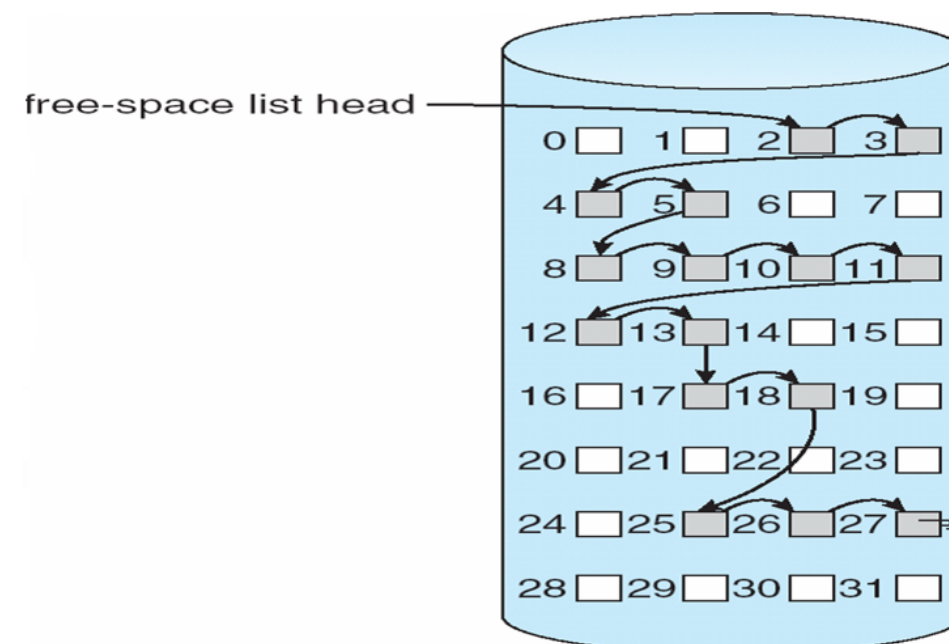
Number of the first free block is calculated as

$$\text{(number of bits per word)} * \text{(number of 0-value words)} + \text{offset of first 1 bit}$$

CPU's have instructions to return offset within word of first "1" bit

ECSE 2660 Computer Architecture, Networks, Operating Systems 4/26/2020 19

Bitmap



Free linked list

Αξιοπιστία και Ανάκτηση Δεδομένων

Η **αξιοπιστία** είναι σημαντική στα συστήματα αρχείων, καθώς τα δεδομένα πρέπει να είναι ασφαλή και προσβάσιμα, ακόμη και σε περίπτωση σφαλμάτων συστήματος. Τα σύγχρονα συστήματα αρχείων χρησιμοποιούν διάφορες τεχνικές για την προστασία των δεδομένων και την ανάκτηση σε περίπτωση σφαλμάτων.

Καταγραφή Συναλλαγών (Journaling)

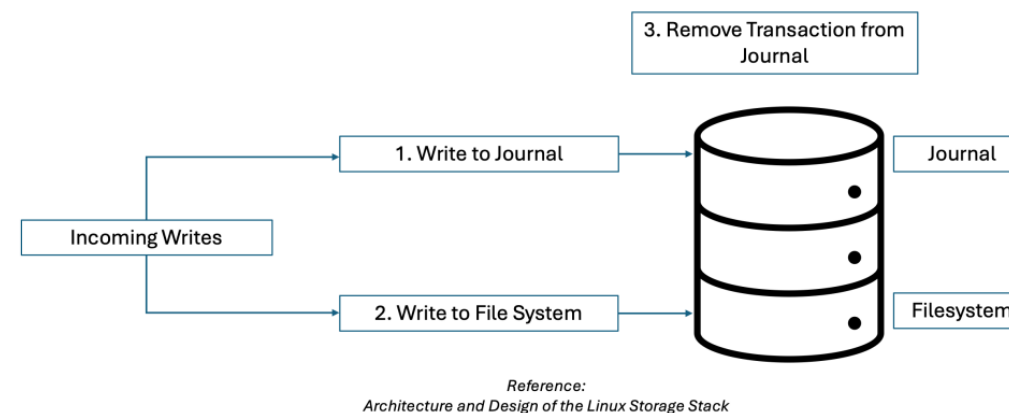
Το **journaling** είναι μια τεχνική που χρησιμοποιείται για να διασφαλιστεί ότι το σύστημα αρχείων παραμένει συνεπές, ακόμη και μετά από αποτυχίες συστήματος. Όλες οι αλλαγές καταγράφονται πρώτα σε ένα αρχείο καταγραφής (journal) πριν γίνουν οι πραγματικές αλλαγές στο σύστημα αρχείων. Σε περίπτωση σφάλματος, το σύστημα μπορεί να ανακτήσει τα δεδομένα χρησιμοποιώντας το journal.

Συστήματα Αντίγραφων Ασφαλείας (Backup Systems)

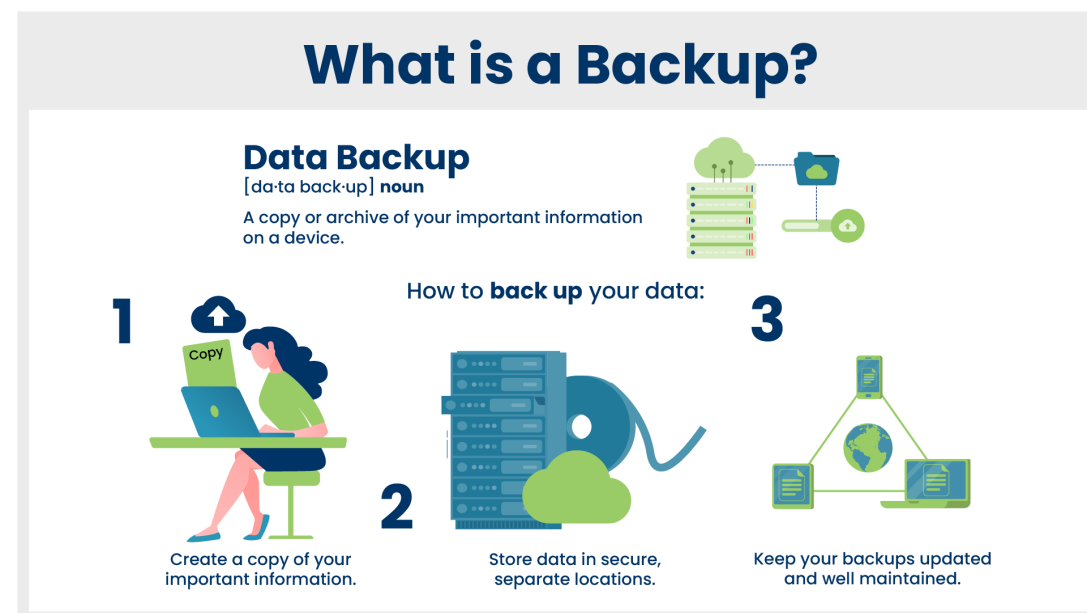
Η δημιουργία **αντιγράφων ασφαλείας** είναι ένα κρίσιμο κομμάτι της προστασίας δεδομένων. Τα αντίγραφα ασφαλείας μπορεί να είναι πλήρη ή αυξητικά, και

αποθηκεύονται σε διαφορετικές συσκευές ή τοποθεσίες για να

εξασφαλιστεί η ανάκτηση σε περίπτωση απώλειας δεδομένων.



Writing changes to journal



Backup

Βελτιστοποίηση Απόδοσης

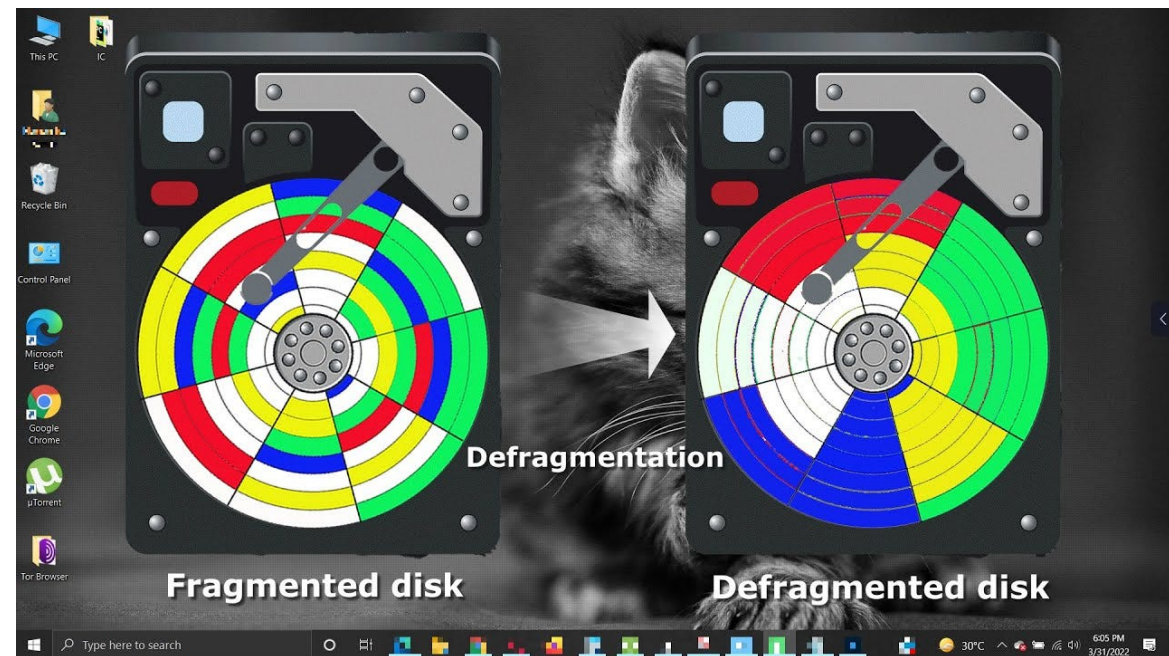
Η **απόδοση των συστημάτων αρχείων** μπορεί να βελτιωθεί με διάφορες τεχνικές, όπως:

Caching: Η χρήση μνήμης cache για την αποθήκευση δεδομένων που χρησιμοποιούνται συχνά.

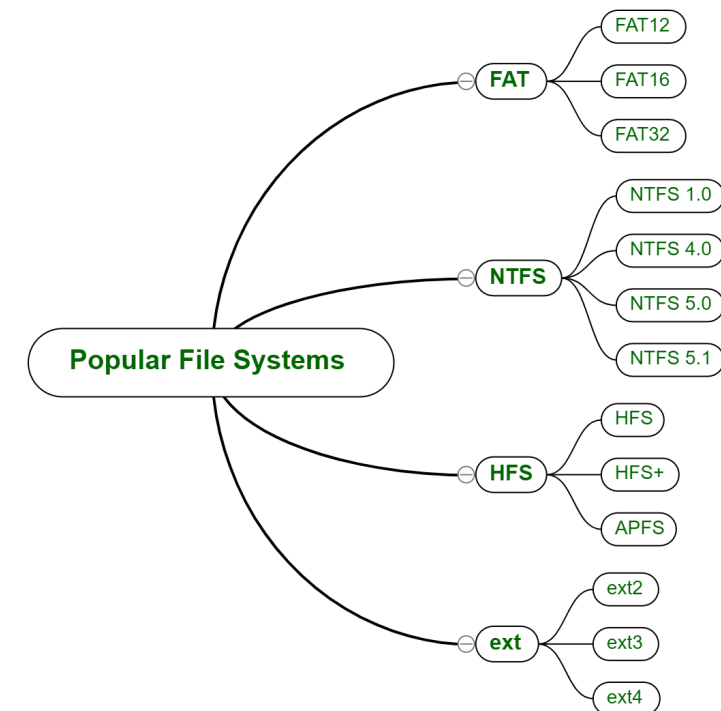
Defragmentation: Η διαδικασία αναδιοργάνωσης των δεδομένων στον δίσκο για να μειωθεί ο κατακερματισμός και να βελτιωθεί η απόδοση.

Συμπερασματικά:

Τα συστήματα αρχείων αποτελούν τη βάση για την αποθήκευση και την ανάκτηση δεδομένων σε έναν υπολογιστή. Μέσα από την κατανόηση των δομών, των λειτουργιών και των μεθόδων διαχείρισης χώρου, μπορούμε να βελτιστοποιήσουμε την αποδοτικότητα και την αξιοπιστία των συστημάτων μας. Από τις βασικές αρχές της κατανομής χώρου έως τις πιο εξελιγμένες τεχνικές όπως το journaling και τα αντίγραφα ασφαλείας, τα συστήματα αρχείων εξασφαλίζουν τη συνεπή και ασφαλή διαχείριση των δεδομένων.



Defragmentation



File systems

Παραδείγματα Συστημάτων Αρχείων

Διάφορα λειτουργικά συστήματα χρησιμοποιούν διαφορετικές υλοποιήσεις συστημάτων αρχείων για να εξυπηρετήσουν τις ανάγκες τους.

NTFS (Windows)

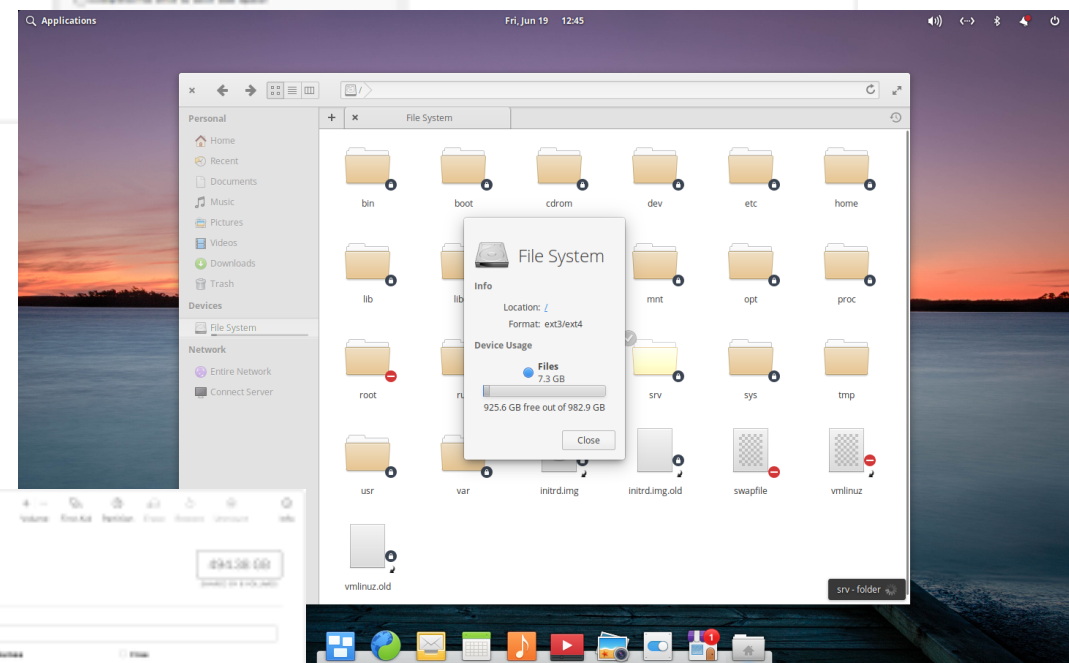
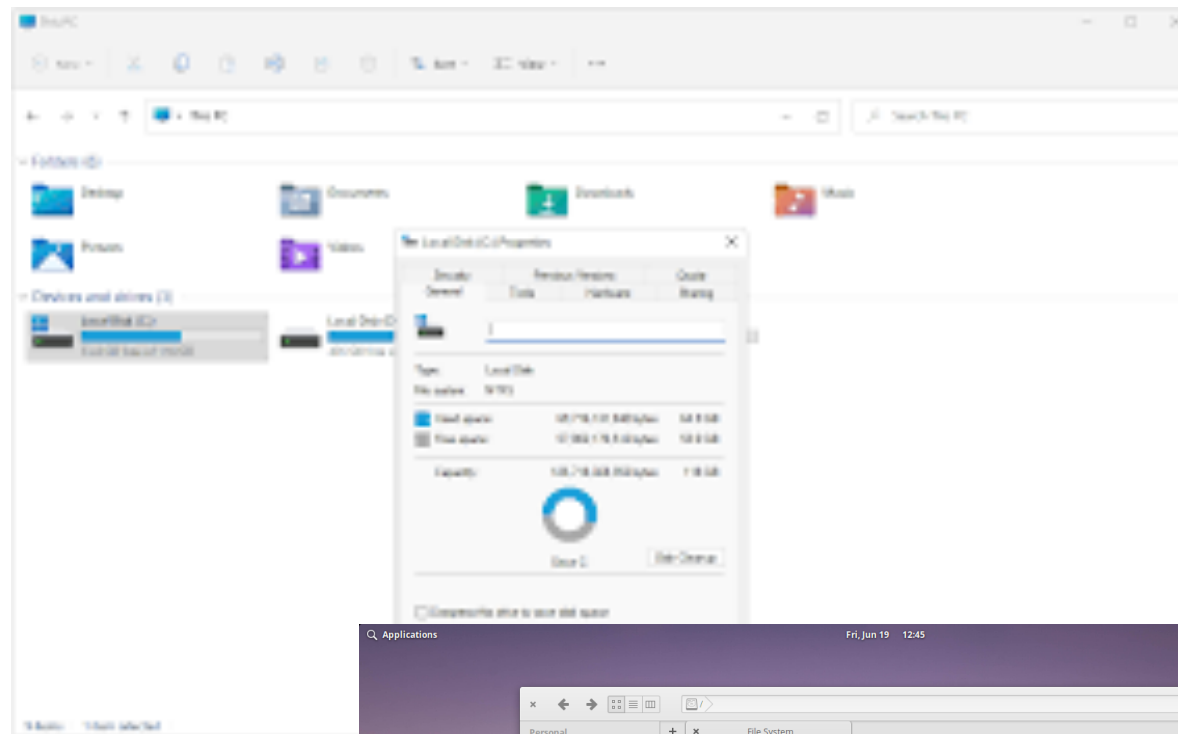
Το **NTFS (New Technology File System)** είναι το προεπιλεγμένο σύστημα αρχείων των Windows. Χρησιμοποιεί ευρετηριασμένη κατανομή και προσφέρει προηγμένες δυνατότητες, όπως υποστήριξη για μεγάλα αρχεία, συμπίεση και κρυπτογράφηση.

ext4 (Linux)

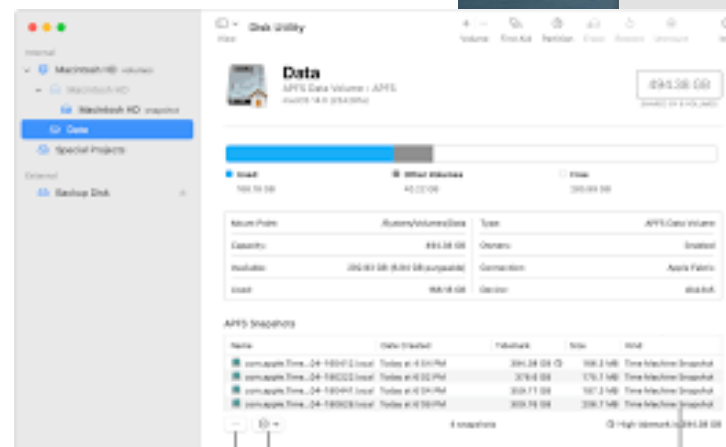
Το **ext4 (Extended File System)** είναι το προεπιλεγμένο σύστημα αρχείων των περισσότερων διανομών Linux. Είναι αξιόπιστο και υποστηρίζει μεγάλα μεγέθη αρχείων, ενώ περιλαμβάνει λειτουργίες journaling για την προστασία των δεδομένων.

APFS (macOS)

Το **APFS (Apple File System)** είναι το προεπιλεγμένο σύστημα αρχείων για macOS και iOS. Σχεδιάστηκε για σύγχρονες συσκευές αποθήκευσης όπως SSD, προσφέροντας αυξημένη ταχύτητα και βελτιωμένη διαχείριση χώρου.



ext4



Click to delete
a snapshot.

Click for
more options.

Snapshot
Browser

APFS

Διαχείριση Εισόδου/Εξόδου (I/O Management)

Στόχοι:

- Κατανόηση των βασικών αρχών της διαχείρισης εισόδου/εξόδου (I/O) στα λειτουργικά συστήματα.
- Ανάλυση των διαφόρων τύπων συσκευών I/O και των αντίστοιχων μεθόδων διαχείρισης.
- Κατανόηση των αρχών λειτουργίας και αλληλεπίδρασης συσκευών I/O με το λειτουργικό σύστημα.
- Εξέταση των στρατηγικών διαχείρισης και βελτιστοποίησης των συστημάτων I/O.
- Παρουσίαση των μηχανισμών οδήγησης συσκευών (device drivers) και της επικοινωνίας μεταξύ συσκευών και λογισμικού.

Εισαγωγή στη Διαχείριση Εισόδου/Εξόδου

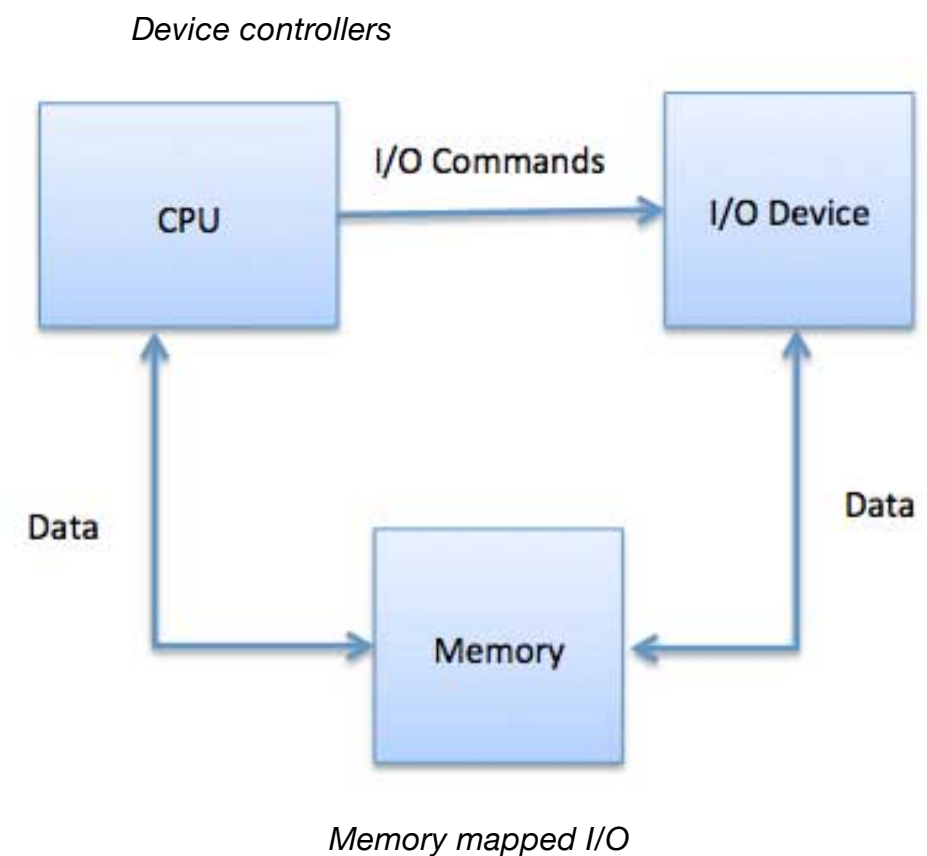
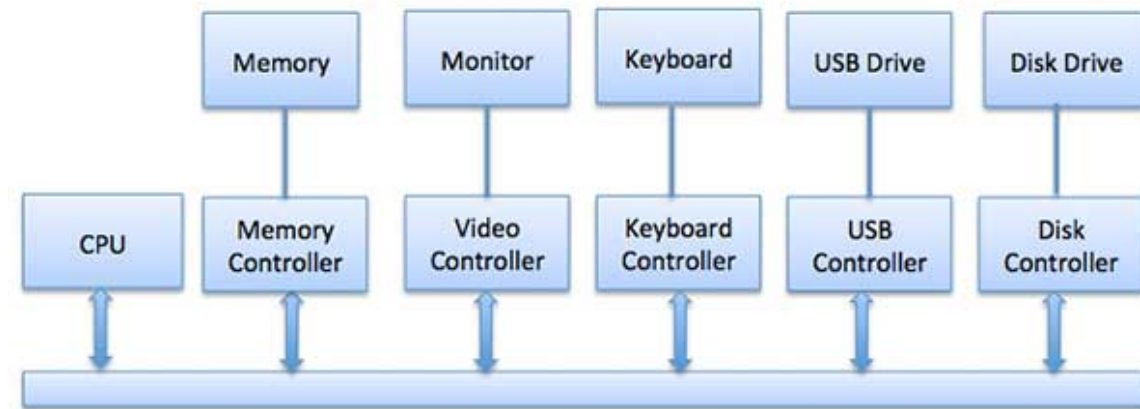
Τα συστήματα **εισόδου/εξόδου (I/O)**

αποτελούν ένα από τα πιο κρίσιμα τμήματα κάθε υπολογιστικού συστήματος. Το I/O αναφέρεται σε κάθε ενέργεια που εμπλέκει την ανταλλαγή δεδομένων μεταξύ του υπολογιστή και του εξωτερικού περιβάλλοντος μέσω συσκευών, όπως σκληροί δίσκοι, πληκτρολόγια, εκτυπωτές, οθόνες και δίκτυα.

Τα συστήματα I/O είναι υπεύθυνα για:

1. **Επικοινωνία με περιφερειακές συσκευές.**
2. **Μεταφορά δεδομένων μεταξύ της κύριας μνήμης και των συσκευών εισόδου/εξόδου.**
3. **Διαχείριση της ταχύτητας των συσκευών** (π.χ., μια συσκευή μπορεί να είναι πολύ πιο αργή από τη CPU).

Η διαχείριση του I/O είναι απαραίτητη για την εξασφάλιση της απόδοσης, της αξιοπιστίας και της σωστής λειτουργίας του συστήματος.



Συσκευές Εισόδου/Εξόδου

Οι **συσκευές εισόδου/εξόδου (I/O devices)** αποτελούν τον τρόπο με τον οποίο οι χρήστες και τα εξωτερικά συστήματα αλληλεπιδρούν με έναν υπολογιστή. Οι συσκευές αυτές ποικίλουν σε ταχύτητα, μέθοδο μετάδοσης δεδομένων και λειτουργία. Μπορούν να χωριστούν σε δύο κατηγορίες:

Συσκευές Μπλοκ (Block Devices)

Οι **συσκευές μπλοκ** διαχειρίζονται δεδομένα σε μονάδες που ονομάζονται μπλοκ, συνήθως 512 bytes ή 4096 bytes. Αυτές οι συσκευές επιτρέπουν την τυχαία πρόσβαση στα δεδομένα και χρησιμοποιούνται για αποθήκευση.

Παραδείγματα: Σκληροί δίσκοι, SSD, USB drives.

Συσκευές Χαρακτήρων (Character Devices)

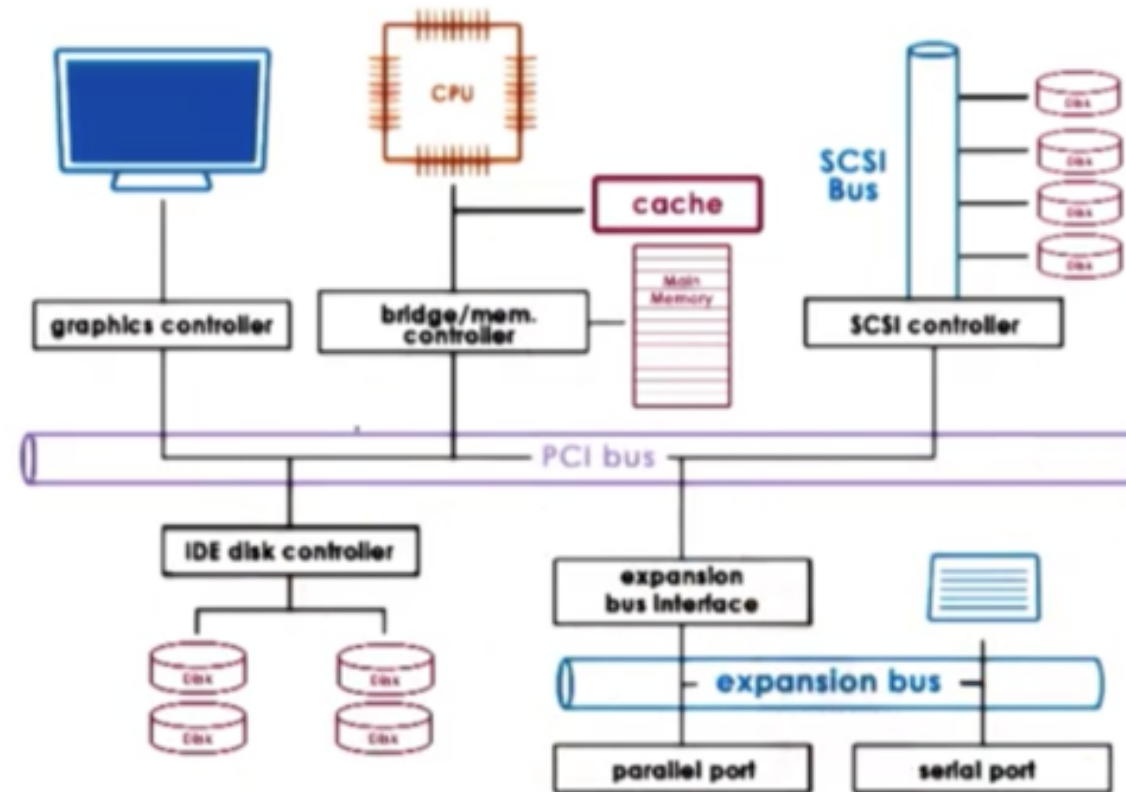
Οι **συσκευές χαρακτήρων** μεταφέρουν δεδομένα κατά byte και δεν υποστηρίζουν τυχαία πρόσβαση. Τα δεδομένα μεταφέρονται σε ροές και χρησιμοποιούνται συχνά για συσκευές που διαχειρίζονται είσοδο και έξοδο σε πραγματικό χρόνο.

Παραδείγματα: Πληκτρολόγια, σειριακές θύρες, εκτυπωτές.

Δίκτυα (Network Devices)

Οι **συσκευές δικτύου** διαχειρίζονται τη μετάδοση δεδομένων μέσω δικτύων και απαιτούν ειδική διαχείριση. Η απόδοση των δικτυακών συσκευών επηρεάζεται από διάφορους παράγοντες, όπως το εύρος ζώνης και η καθυστέρηση δικτύου.

Παραδείγματα: Κάρτες δικτύου, routers, switches.



I/O Devices

Επικοινωνία Μεταξύ Λειτουργικού Συστήματος και Συσκευών I/O

Το λειτουργικό σύστημα χρειάζεται να επικοινωνεί με τις συσκευές I/O για να ελέγχει και να διαχειρίζεται τη λειτουργία τους. Υπάρχουν τρεις κύριοι τρόποι επικοινωνίας μεταξύ των συσκευών I/O και του λειτουργικού συστήματος:

Ανίχνευση Κατάστασης (Polling)

Η **ανίχνευση κατάστασης (polling)** είναι η μέθοδος με την οποία το λειτουργικό σύστημα ελέγχει επαναλαμβανόμενα την κατάσταση μιας συσκευής για να δει αν είναι έτοιμη για ανάγνωση ή εγγραφή. Αν και είναι εύκολο να υλοποιηθεί, η ανίχνευση κατάστασης μπορεί να είναι αναποτελεσματική καθώς καταναλώνει σημαντικούς πόρους επεξεργαστή, ιδιαίτερα σε αργές συσκευές.

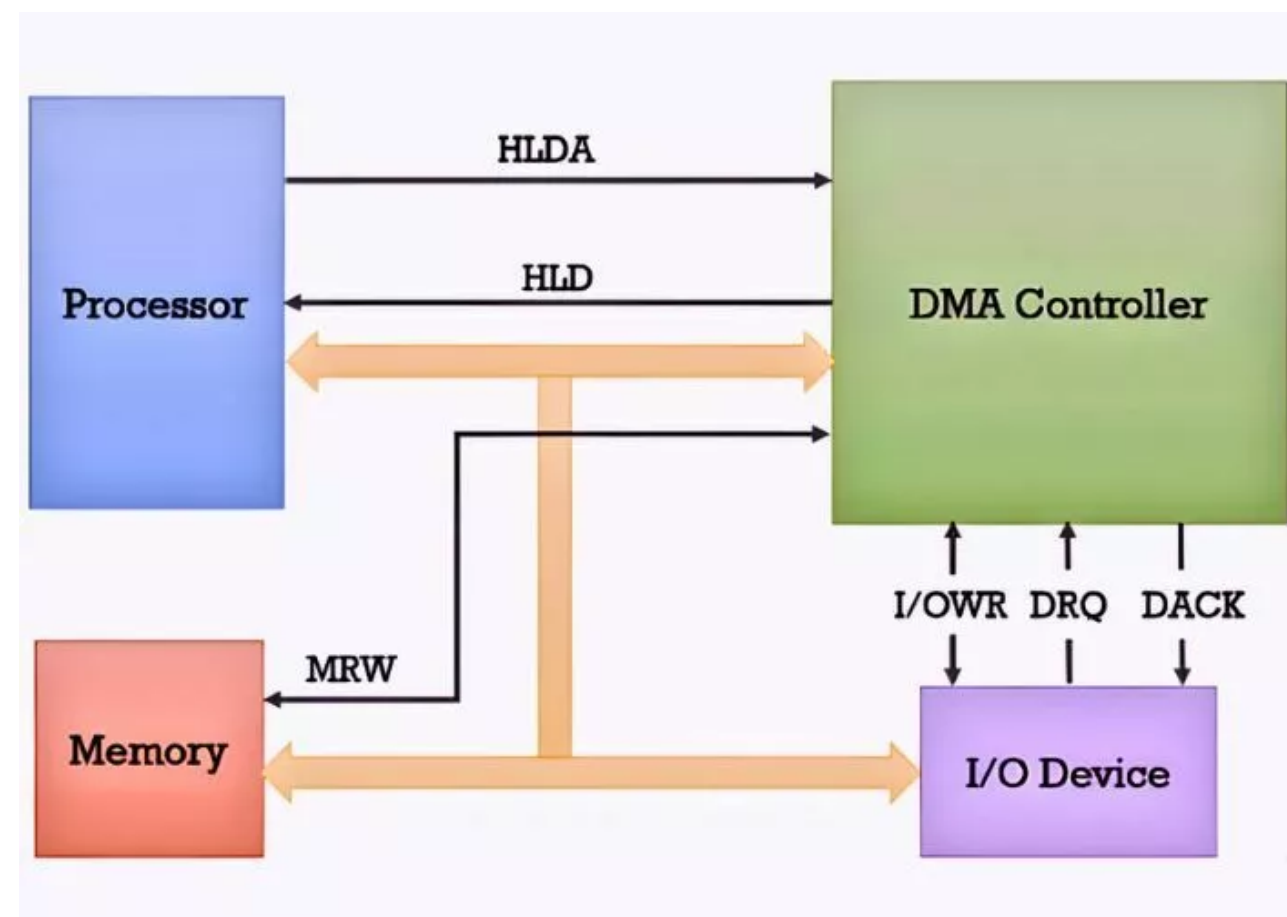
Διακοπές (Interrupts)

Οι **διακοπές (interrupts)** είναι μια πιο αποδοτική μέθοδος επικοινωνίας, κατά την οποία η συσκευή ενημερώνει το λειτουργικό σύστημα μόλις είναι έτοιμη να στείλει ή να λάβει δεδομένα. Η μέθοδος αυτή επιτρέπει στην CPU να εκτελεί άλλες εργασίες μέχρι να λάβει το σήμα από τη συσκευή.

DMA (Direct Memory Access)

Το **DMA (Direct Memory Access)** επιτρέπει στις συσκευές να μεταφέρουν

δεδομένα απευθείας στη μνήμη του υπολογιστή χωρίς την άμεση συμμετοχή της CPU. Αυτή η μέθοδος βελτιώνει την απόδοση σε συστήματα που χρειάζονται μεγάλες ποσότητες δεδομένων να μεταφέρονται γρήγορα.



DMA

Επικοινωνία Μεταξύ Λειτουργικού Συστήματος και Συσκευών I/O

Οδηγοί Συσκευών (Device Drivers)

Οι **οδηγοί συσκευών (device drivers)** είναι ειδικά προγράμματα που επιτρέπουν στο λειτουργικό σύστημα να επικοινωνεί με τις συσκευές I/O. Οι οδηγοί δρουν ως διεπαφή ανάμεσα στο λειτουργικό σύστημα και τη συσκευή, μεταφράζοντας τις εντολές σε μορφή κατανοητή από τη συσκευή.

Λειτουργία Οδηγών Συσκευών

Οι οδηγοί συσκευών πρέπει να:

Διαχειρίζονται την επικοινωνία με τη συσκευή μέσω polling, interrupts ή DMA.

Μεταφράζουν τις λειτουργίες του λειτουργικού συστήματος σε εντολές που καταλαβαίνει η συσκευή.

Αντιμετωπίζουν σφάλματα και εξασφαλίζουν τη σωστή λειτουργία της συσκευής.

Διαφορετικοί Τύποι Οδηγών Συσκευών

Υπάρχουν οδηγοί για κάθε τύπο συσκευής I/O, όπως:

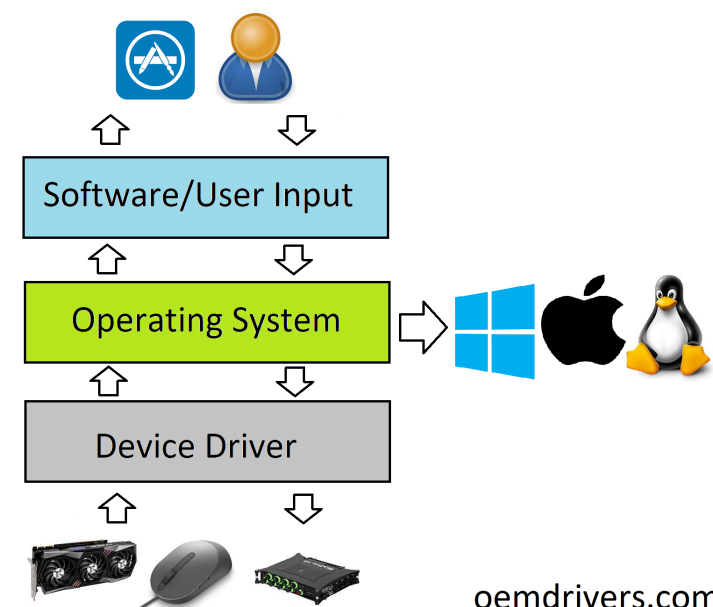
Οδηγοί εκτυπωτών.

Οδηγοί δικτύου.

Οδηγοί δίσκων.



Device drivers



Στρατηγικές Διαχείρισης Εισόδου/Εξόδου

Υπάρχουν διάφορες στρατηγικές που χρησιμοποιούνται για τη διαχείριση των συσκευών I/O. Η επιλογή της σωστής στρατηγικής είναι κρίσιμη για την απόδοση του συστήματος, ιδιαίτερα σε συστήματα που διαχειρίζονται πολλές συσκευές I/O ταυτόχρονα.

Προγραμματισμός Δίσκων (Disk Scheduling)

Ο **προγραμματισμός δίσκων** είναι η διαδικασία με την οποία το λειτουργικό σύστημα αποφασίζει ποια αιτήματα I/O θα εκτελεστούν πρώτα, προκειμένου να ελαχιστοποιηθεί ο χρόνος αναμονής και να βελτιωθεί η αποδοτικότητα.

Αλγόριθμοι Προγραμματισμού Δίσκων:

FCFS (First Come, First Served): Τα αιτήματα εκτελούνται με τη σειρά που λαμβάνονται.

SSTF (Shortest Seek Time First):

Προτεραιότητα στα αιτήματα που απαιτούν τη μικρότερη κίνηση της κεφαλής του δίσκου.

SCAN (Elevator Algorithm): Η κεφαλή του δίσκου κινείται από τη μία άκρη στην άλλη, εξυπηρετώντας τα αιτήματα καθώς περνάει από αυτά.

Αποθήκευση σε Cache

Η **αποθήκευση σε cache** είναι μια τεχνική που χρησιμοποιείται για τη βελτίωση της απόδοσης των συστημάτων I/O. Τα δεδομένα που χρησιμοποιούνται συχνά αποθηκεύονται προσωρινά στη μνήμη cache, επιτρέποντας ταχύτερη πρόσβαση στις μελλοντικές αιτήσεις.

Αντιμετώπιση Σφαλμάτων

Η **αντιμετώπιση σφαλμάτων** είναι κρίσιμη για τη διαχείριση των συσκευών I/O, καθώς οι συσκευές αυτές είναι επιρρεπείς

σε σφάλματα, όπως απώλεια δεδομένων ή καταστροφή αρχείων. Το λειτουργικό σύστημα πρέπει να ενσωματώνει στρατηγικές όπως η επανεκκίνηση συσκευών ή η χρήση εφεδρικών αντιγράφων.

Name	Description	Remarks
Selection according to requestor		
RSS	Random scheduling	For analysis and simulation
FIFO	First in first out	Fairest of them all
PRI	Priority by process	Control outside of disk queue management
LIFO	Last in first out	Maximize locality and resource utilization
Selection according to requested item		
SSTF	Shortest service time first	High utilization, small queues
SCAN	Back and forth over disk	Better service distribution
C-SCAN	One way with fast return	Lower service variability
N-step-SCAN	SCAN of N records at a time	Service guarantee
FSCAN	N-step-SCAN with N = queue size at beginning of SCAN cycle	Load sensitive

Disk scheduling algorithms

Βελτιστοποίηση των Συστημάτων I/O

Βελτιστοποίηση των Συστημάτων I/O

Η **βελτιστοποίηση των συστημάτων I/O** είναι σημαντική για τη διασφάλιση της απόδοσης και της αξιοπιστίας των υπολογιστικών συστημάτων. Οι παρακάτω τεχνικές μπορούν να χρησιμοποιηθούν για τη βελτίωση της απόδοσης των συσκευών I/O:

Συλλογή Αιτημάτων (Request Merging)

Η **συλλογή αιτημάτων** είναι μια τεχνική που χρησιμοποιείται για να συνδυάσει πολλά αιτήματα I/O σε ένα μεγαλύτερο αίτημα, μειώνοντας έτσι την ανάγκη για πολλαπλές προσπελάσεις στις συσκευές.

Χρήση Cache

Όπως αναφέρθηκε νωρίτερα, η **χρήση μνήμης cache** για την αποθήκευση συχνά χρησιμοποιούμενων δεδομένων μπορεί να μειώσει τον χρόνο πρόσβασης και να βελτιώσει την απόδοση.

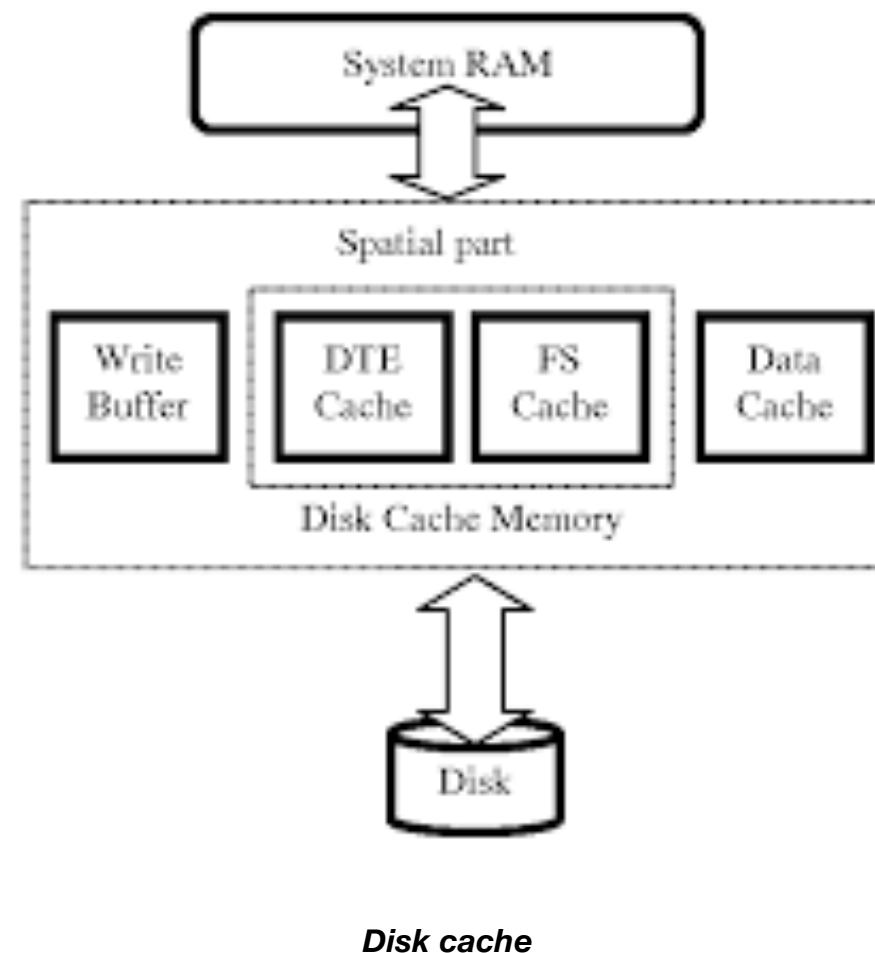
Παραλληλισμός (Parallelism)

Η **παραλληλία** στη διαχείριση I/O μπορεί να βελτιώσει την απόδοση μέσω της εκτέλεσης πολλών εργασιών I/O ταυτόχρονα, χρησιμοποιώντας πολλαπλά κανάλια I/O.

Συμπεράσματα

Η διαχείριση της εισόδου/εξόδου είναι κρίσιμη για την αποδοτική λειτουργία των υπολογιστικών συστημάτων. Η σωστή διαχείριση συσκευών I/O, η χρήση τεχνικών όπως οι διακοπές, το DMA και οι οδηγοί συσκευών, και η εφαρμογή στρατηγικών προγραμματισμού δίσκων και

αποθήκευσης σε cache είναι απαραίτητες για τη βελτίωση της απόδοσης. Τα σύγχρονα λειτουργικά συστήματα υιοθετούν διαφορετικές προσεγγίσεις για τη βελτιστοποίηση των λειτουργιών I/O και την εξασφάλιση της σωστής λειτουργίας των συσκευών.

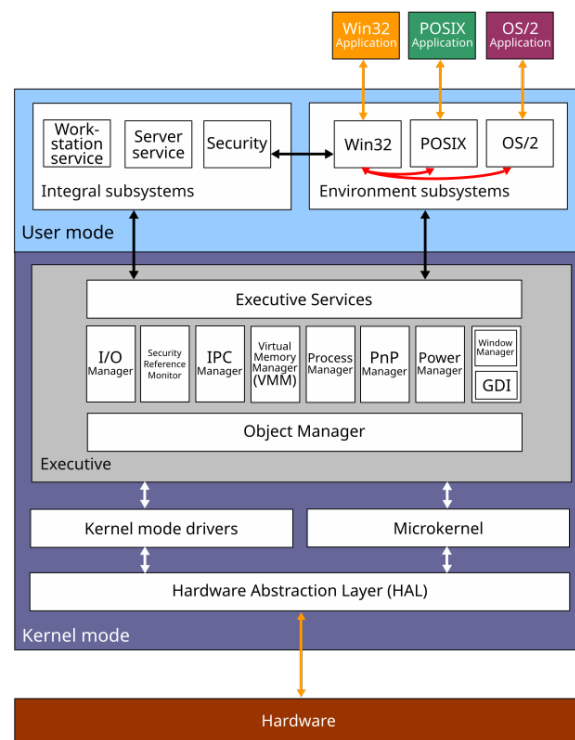


Παραδείγματα Υλοποίησης I/O σε Λειτουργικά Συστήματα

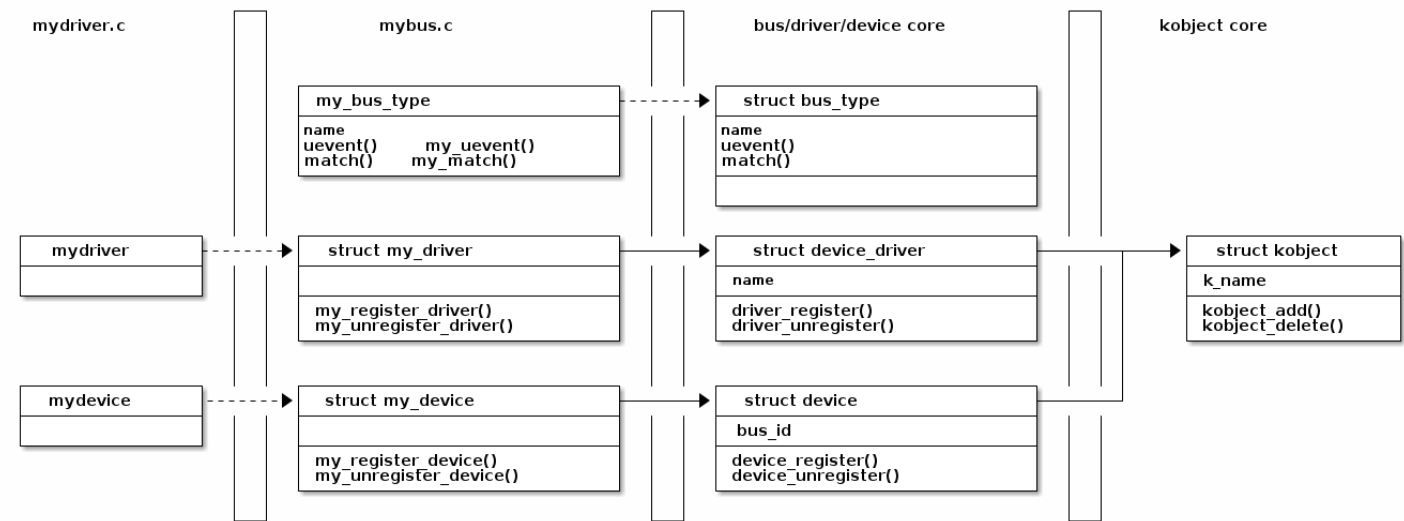
Διάφορα λειτουργικά συστήματα χρησιμοποιούν διαφορετικές προσεγγίσεις για τη διαχείριση του I/O, ανάλογα με τις ανάγκες τους.

Διαχείριση I/O στα Windows

Τα **Windows** χρησιμοποιούν μια ιεραρχία διαχείρισης I/O, όπου οι συσκευές οργανώνονται σε επίπεδα, με τους οδηγούς συσκευών να παίζουν σημαντικό ρόλο στην επικοινωνία με το λειτουργικό σύστημα. Το μοντέλο I/O των Windows περιλαμβάνει υποστήριξη για διακοπές και DMA.



Windows drivers

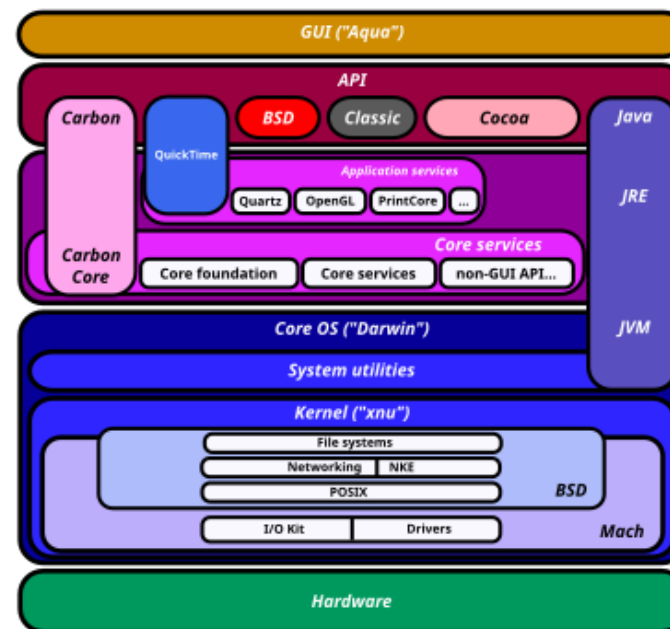


Διαχείριση I/O στο Linux

Το **Linux** χρησιμοποιεί ένα ευέλικτο μοντέλο διαχείρισης I/O, με έμφαση στην υποστήριξη πολυάριθμων συσκευών και τη βελτιστοποίηση της απόδοσης μέσω τεχνικών όπως η χρήση μνήμης cache και ο προγραμματισμός δίσκων. Το Linux περιλαμβάνει επίσης μια σειρά αλγορίθμων προγραμματισμού για τη βελτίωση της απόδοσης.

Διαχείριση I/O στο macOS

Το **macOS** χρησιμοποιεί ένα μοντέλο διαχείρισης I/O που βασίζεται σε τεχνολογίες όπως το **I/O Kit**, το οποίο προσφέρει μια αντικειμενοστραφή πλατφόρμα για την ανάπτυξη οδηγών συσκευών και τη βελτιστοποίηση των διεργασιών I/O.



MacOS drivers

Διαχείριση Πόρων και Προγραμματισμός CPU (Resource Management and CPU)

Στόχοι:

- Κατανόηση της διαχείρισης πόρων και του ρόλου της στο λειτουργικό σύστημα.
- Ανάλυση των αλγορίθμων προγραμματισμού CPU.
- Εξέταση των διαφορετικών στρατηγικών κατανομής πόρων και προγραμματισμού CPU.
- Κατανόηση της βελτιστοποίησης της απόδοσης της CPU και των διεργασιών.
- Παρουσίαση πρακτικών παραδειγμάτων και υλοποιήσεων.

Εισαγωγή στη Διαχείριση Πόρων / Προγραμματισμός CPU (CPU Scheduling)

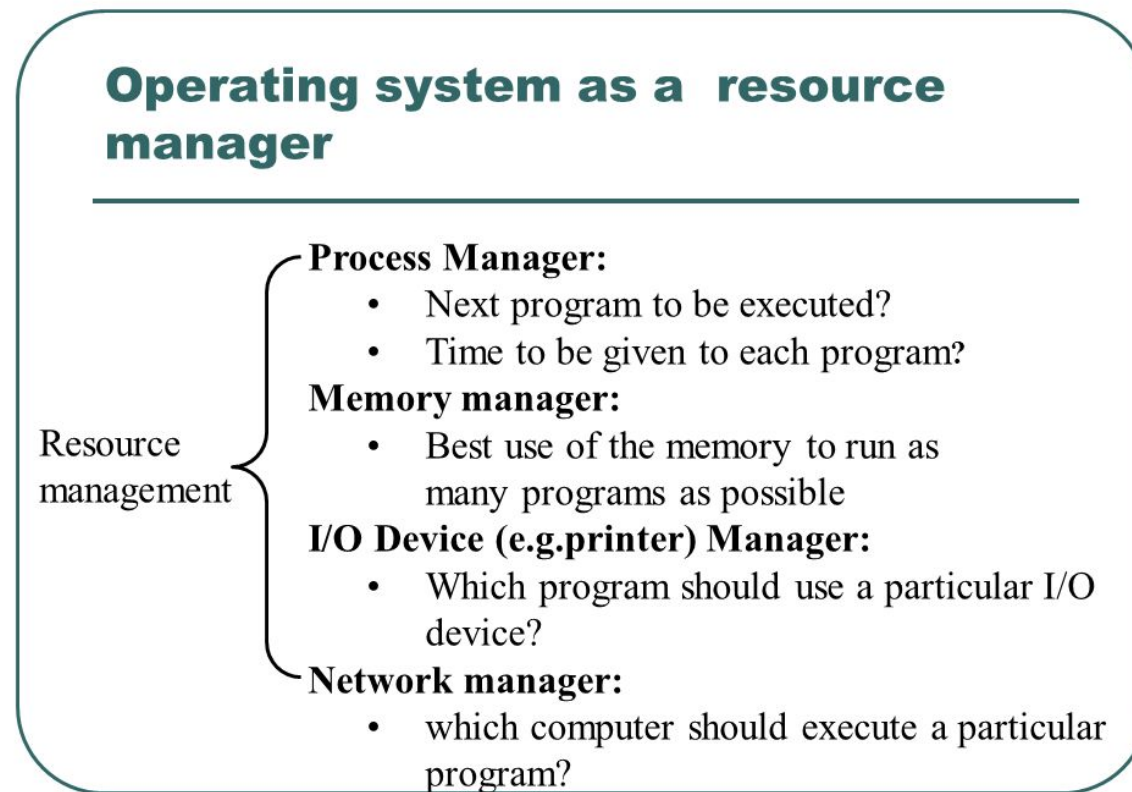
Η **διαχείριση πόρων** είναι μία από τις βασικές λειτουργίες ενός λειτουργικού συστήματος και αναφέρεται στη διαδικασία κατανομής και ελέγχου των πόρων του συστήματος, όπως ο επεξεργαστής (CPU), η μνήμη, οι συσκευές εισόδου/εξόδου (I/O), και ο αποθηκευτικός χώρος. Το λειτουργικό σύστημα ενεργεί ως «μεσολαβητής», διασφαλίζοντας ότι οι πόροι χρησιμοποιούνται αποδοτικά από τις εφαρμογές και τις διεργασίες χωρίς συγκρούσεις ή σπατάλη.

Οι πόροι ενός συστήματος μπορούν να χωριστούν σε δύο κύριες κατηγορίες:

Υπολογιστικοί πόροι (Computing resources): Αυτοί περιλαμβάνουν την CPU, την κύρια μνήμη (RAM), και τα συστήματα αποθήκευσης.

Περιφερειακοί πόροι (Peripheral resources): Αυτοί περιλαμβάνουν συσκευές εισόδου/εξόδου όπως δίσκους, πληκτρολόγια, οθόνες και δικτυακές κάρτες.

Η αποδοτική διαχείριση πόρων απαιτεί ακριβή παρακολούθηση, κατανομή, και συντονισμό, ώστε να βελτιστοποιηθεί η απόδοση και να αποφευχθούν προβλήματα όπως η υπερφόρτωση και οι deadlocks.



resources management

Προγραμματισμός CPU (CPU Scheduling)

Ο προγραμματισμός της CPU είναι μία από τις πιο σημαντικές λειτουργίες του λειτουργικού συστήματος, καθώς καθορίζει ποια διεργασία θα εκτελεστεί από την CPU ανά πάσα στιγμή. Δεδομένου ότι η CPU είναι ένας περιορισμένος πόρος, ο προγραμματισμός της πρέπει να γίνει με τέτοιο τρόπο ώστε να μεγιστοποιείται η

απόδοση και η αποδοτικότητα του συστήματος.

Στόχοι

Προγραμματισμού CPU:

Αποδοτική χρήση της CPU:

Ελαχιστοποίηση του χρόνου που η CPU παραμένει αδρανής.

Ελαχιστοποίηση του χρόνου απόκρισης (response time): Ο χρόνος μεταξύ της υποβολής μιας διεργασίας και της πρώτης απόκρισής της.

Ελαχιστοποίηση του χρόνου αναμονής (waiting time): Ο χρόνος που η διεργασία περιμένει να εκτελεστεί.

Μέγιστη παραγωγικότητα (throughput): Ο αριθμός διεργασιών που ολοκληρώνονται ανά μονάδα χρόνου.

Υπάρχουν διάφοροι αλγόριθμοι προγραμματισμού CPU, καθένας με τα δικά του χαρακτηριστικά και πλεονεκτήματα.

Αλγόριθμοι Προγραμματισμού CPU 1/2

Υπάρχουν αρκετοί **αλγόριθμοι προγραμματισμού CPU**, οι οποίοι χρησιμοποιούνται για να καθορίσουν ποια διεργασία θα εκτελεστεί από την CPU. Παρακάτω παρουσιάζονται οι πιο κοινοί αλγόριθμοι:

First-Come, First-Served (FCFS)

Ο **FCFS** είναι ο απλούστερος αλγόριθμος προγραμματισμού. Οι διεργασίες εκτελούνται με τη σειρά που φτάνουν στην ουρά αναμονής. Αυτή η μέθοδος είναι εύκολη στην υλοποίηση, αλλά μπορεί να προκαλέσει προβλήματα όπως η μεγάλη αναμονή για διεργασίες με μικρή διάρκεια (πρόβλημα «convo effect»).

Πλεονεκτήματα:

Εύκολη υλοποίηση.
Προβλέψιμη συμπεριφορά.

Μειονεκτήματα:

Προκαλεί μεγάλα διαστήματα αναμονής για σύντομες διεργασίες.
Δεν βελτιστοποιεί τον χρόνο απόκρισης.

Shortest Job First (SJF)

Ο **αλγόριθμος SJF** εκτελεί πρώτα τη διεργασία με τη μικρότερη προβλεπόμενη διάρκεια εκτέλεσης. Αυτός ο αλγόριθμος

FCFS (Example)

Process	Duration	Oder	Arrival Time
P1	24	1	0
P2	3	2	0
P3	4	3	0

Gantt Chart :



P1 waiting time : 0
P2 waiting time : 24
P3 waiting time : 27

The Average waiting time :
 $(0+24+27)/3 = 17$

FCFS

ελαχιστοποιεί τον μέσο χρόνο αναμονής, αλλά μπορεί να προκαλέσει το πρόβλημα της πείνας (starvation) για μεγάλες διεργασίες.

Πλεονεκτήματα:

Ελαχιστοποιεί τον μέσο χρόνο αναμονής.

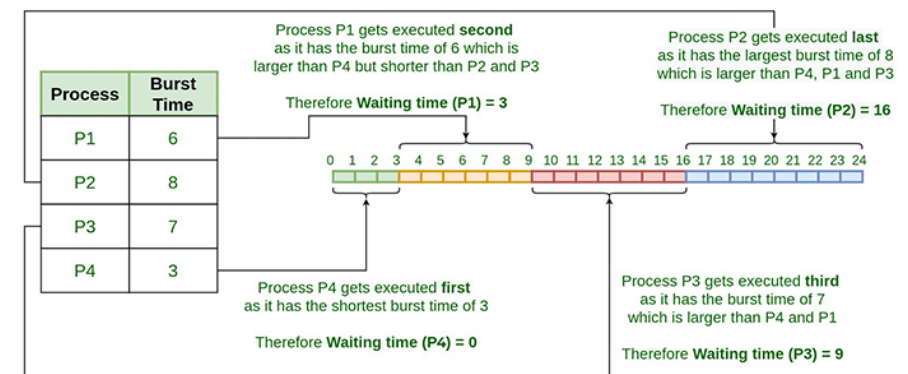
Είναι βελτιστοποιημένος για μικρές διεργασίες.

Μειονεκτήματα:

Είναι δύσκολο να προβλεφθεί με ακρίβεια η διάρκεια της εκτέλεσης.

Μεγάλες διεργασίες μπορεί να καθυστερήσουν σημαντικά.

Shortest Job First (SJF) Scheduling Algorithm



SJF

Αλγόριθμοι Προγραμματισμού CPU 2/2

Round Robin (RR)

Ο **Round Robin** είναι ένας προληπτικός αλγόριθμος προγραμματισμού, όπου κάθε διεργασία εκτελείται για ένα καθορισμένο χρονικό διάστημα, που ονομάζεται χρονική φέτα (time slice), πριν επιστρέψει στην ουρά αναμονής. Ο RR διασφαλίζει ότι όλες οι διεργασίες λαμβάνουν ένα ίσο ποσοστό της CPU.

Πλεονεκτήματα:

Εξασφαλίζει ισότιμη κατανομή της CPU.

Είναι αποδοτικός για συστήματα με πολλές διεργασίες.

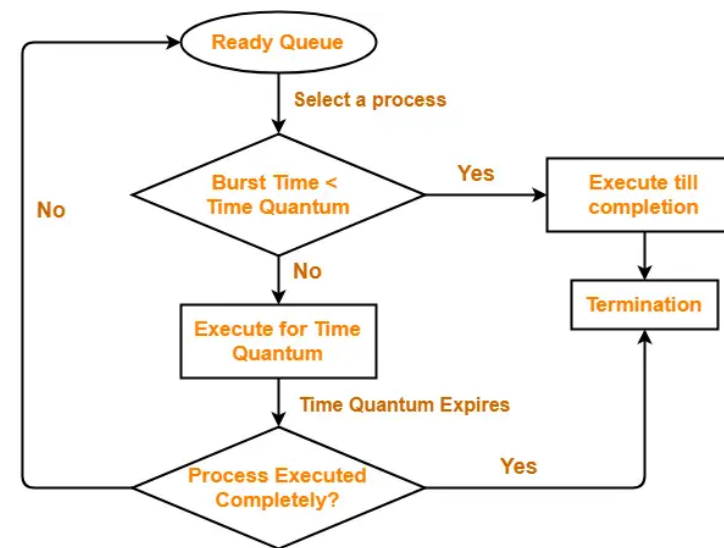
Μειονεκτήματα:

Μπορεί να δημιουργήσει overhead λόγω συχνών αλλαγών κοντεξτ (context switching).

είτε μη προληπτικός.

Πλεονεκτήματα:

Προτεραιότητα στις πιο σημαντικές διεργασίες.



Round Robin Scheduling

RR

Η απόδοση εξαρτάται από το μέγεθος της χρονικής φέτας.

Priority Scheduling

Στον αλγόριθμο προγραμματισμού προτεραιότητας (Priority Scheduling), κάθε διεργασία έχει μια προτεραιότητα και η CPU εκτελεί πρώτα τις διεργασίες με την υψηλότερη προτεραιότητα. Μπορεί να είναι είτε προληπτικός

Process	Arrival Time	Burst Time	Priority
P1	0	11	2
P2	5	28	0
P3	12	2	3
P4	2	10	1
P5	9	16	4

P1	P2	P3	P4	P5	P6	P1	P3	P4	P5	P6	P3	P4	P5
0	5	9	14	19	24	29	31	36	41	46	50	55	56

Process ID	Arrival Time	Burst Time	Completion Time	Turn Around Time	Waiting Time
P1	0	7	31	31	24
P2	1	4	9	8	4
P3	2	15	55	53	38
P4	3	11	56	53	42
P5	4	20	66	62	42
P6	4	9	50	46	37

RR

Priority scheduling

Καλή διαχείριση για συστήματα πραγματικού χρόνου.

Μειονεκτήματα:

Μπορεί να προκαλέσει πρόβλημα πείνας (starvation) για διεργασίες με χαμηλή προτεραιότητα.

Απαιτεί σωστή ρύθμιση των προτεραιοτήτων.

Βελτιστοποίηση της Απόδοσης CPU / Διαχείριση Πόρων

Βελτιστοποίηση της Απόδοσης CPU

Η βελτιστοποίηση της απόδοσης της CPU είναι σημαντική για τη μεγιστοποίηση της παραγωγικότητας και της αποδοτικότητας του συστήματος. Οι κύριες τεχνικές βελτιστοποίησης περιλαμβάνουν:

Χρονισμός Προγραμματισμού

Ο σωστός χρονισμός του προγραμματισμού διεργασιών μπορεί να μειώσει τον χρόνο αναμονής και να αυξήσει την απόδοση της CPU. Η επιλογή του κατάλληλου αλγορίθμου προγραμματισμού παίζει βασικό ρόλο σε αυτό.

Αποφυγή Overhead

Η συχνή εναλλαγή κοντεξτ (context switching) μπορεί να προκαλέσει σημαντικό overhead, επιβραδύνοντας τη συνολική απόδοση του συστήματος. Η χρήση αλγορίθμων όπως ο Round Robin πρέπει να γίνει προσεκτικά, για να αποφεύγεται η υπερβολική εναλλαγή.

Η **διαχείριση πόρων είναι ένα βασικό κομμάτι της συνολικής διαχείρισης του λειτουργικού συστήματος**. Πέρα από την CPU, το λειτουργικό σύστημα πρέπει να διαχειρίζεται και άλλους πόρους, όπως η μνήμη, οι συσκευές I/O και η αποθήκευση. Η αποτελεσματική διαχείριση πόρων διασφαλίζει ότι όλοι οι πόροι

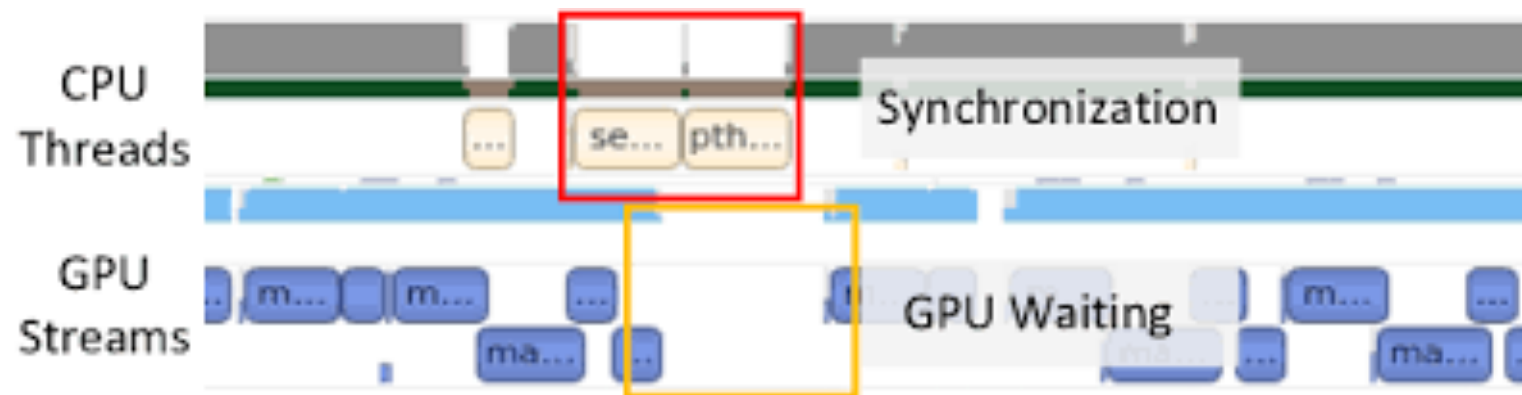
χρησιμοποιούνται με τον πιο αποδοτικό τρόπο.

Διαχείριση Μνήμης

Η διαχείριση της μνήμης περιλαμβάνει τη σωστή κατανομή και απελευθέρωση της μνήμης που απαιτείται από τις διεργασίες. Το λειτουργικό σύστημα χρησιμοποιεί τεχνικές όπως η σελιδοποίηση και η κατάτμηση για τη διαχείριση της μνήμης.

Διαχείριση Συσκευών I/O

Η διαχείριση των συσκευών εισόδου/εξόδου είναι εξίσου κρίσιμη για τη διαχείριση πόρων. Το λειτουργικό σύστημα πρέπει να διαχειρίζεται αιτήματα I/O από πολλές διεργασίες ταυτόχρονα, εξασφαλίζοντας αποδοτική και ασφαλή πρόσβαση στις συσκευές.



CPU Threads / GPU Streams

Παραδείγματα Διαχείρισης και Προγραμματισμού CPU σε Λειτουργικά

Διάφορα λειτουργικά συστήματα εφαρμόζουν διαφορετικές προσεγγίσεις για τη διαχείριση των πόρων και τον προγραμματισμό της CPU.

Προγραμματισμός CPU στα **Windows**

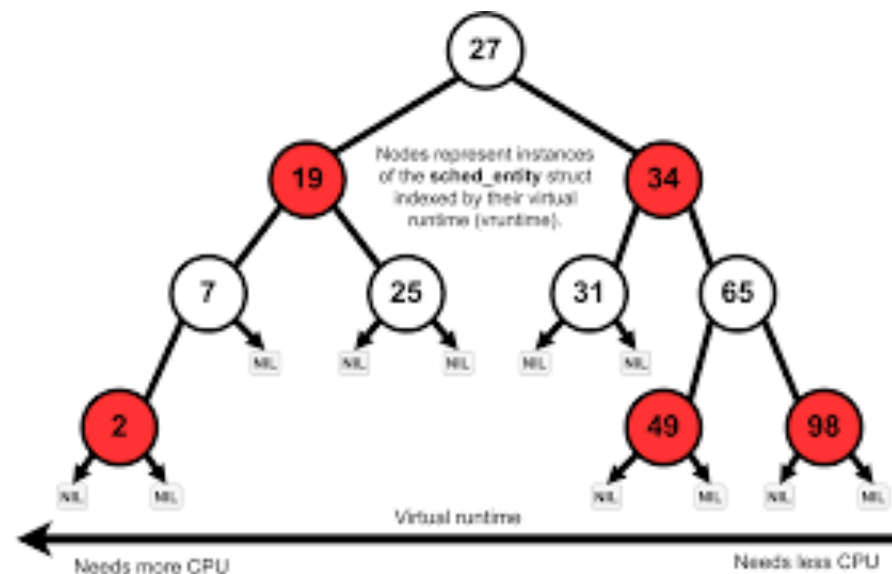
Τα Windows χρησιμοποιούν έναν συνδυασμό αλγορίθμων προγραμματισμού προτεραιότητας, Round Robin, και πολυεπεξεργασίας για να διαχειρίζονται τις διεργασίες τους. Ένα από τα κύρια χαρακτηριστικά είναι η δυνατότητα διαχείρισης πολλών νημάτων ταυτόχρονα.

Προγραμματισμός CPU στο **Linux**

Το Linux χρησιμοποιεί τον αλγόριθμο **Completely Fair Scheduler (CFS)**, που είναι σχεδιασμένος για να παρέχει ισότιμη κατανομή της CPU στις διεργασίες, λαμβάνοντας υπόψη τόσο την προτεραιότητα όσο και τον χρόνο εκτέλεσης.

Προγραμματισμός CPU στο **macOS**

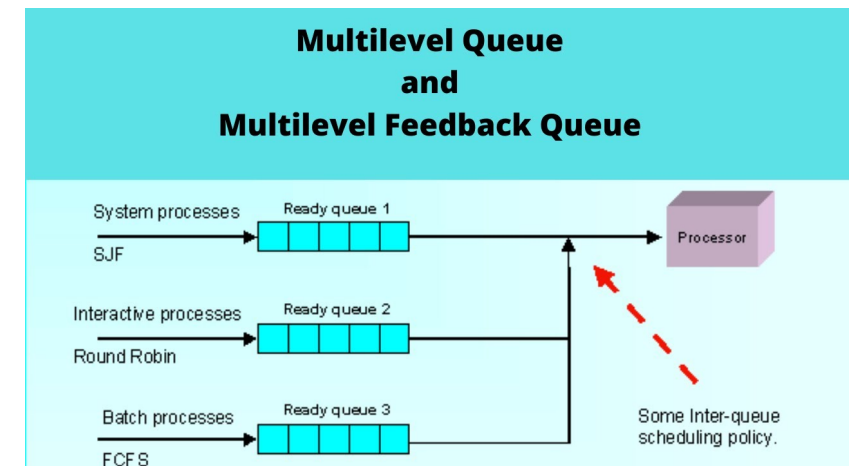
Το macOS χρησιμοποιεί ένα μοντέλο προγραμματισμού που βασίζεται σε προτεραιότητες, με έμφαση στην αποδοτική διαχείριση των διεργασιών και την προσαρμογή της απόδοσης για συσκευές με περιορισμένους πόρους, όπως τα MacBooks.



CFS

Συμπεράσματα

Η διαχείριση πόρων και ο προγραμματισμός της CPU είναι κρίσιμες λειτουργίες ενός λειτουργικού συστήματος. Μέσω της σωστής κατανομής των πόρων, της επιλογής κατάλληλων αλγορίθμων προγραμματισμού και της βελτιστοποίησης της απόδοσης της CPU, τα λειτουργικά συστήματα εξασφαλίζουν τη μέγιστη αποδοτικότητα και αξιοπιστία. Η κατανόηση των τεχνικών αυτών επιτρέπει στους προγραμματιστές και διαχειριστές συστημάτων να μεγιστοποιήσουν την απόδοση των



multi level scheduling

Ασφάλεια και Προστασία (Security and Protection)

Στόχοι:

- Κατανόηση των βασικών εννοιών της ασφάλειας και της προστασίας σε λειτουργικά συστήματα.
- Ανάλυση των επιπέδων προστασίας και των μηχανισμών ασφαλείας.
- Κατανόηση των τεχνικών ελέγχου πρόσβασης και των μοντέλων ασφαλείας.
- Εξέταση των μηχανισμών προστασίας από εξωτερικές και εσωτερικές απειλές.
- Παρουσίαση στρατηγικών πρόληψης και αντιμετώπισης επιθέσεων.

Εισαγωγή στην Ασφάλεια και την Προστασία. /Επίπεδα προστασίας

Η **ασφάλεια** στα λειτουργικά συστήματα αναφέρεται στην προστασία των δεδομένων και των πόρων του συστήματος από μη εξουσιοδοτημένη πρόσβαση και καταστροφή. Η ασφάλεια και η προστασία είναι δύο διαφορετικές, αλλά αλληλένδετες έννοιες:

Ασφάλεια (Security): Περιλαμβάνει την προστασία από εξωτερικές και εσωτερικές απειλές, όπως οι κυβερνοεπιθέσεις.

Προστασία (Protection): Αφορά την εσωτερική ασφάλεια ενός συστήματος, διασφαλίζοντας ότι οι διεργασίες και οι χρήστες δεν επηρεάζουν αρνητικά ο ένας τον άλλον.

Τα λειτουργικά συστήματα παρέχουν διάφορους μηχανισμούς προστασίας, όπως η διαχείριση των δικαιωμάτων πρόσβασης, τα εργαλεία ασφαλείας, και τα επίπεδα προστασίας.

Σημαντικοί στόχοι ασφάλειας:

Εμπιστευτικότητα (Confidentiality):

Εξασφαλίζει ότι μόνο εξουσιοδοτημένοι χρήστες έχουν πρόσβαση σε δεδομένα.

Ακεραιότητα (Integrity):

Διασφαλίζει ότι τα δεδομένα δεν μπορούν να τροποποιηθούν από μη εξουσιοδοτημένα άτομα.

Διαθεσιμότητα (Availability):

Εξασφαλίζει ότι οι πόροι και τα δεδομένα είναι διαθέσιμα στους εξουσιοδοτημένους χρήστες όταν απαιτείται.

Επίπεδα Προστασίας

Η προστασία στα λειτουργικά συστήματα

παρέχεται σε διάφορα επίπεδα. Κάθε επίπεδο προσφέρει διαφορετικό βαθμό ασφάλειας και απομόνωσης των πόρων, επιτρέποντας στον διαχειριστή του συστήματος να προσαρμόσει το σύστημα στις ανάγκες του.

Χρήστες και Δικαιώματα Πρόσβασης

Οι χρήστες σε ένα λειτουργικό σύστημα κατηγοριοποιούνται συνήθως σε:

Κανονικούς χρήστες: Έχουν περιορισμένη πρόσβαση στους πόρους του συστήματος.

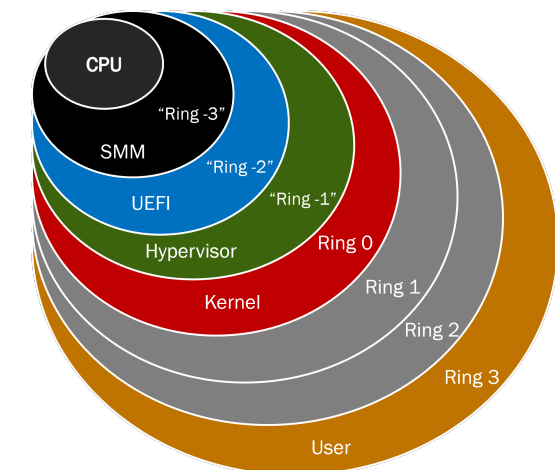
Διαχειριστές (Administrators): Έχουν πλήρη έλεγχο στους πόρους του συστήματος.

Κάθε χρήστης ή διεργασία έχει συγκεκριμένα **δικαιώματα πρόσβασης** που καθορίζουν τι μπορεί και δεν μπορεί να κάνει:

Ανάγνωση (Read): Δικαίωμα προβολής δεδομένων.

Γράψιμο (Write): Δικαίωμα τροποποίησης ή δημιουργίας νέων δεδομένων.

Εκτέλεση (Execute): Δικαίωμα εκτέλεσης προγραμμάτων.



Αρχιτεκτονική

Δακτυλίων (Ring Architecture)

Η αρχιτεκτονική δακτυλίων (ring architecture) είναι μια μέθοδος διαχείρισης της ασφάλειας σε επίπεδα, με το Ring 0 να έχει πλήρη πρόσβαση στο υλικό (kernel mode) και τα υπόλοιπα δακτυλίδια να έχουν περιορισμένη πρόσβαση.

Principles of Protection

- Principles of Least Privilege,
- Fine-grained Access Controls,
- Role-based Access Control (RBAC).



Αρχές προστασίας

Έλεγχος Πρόσβασης

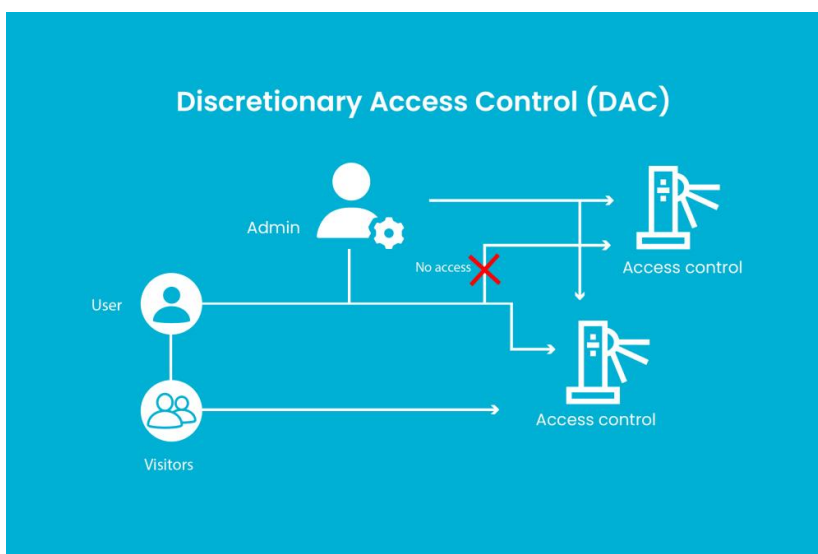
Ο **έλεγχος πρόσβασης (access control)** είναι μια σημαντική πτυχή της ασφάλειας, καθώς διασφαλίζει ότι μόνο οι εξουσιοδοτημένοι χρήστες και διεργασίες έχουν πρόσβαση στους πόρους του συστήματος.

Μοντέλα Ελέγχου Πρόσβασης

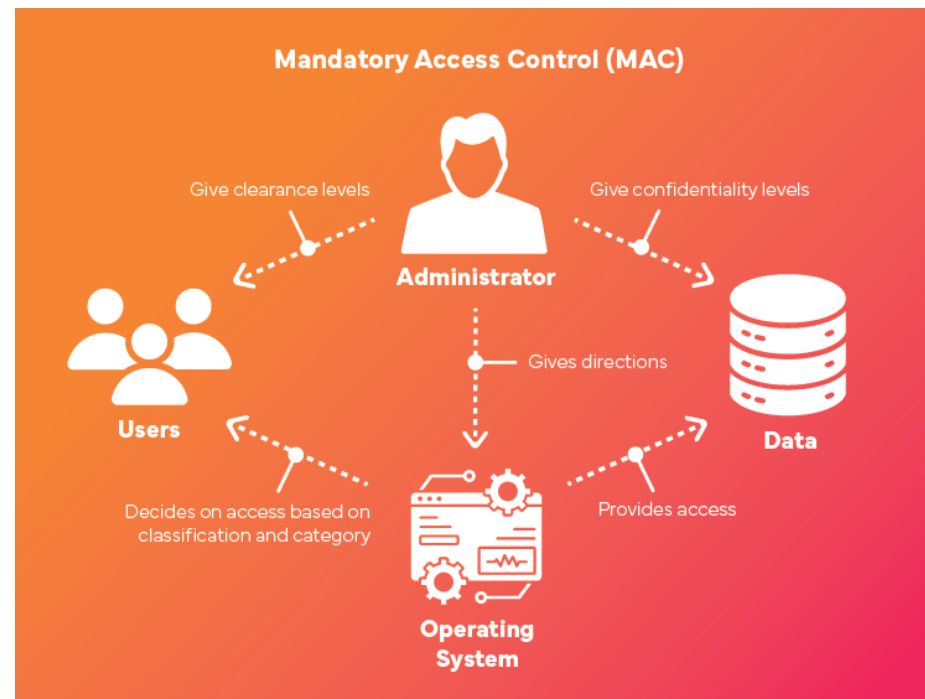
Υπάρχουν διάφορα μοντέλα ελέγχου πρόσβασης που χρησιμοποιούνται σε λειτουργικά συστήματα:

Discretionary Access Control (DAC)

Στο **DAC**, κάθε ιδιοκτήτης αρχείου ή πόρου μπορεί να καθορίσει ποιοι χρήστες έχουν πρόσβαση και σε τι βαθμό. Αυτή η μέθοδος είναι ευέλικτη, αλλά μπορεί να



DAC



MAC

προκαλέσει κενά ασφαλείας λόγω ανθρώπινου λάθους.

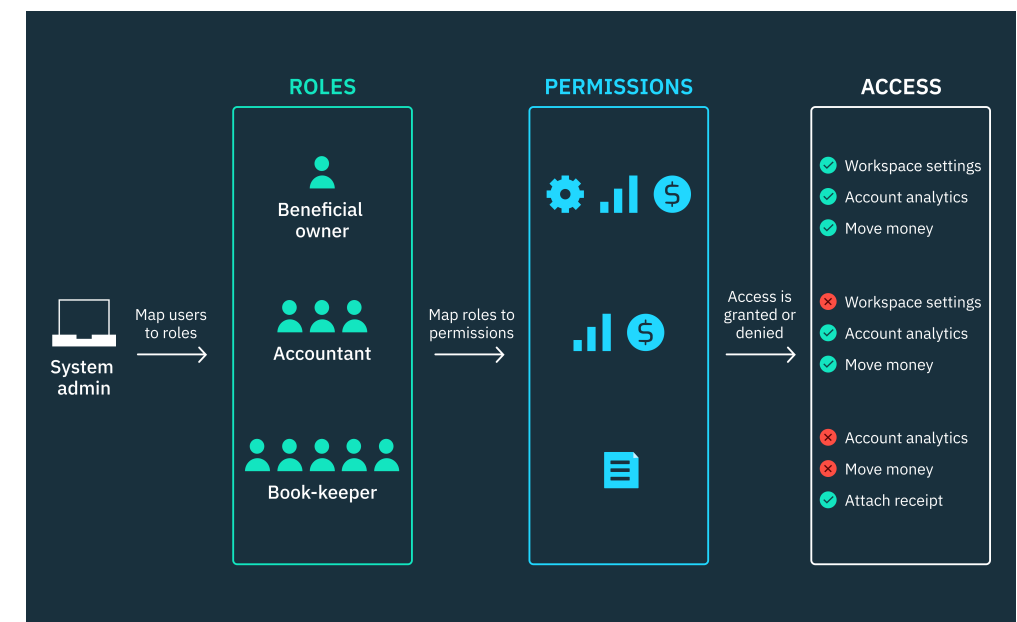
Mandatory Access Control (MAC)

Το **MAC** είναι πιο αυστηρό από το DAC, καθώς τα δικαιώματα πρόσβασης καθορίζονται από τον διαχειριστή του συστήματος και δεν μπορούν να αλλάξουν από τους χρήστες. Αυτό

το μοντέλο χρησιμοποιείται σε συστήματα με αυξημένες απαιτήσεις ασφαλείας, όπως τα στρατιωτικά συστήματα.

Role-Based Access Control (RBAC)

Το **RBAC** βασίζεται στους ρόλους των χρηστών. Κάθε χρήστης έχει έναν συγκεκριμένο ρόλο και οι ρόλοι καθορίζουν ποιοι πόροι είναι προσβάσιμοι. Αυτή η προσέγγιση χρησιμοποιείται ευρέως σε εταιρικά περιβάλλοντα, όπου οι χρήστες έχουν προκαθορισμένους ρόλους.



RBAC

Μηχανισμοί Προστασίας

Τα λειτουργικά συστήματα χρησιμοποιούν διάφορους μηχανισμούς για την προστασία των πόρων και των δεδομένων από κακόβουλες επιθέσεις ή ατυχήματα.

Χρησιμοποίηση Πιστοποιητικών και Κρυπτογράφησης

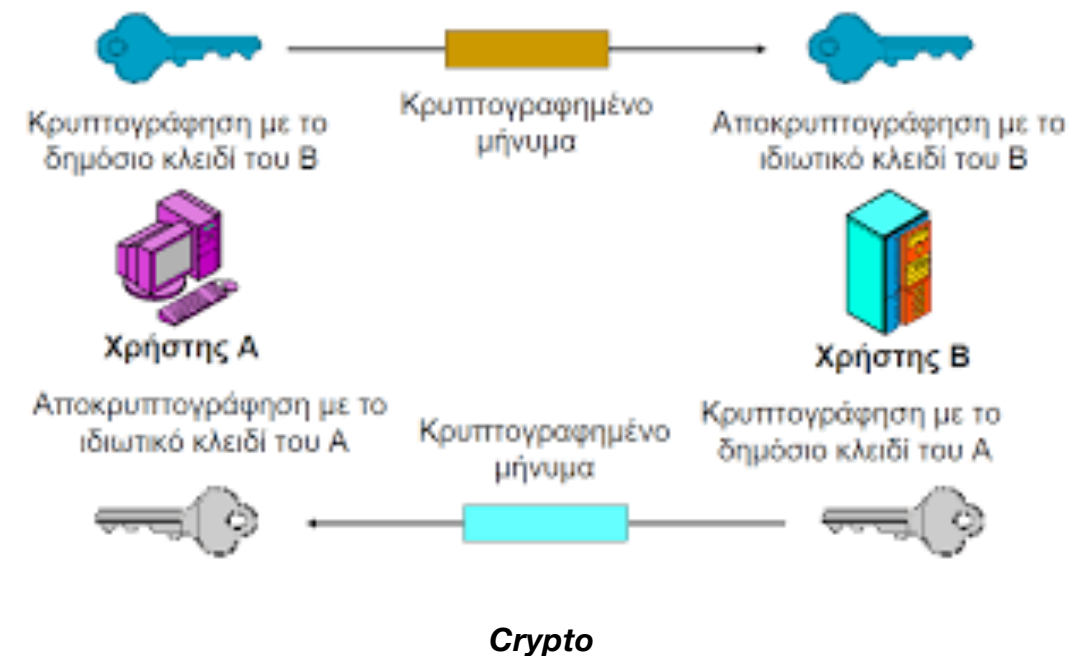
Η **κρυπτογράφηση** είναι ένας από τους πιο σημαντικούς μηχανισμούς προστασίας, καθώς εξασφαλίζει ότι τα δεδομένα δεν μπορούν να αναγνωστούν χωρίς το κατάλληλο κλειδί αποκρυπτογράφησης. Τα λειτουργικά συστήματα ενσωματώνουν κρυπτογραφικούς αλγόριθμους για την προστασία των δεδομένων, τόσο σε κατάσταση ανάπαυσης όσο και κατά τη μετάδοση.

Επιπλέον, τα πιστοποιητικά παρέχουν έναν μηχανισμό για την ταυτοποίηση χρηστών και συσκευών, διασφαλίζοντας ότι μόνο εξουσιοδοτημένα μέρη έχουν πρόσβαση σε ευαίσθητα δεδομένα.

Firewalls και Λογισμικό Ανίχνευσης Απειλών

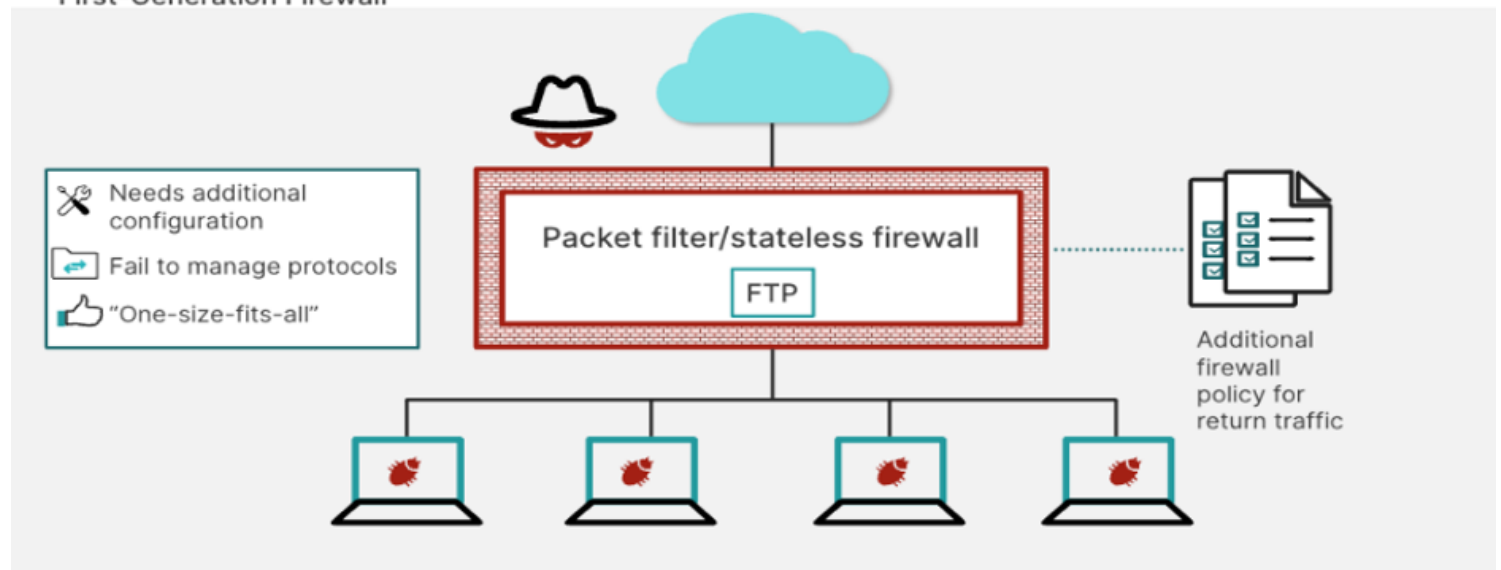
Τα **firewalls** είναι λογισμικά ή υλισμικά συστήματα που ελέγχουν την κυκλοφορία δικτύου μεταξύ του εσωτερικού δικτύου και του εξωτερικού κόσμου. Τα firewalls μπορούν να εμποδίσουν την πρόσβαση από μη εξουσιοδοτημένους χρήστες και να σταματήσουν κακόβουλες ενέργειες.

Επίσης, το **λογισμικό ανίχνευσης απειλών** χρησιμοποιείται για την παρακολούθηση της δραστηριότητας στο σύστημα, εντοπίζοντας πιθανά κακόβουλα προγράμματα και επιθέσεις, όπως ιούς, worms, και ransomware.



How Firewalls Work

First-Generation Firewall



Firewall

Απειλές και Επιθέσεις

Οι απειλές και οι επιθέσεις στα λειτουργικά συστήματα μπορούν να προέλθουν είτε από εξωτερικούς παράγοντες είτε από εσωτερικές αδυναμίες. Οι πιο κοινές μορφές επιθέσεων περιλαμβάνουν:

Κακόβουλο Λογισμικό (Malware)

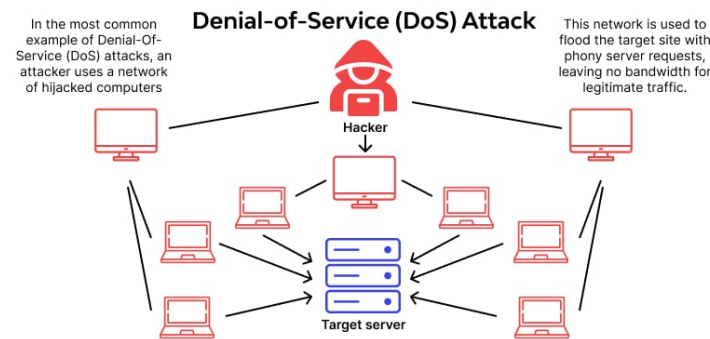
Το **κακόβουλο λογισμικό (malware)** περιλαμβάνει ιούς, worms, και trojans, που έχουν σχεδιαστεί για να καταστρέψουν ή να αποκτήσουν πρόσβαση σε ευαίσθητα δεδομένα. Οι επιθέσεις malware μπορούν να προκαλέσουν σοβαρά προβλήματα, από την απώλεια δεδομένων έως την πλήρη καταστροφή του συστήματος.

Επιθέσεις Αρνησης Υπηρεσιών ()

Οι επιθέσεις **DoS** στοχεύουν στην εξάντληση των πόρων του συστήματος, καθιστώντας το ανίκανο να εξυπηρετήσει νόμιμες αιτήσεις. Αυτό μπορεί να γίνει μέσω της υπερφόρτωσης της CPU, της μνήμης ή της γραμμής δικτύου.

Εκμετάλλευση Ευπαθειών (Exploitation of Vulnerabilities)

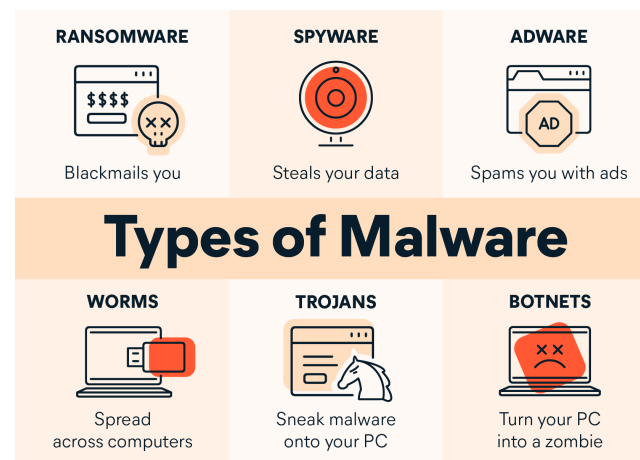
Οι **ευπάθειες** σε ένα λειτουργικό σύστημα μπορούν να επιτρέψουν σε έναν επιτιθέμενο να αποκτ



ήσει μη εξουσιοδοτημένη πρόσβαση σε ευαίσθητα δεδομένα. Οι επιθέσεις αυτού του τύπου εκμεταλλεύονται αδυναμίες στον κώδικα ή στη διαμόρφωση του συστήματος.

Στρατηγικές Πρόληψης και Αντιμετώπισης Επιθέσεων

Για την αποτροπή επιθέσεων και την ελαχιστοποίηση των κινδύνων, τα λειτουργικά συστήματα πρέπει να



Malware

υιοθετούν στρατηγικές πρόληψης και ανίχνευσης.

Ενημερώσεις Ασφαλείας (Security Patches)

Οι **ενημερώσεις ασφαλείας** είναι κρίσιμες για την προστασία των συστημάτων από γνωστές ευπάθειες. Οι διαχειριστές συστημάτων πρέπει να παρακολουθούν συνεχώς για νέες ενημερώσεις και να εφαρμόζουν τις διορθώσεις εγκαίρως.

Δικαιώματα Χρηστών και Πολιτικές Πρόσβασης

Η καθιέρωση **πολιτικών πρόσβασης** που περιορίζουν τα δικαιώματα των χρηστών μπορεί να βοηθήσει στην πρόληψη της μη εξουσιοδοτημένης πρόσβασης. Ο κανόνας του ελάχιστου προνομίου (least privilege) είναι ένα παράδειγμα, όπου οι χρήστες λαμβάνουν μόνο τα δικαιώματα που είναι απαραίτητα για τη δουλειά τους.

Αντιμετώπιση των Απειλών (Threat Mitigation)

Η ενεργή παρακολούθηση και η χρήση **συστήματος ανίχνευσης εισβολών (Intrusion Detection Systems - IDS)** μπορεί να βοηθήσει στον εντοπισμό ύποπτων δραστηριοτήτων και στην άμεση αντιμετώπιση των απειλών.

Εισαγωγή στις Τεχνολογίες Νέφους (Cloud Technologies)

Στόχοι:

- Κατανόηση των βασικών αρχών του cloud computing.
- Ανάλυση των τύπων υποδομών νέφους (IaaS, PaaS, SaaS).
- Εισαγωγή στις τεχνολογίες εικονικοποίησης και containers.
- Εξέταση της ασφάλειας και των προκλήσεων στο cloud.
- Ανάλυση των σύγχρονων τάσεων και τεχνολογιών στον χώρο του cloud.

Εισαγωγή στο Cloud Computing / Τύποι Υποδομών Νέφους (IaaS, PaaS, SaaS)

Το cloud computing έχει μεταμορφώσει τον τρόπο με τον οποίο οι επιχειρήσεις και οι οργανισμοί αποθηκεύουν και επεξεργάζονται δεδομένα, προσφέροντας ευελιξία, κλιμακωσιμότητα και οικονομία. Η βασική ιδέα πίσω από το cloud είναι η δυνατότητα πρόσβασης σε υπολογιστικούς πόρους μέσω του Διαδικτύου, χωρίς να χρειάζεται ο χρήστης να διαχειρίζεται το φυσικό υλικό (hardware).

Ιστορική Εξέλιξη του Cloud Computing:

1960s: Η έννοια του time-sharing σε κεντρικούς υπολογιστές (mainframes) αποτέλεσε τον πρόδρομο του cloud.

1990s: Η ανάπτυξη του Internet και των υποδομών δικτύου οδήγησε στην ανάπτυξη των πρώτων υπηρεσιών που θυμίζουν το σημερινό cloud.

2000s και μετά: Η εμπορική ανάπτυξη από εταιρείες όπως η Amazon, με το AWS (Amazon Web Services), έκανε το cloud ευρέως διαθέσιμο, επιτρέποντας στις επιχειρήσεις να αξιοποιούν δυναμικά υπολογιστικούς πόρους.

Οι υπηρεσίες του cloud μπορούν να κατηγοριοποιηθούν σε διάφορα επίπεδα, με τους πιο **σημαντικούς τύπους να είναι το Infrastructure as a Service (IaaS), το Platform as a Service (PaaS), και το Software as a Service (SaaS).**

Infrastructure as a Service (IaaS)

Το IaaS παρέχει βασική υποδομή, όπως εικονικούς διακομιστές, αποθηκευτικό χώρο και δικτυακούς πόρους, στους οποίους οι χρήστες μπορούν να έχουν πρόσβαση μέσω του Διαδικτύου.

Πλεονεκτήματα: Ελαχιστοποίηση του κόστους υλικού, ευελιξία στην κλιμάκωση.

Παραδείγματα: Amazon EC2, Microsoft Azure, Google Compute Engine.

Platform as a Service (PaaS)

Το PaaS προσφέρει ένα επίπεδο ανάπτυξης εφαρμογών πάνω από το IaaS, παρέχοντας ένα περιβάλλον για την ανάπτυξη, τη δοκιμή και την παράδοση εφαρμογών.

Πλεονεκτήματα: Απλοποιεί τη διαδικασία ανάπτυξης, μειώνει την ανάγκη διαχείρισης υποδομών.

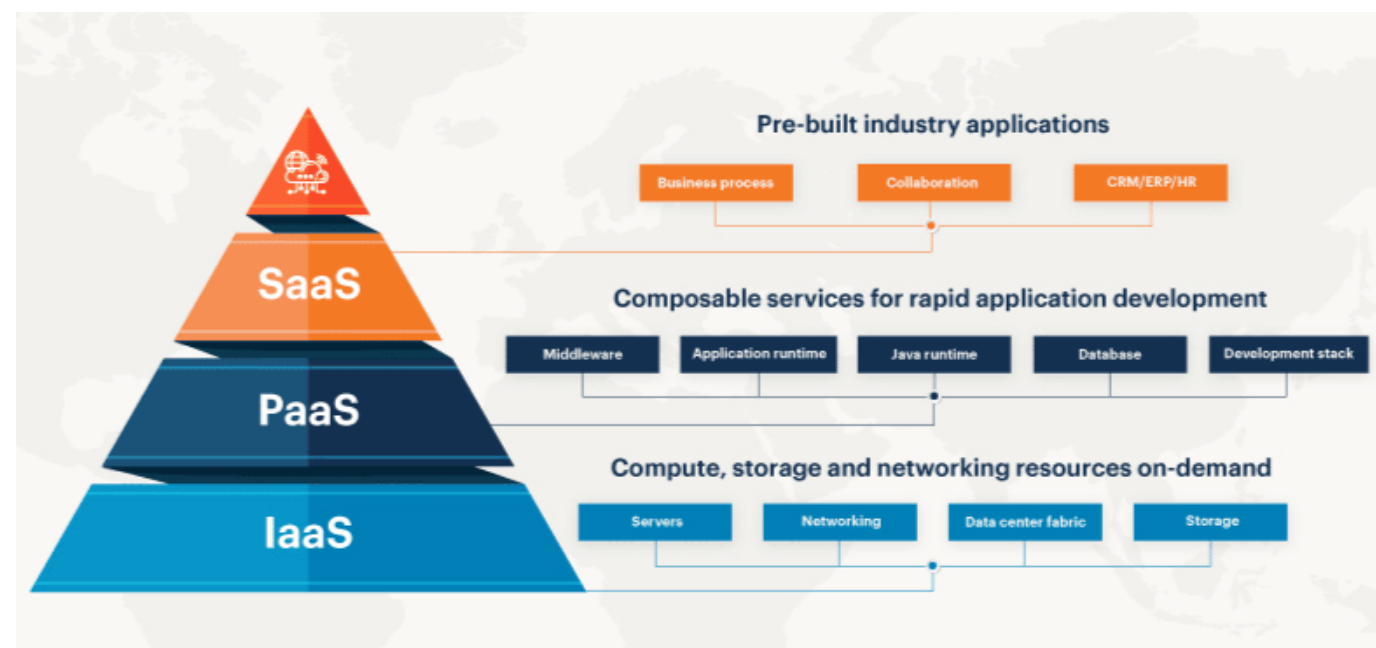
Παραδείγματα: Google App Engine, Heroku, Microsoft Azure App Services.

Software as a Service (SaaS)

Το SaaS προσφέρει εφαρμογές έτοιμες προς χρήση μέσω του Διαδικτύου. Οι χρήστες δεν χρειάζεται να εγκαταστήσουν ή να διαχειριστούν το λογισμικό, καθώς όλα γίνονται μέσω του cloud.

Πλεονεκτήματα: Ευκολία χρήσης, προσβασιμότητα από οπουδήποτε.

Παραδείγματα: Google Workspace, Microsoft 365, Salesforce.



Τύποι Cloud

Τεχνολογίες Εικονικοποίησης και Containers

Η **εικονικοποίηση** είναι η βάση του cloud computing, επιτρέποντας τη δημιουργία πολλαπλών εικονικών μηχανών (**VMs**) που λειτουργούν σε έναν φυσικό διακομιστή. Αυτό βελτιώνει τη χρήση των πόρων και επιτρέπει την απομόνωση των εφαρμογών.

Hypervisors (Ανάλυση στο επόμενο κεφάλαιο)

Οι hypervisors είναι λογισμικά που επιτρέπουν τη δημιουργία και τη διαχείριση των VMs.

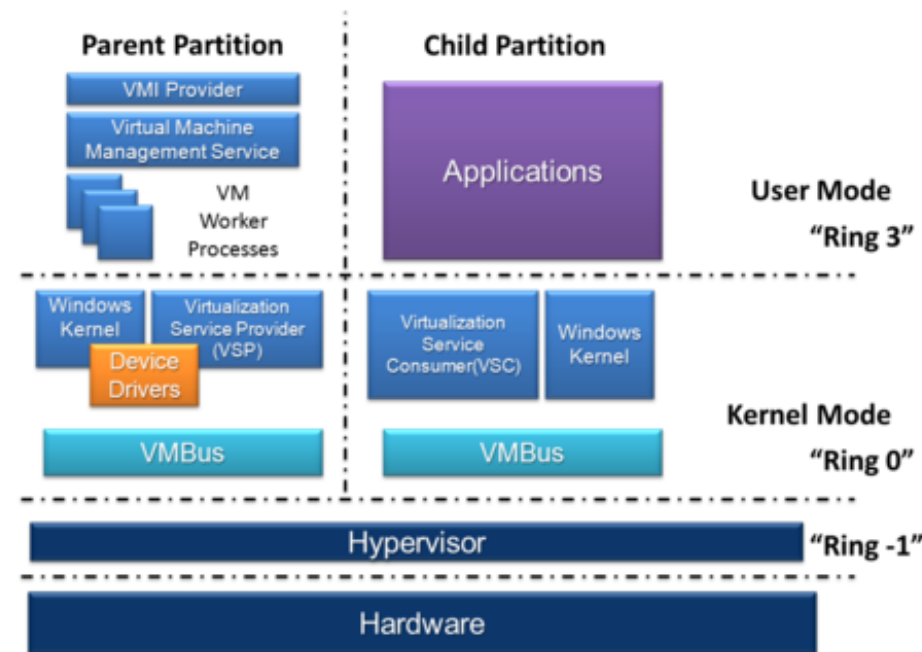
Type 1 Hypervisors: Λειτουργούν απευθείας στο υλικό του διακομιστή (**bare-metal**). Παραδείγματα: VMware ESXi, Microsoft Hyper-V.

Type 2 Hypervisors: Λειτουργούν πάνω από ένα λειτουργικό σύστημα. Παραδείγματα: Oracle VirtualBox, VMware Workstation.

Containers και Docker

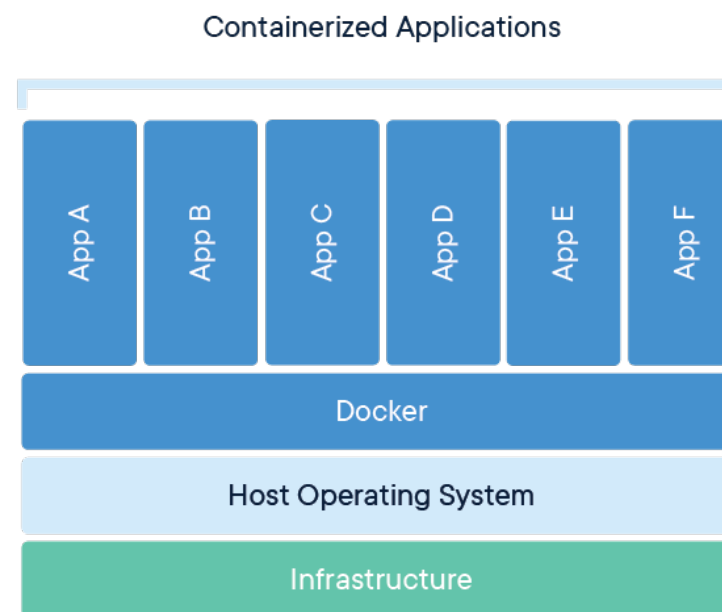
Τα containers είναι **ελαφρύτερα** από τα VMs και επιτρέπουν την εκτέλεση εφαρμογών σε απομονωμένα περιβάλλοντα χωρίς την ανάγκη πλήρους λειτουργικού συστήματος.

Docker: Το πιο δημοφιλές εργαλείο για την εκτέλεση containers, προσφέροντας μια ευέλικτη και αποδοτική πλατφόρμα



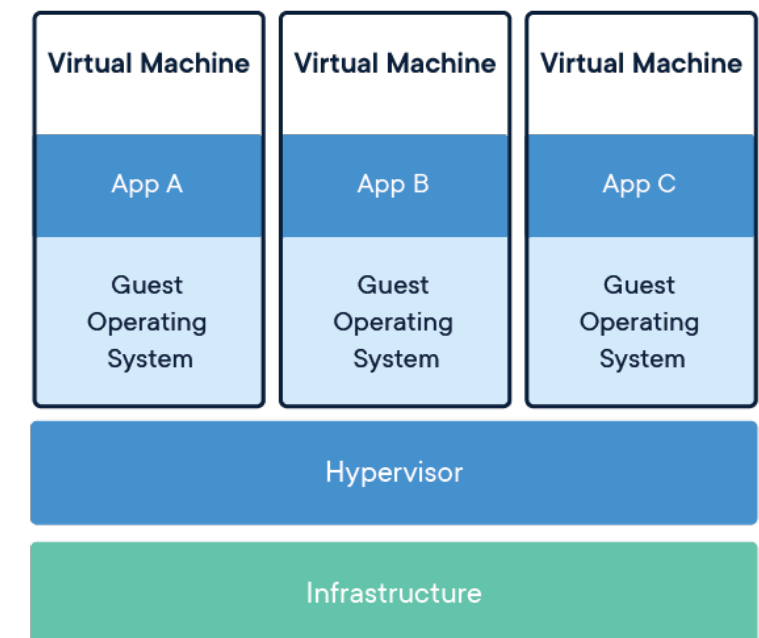
Hyper - V

ανάπτυξης και εκτέλεσης εφαρμογών.



Docker / Hypervisor

Kubernetes: Ένα εργαλείο για τη διαχείριση containers, επιτρέποντας την αυτοματοποιημένη κλιμάκωση και τη διαχείριση clusters.



Τεχνολογίες Εικονικοποίησης και Containers (kubernetes)

Τι είναι το Kubernetes;

Το Kubernetes, συχνά συντομευόμενο σε k8s, είναι μια πλατφόρμα ανοιχτού κώδικα που χρησιμοποιείται για την **αυτοματοποίηση της ανάπτυξης, της κλιμάκωσης και της διαχείρισης εφαρμογών σε container**. Σκέψου το σαν μία ορχήστρα που διευθύνει μια συμφωνία, όπου τα όργανα είναι τα containers και ο μαέστρος είναι το Kubernetes.

Βασικές Λειτουργίες του Kubernetes:

Ενορχήστρωση Container: Διαχειρίζεται την τοποθέτηση, την κλιμάκωση και την επικοινωνία μεταξύ των containers.

Αυτόματη Ανάκτηση: Επανεκκινεί αυτόματα τα containers που αποτυγχάνουν και διασφαλίζει την υψηλή διαθεσιμότητα των εφαρμογών.

Εξισορρόπηση Φορτίου: Κατανέμει το φορτίο εργασίας ομοιόμορφα μεταξύ των containers.

Αποθήκευση: Παρέχει τρόπους για την αποθήκευση και διαχείριση των δεδομένων των εφαρμογών.

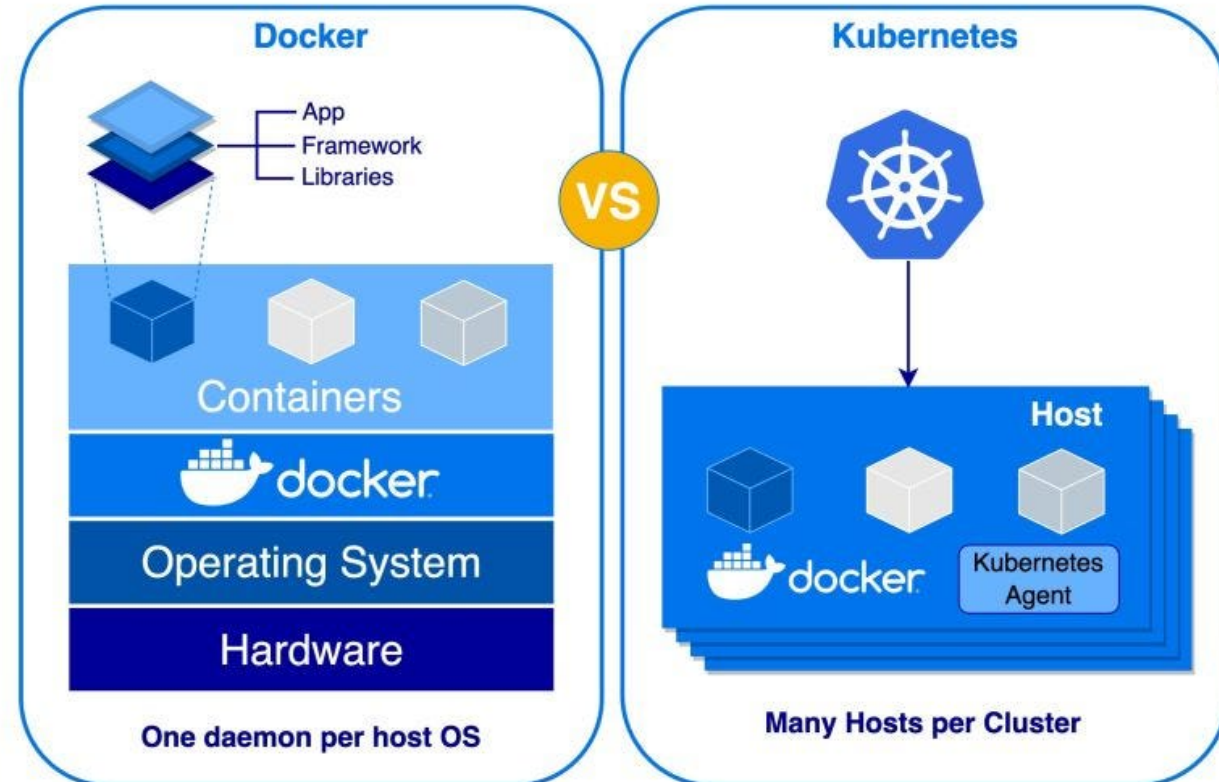
Δίκτυωση: Διαχειρίζεται τη δικτύωση μεταξύ των containers και με τον εξωτερικό κόσμο.

Τι είναι το Docker;

Το Docker είναι μια πλατφόρμα που επιτρέπει τη δημιουργία και την εκτέλεση εφαρμογών σε απομονωμένα περιβάλλοντα που ονομάζονται containers.

Ένα container είναι σαν ένα ελαφρύ εικονικό μηχάνημα που περιέχει όλες τις εξαρτήσεις που χρειάζεται μια εφαρμογή για να λειτουργήσει.

Βασικές Λειτουργίες του Docker:



Docker / Kubernetes

Δημιουργία Containers: Επιτρέπει τη δημιουργία containers από εικόνες (images) που περιέχουν όλες τις απαραίτητες πληροφορίες για την εκτέλεση μιας εφαρμογής.

Διαχείριση Containers: Επιτρέπει την εκκίνηση, τη διακοπή, την εκκίνηση ξανά και την αφαίρεση των containers.

Εικόνες Containers: Παρέχει ένα μηχανισμό για την δημιουργία, την αποθήκευση και την κοινή χρήση εικόνων containers.

Τεχνολογίες Εικονικοποίησης και Containers (kubernetes)

Πώς συνεργάζονται Kubernetes και Docker;

Το Kubernetes και το Docker συμπληρώνουν το ένα το άλλο και λειτουργούν συχνά μαζί. Το **Docker παρέχει τα containers** και το **Kubernetes παρέχει τη "κολλά" που τα συνδέει και τα οργανώνει σε ένα σύνολο.**

Docker δημιουργεί τα containers: Οι εφαρμογές πακετάρονται σε containers Docker, τα οποία είναι οι βασικές μονάδες που διαχειρίζεται το Kubernetes.

Kubernetes ορχηστρώνει τα containers:

Το Kubernetes λαμβάνει αυτά τα containers και τα τοποθετεί, τα κλιμακώνει και τα διαχειρίζεται σε ένα cluster υπολογιστών.

Απλοποίηση της διαχείρισης: Το Kubernetes αυτοματοποιεί πολλές από τις εργασίες που απαιτούνται για τη διαχείριση εφαρμογών σε μεγάλη κλίμακα, επιτρέποντας στους προγραμματιστές να επικεντρωθούν στην ανάπτυξη της εφαρμογής τους.

Πλεονεκτήματα της χρήσης Kubernetes και Docker:

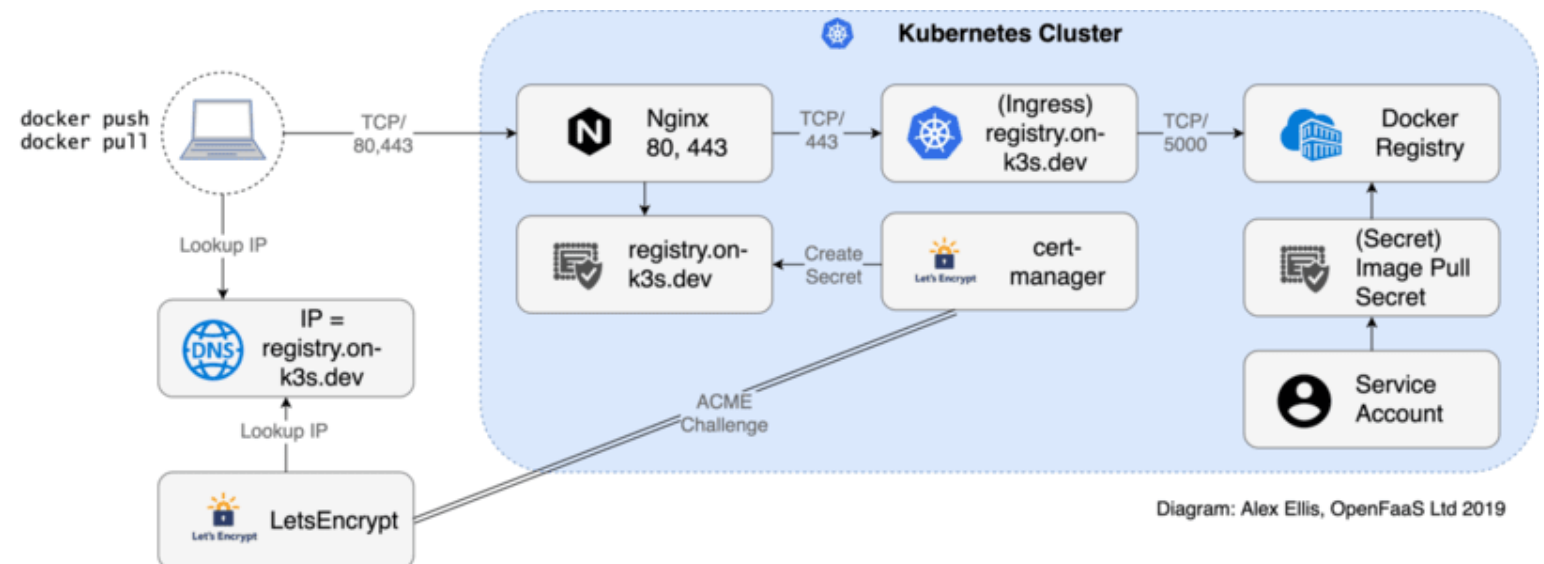
Αυξημένη ευελιξία: Οι εφαρμογές μπορούν να αναπτυχθούν, να δοκιμαστούν και να αναπτυχθούν γρηγορότερα.

Μειωμένος χρόνος διακοπής: Η αυτόματη ανάκτηση και η εξισορρόπηση φορτίου μειώνουν τον χρόνο διακοπής λειτουργίας των εφαρμογών.

Καλύτερη αξιοποίηση των πόρων: Οι πόροι του υπολογιστή μπορούν να χρησιμοποιηθούν πιο αποτελεσματικά.

Αυξημένη φορητότητα: Οι εφαρμογές μπορούν να εκτελεστούν σε οποιοδήποτε περιβάλλον που υποστηρίζει το Kubernetes.

Σε **σύνοψη**, το Kubernetes και το Docker είναι δύο ισχυρά εργαλεία που, όταν χρησιμοποιούνται μαζί, μπορούν να μεταμορφώσουν τον τρόπο με τον οποίο αναπτύσσουμε και διαχειριζόμαστε τις εφαρμογές μας. Παρέχουν έναν αποτελεσματικό και ευέλικτο τρόπο για τη δημιουργία και την εκτέλεση σύγχρονων εφαρμογών σε cloud περιβάλλοντα.



Docker / Kubernetes

Ασφάλεια στο Cloud

Η **ασφάλεια στο cloud** είναι μια κρίσιμη πτυχή για την αποδοχή και την υιοθέτηση του cloud computing από τις επιχειρήσεις. Οι πάροχοι cloud πρέπει να διασφαλίζουν ότι οι πόροι και τα δεδομένα των πελατών είναι ασφαλή.

Κρυπτογράφηση Δεδομένων

Η κρυπτογράφηση δεδομένων είναι απαραίτητη για την προστασία των δεδομένων κατά τη μεταφορά και την αποθήκευση. Τα δεδομένα πρέπει να είναι

κρυπτογραφημένα τόσο στις γραμμές επικοινωνίας όσο και στους αποθηκευτικούς χώρους.

Πολιτικές Ελέγχου Πρόσβασης

Η διαχείριση πρόσβασης στο cloud γίνεται μέσω αυστηρών πολιτικών ελέγχου πρόσβασης που βασίζονται σε ρόλους (RBAC). Αυτές οι πολιτικές διασφαλίζουν ότι μόνο εξουσιοδοτημένα άτομα μπορούν να έχουν πρόσβαση σε ευαίσθητα δεδομένα.

Κανονιστική Συμμόρφωση

Η συμμόρφωση με κανονισμούς, όπως το GDPR, το HIPAA, και το ISO 27001, είναι κρίσιμη για την προστασία των προσωπικών δεδομένων και την αποτροπή παραβιάσεων.



How To Be GDPR Compliant

1

OBTAINING CONSENT

Your terms of consent must be clear. Consent must be easily given and freely withdrawn at any time.

2

TIMELY BREACH NOTIFICATION

If a security breach occurs, you have 72 hours to report the data breach to both your customers and any data controllers, if your company is large enough to require a GDPR data controller. Failure to report breaches within this timeframe will lead to fines.

3

RIGHT TO DATA ACCESS

If your users request their existing data profile, you must be able to serve them with a fully detailed and free electronic copy of the data you've collected about them. This report must also include the various ways you're using their information.

4

RIGHT TO BE FORGOTTEN

Also known as the right to data deletion, once the original purpose or use of the customer data has been realized, your customers have the right to request that you totally erase their personal data.

5

DATA PORTABILITY

This gives users rights to their own data. They must be able to obtain their data from you and reuse that same data in different environments outside of your company.

6

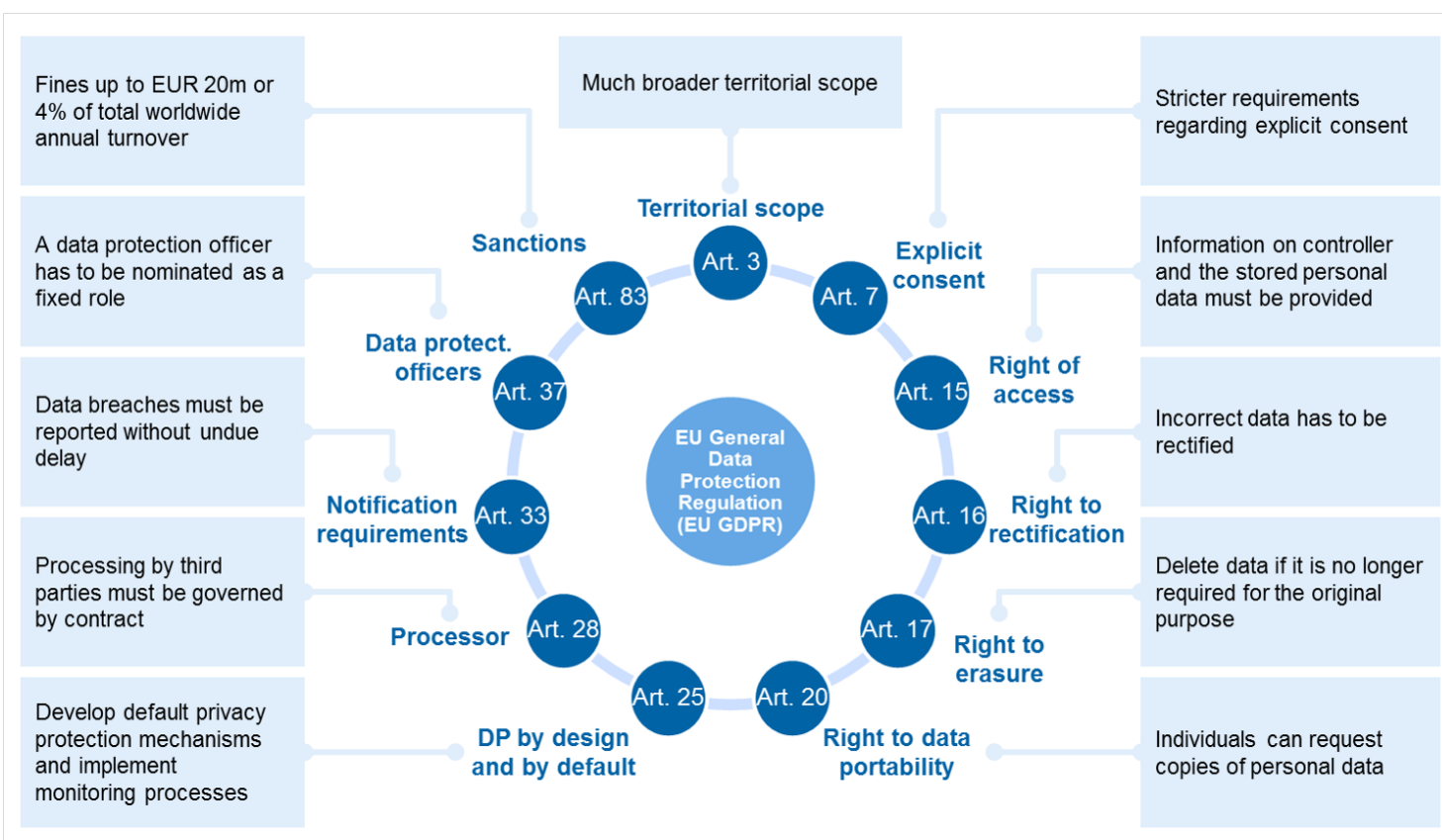
PRIVACY BY DESIGN

This section of GDPR requires companies to design their systems with the proper security protocols in place from the start. Failure to design your systems of data collection the right way will result in a fine.

7

POTENTIAL DATA PROTECTION OFFICERS

In some cases, your company may need to appoint a data protection officer (DPO). Whether or not you need an officer depends upon the size of your company and at what level you currently process and collect data.



GDPR

Σύγχρονες Τάσεις και Τεχνολογίες

Το πεδίο του cloud computing εξελίσσεται συνεχώς, και νέες τεχνολογίες και μοντέλα αλλάζουν τον τρόπο με τον οποίο οι επιχειρήσεις χρησιμοποιούν το cloud.

Multi-Cloud και Hybrid Cloud

Multi-Cloud: Η χρήση περισσότερων από έναν πάροχο cloud για την κάλυψη διαφορετικών αναγκών. Αυτό επιτρέπει στις επιχειρήσεις να επιλέγουν την καλύτερη υπηρεσία για κάθε χρήση.

Hybrid Cloud: Ο συνδυασμός ιδιωτικού και δημόσιου cloud για την εξασφάλιση

ευελιξίας και την προστασία ευαίσθητων δεδομένων.

Serverless Computing

Το **serverless** computing επιτρέπει στους προγραμματιστές να εκτελούν κώδικα χωρίς να διαχειρίζονται τους διακομιστές. Οι πάροχοι cloud διαχειρίζονται τη δυναμική κατανομή των πόρων.

Πλεονεκτήματα: Μείωση κόστους και απλούστευση της ανάπτυξης.

Παραδείγματα: AWS Lambda, Google Cloud Functions.

Edge Computing

Το **edge computing** μεταφέρει την επεξεργασία δεδομένων κοντά στον χρήστη ή τη συσκευή, μειώνοντας την καθυστέρηση και βελτιώνοντας την ταχύτητα απόκρισης.

Χρήση σε IoT συσκευές και βιομηχανικές εφαρμογές για τη διαχείριση μεγάλων ποσοτήτων δεδομένων σε πραγματικό χρόνο.

Προκλήσεις και Μέλλον του Cloud Computing

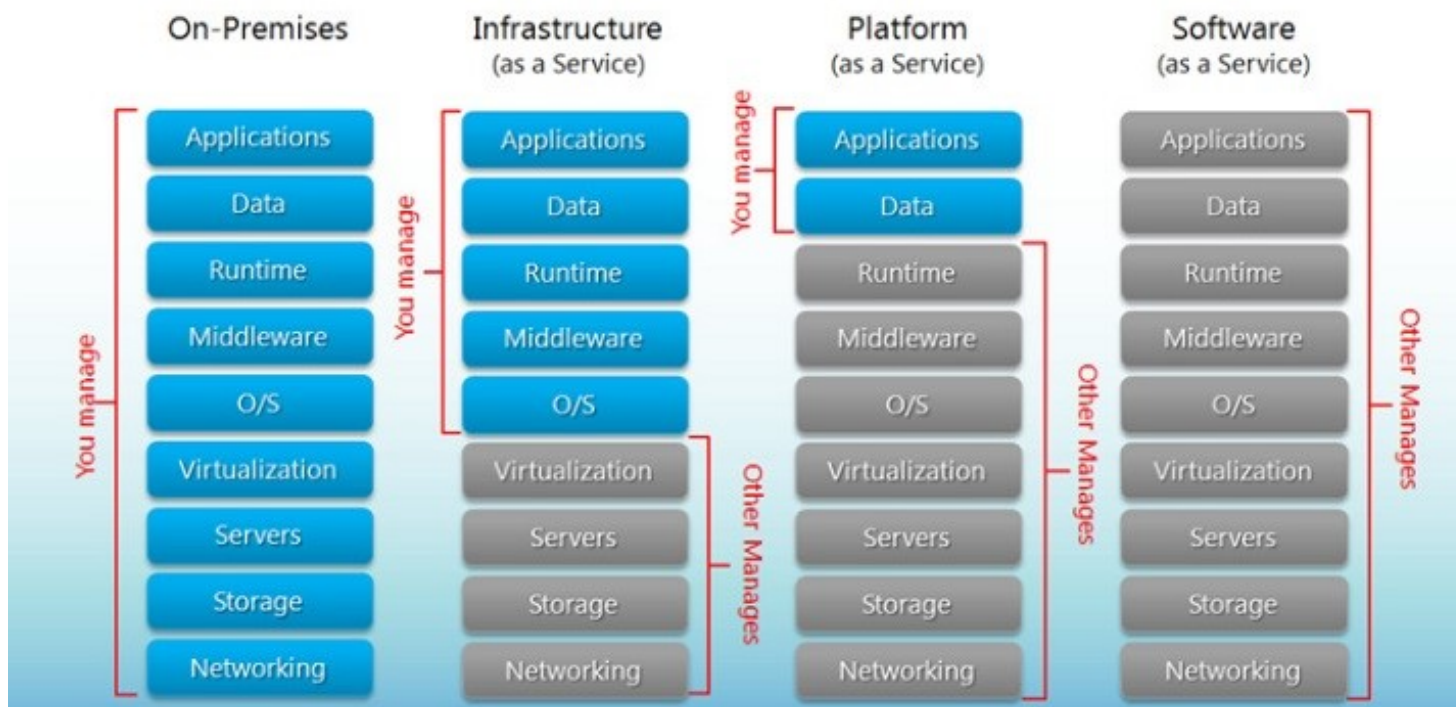
Παρά τις πολλές δυνατότητες, το cloud computing αντιμετωπίζει και προκλήσεις.

Ασφάλεια και Ιδιωτικότητα

Η διατήρηση της ασφάλειας των δεδομένων παραμένει μία από τις μεγαλύτερες προκλήσεις. Οι χρήστες πρέπει να εμπιστεύονται τους παρόχους για την προστασία των δεδομένων τους, ενώ οι πάροχοι πρέπει να διασφαλίζουν τη συμμόρφωση με τις κανονιστικές απαιτήσεις.

Κόστος και Διαχείριση Πόρων

Η διαχείριση του κόστους είναι ένα σημαντικό θέμα για τις επιχειρήσεις που χρησιμοποιούν cloud. Είναι απαραίτητο να βρεθεί η ισορροπία ανάμεσα στην κλιμάκωση και την αποδοτική χρήση των



Serverless Computing

Διάλεξη 10

Εικονικοποίηση και Διαχείριση Εικονικών Μηχανών (Virtualization and VM Management)

Στόχοι:

- Κατανόηση της έννοιας Virtual Machine.
- Ανάλυση των διαφορετικών τύπων hypervisors και των εφαρμογών τους.
- Εξέταση των τεχνικών διαχείρισης των εικονικών μηχανών και των βέλτιστων πρακτικών.
- Ανάλυση της ασφάλειας των εικονικών περιβαλλόντων και των προκλήσεων που προκύπτουν.
- Παρουσίαση των σύγχρονων εργαλείων διαχείρισης εικονικών μηχανών και των χρήσεών τους.

Εισαγωγή στην Εικονικοποίηση 1/2

Η εικονικοποίηση (virtualization) είναι μια τεχνολογία που επιτρέπει τη δημιουργία πολλαπλών εικονικών υπολογιστικών περιβαλλόντων σε έναν μόνο φυσικό διακομιστή. Κάθε εικονική μηχανή (VM) λειτουργεί σαν ξεχωριστός υπολογιστής, με δικούς της πόρους, όπως CPU, μνήμη, και αποθηκευτικό χώρο.

Πώς λειτουργούν;

Ένα λογισμικό, που ονομάζεται hypervisor, δημιουργεί ένα εικονικό περιβάλλον που μιμείται τον φυσικό hardware του υπολογιστή. Αυτό το περιβάλλον παρέχει στην εικονική μηχανή τους πόρους που χρειάζεται για να λειτουργήσει, όπως CPU, μνήμη και αποθηκευτικό χώρο.

Γιατί να χρησιμοποιήσεις μια εικονική μηχανή;

Δοκιμή νέων λειτουργικών συστημάτων ή εφαρμογών: Μπορείς να δοκιμάσεις νέα λειτουργικά συστήματα ή εφαρμογές σε μια εικονική μηχανή χωρίς να επηρεάσεις το κύριο λειτουργικό σύστημα του υπολογιστή σου.

Απομόνωση εφαρμογών: Μπορείς να τρέξεις εφαρμογές που απαιτούν συγκεκριμένες διαμορφώσεις σε μια εικονική μηχανή, χωρίς να επηρεάσεις άλλες εφαρμογές στον υπολογιστή σου.

Ανάπτυξη και δοκιμή: Οι προγραμματιστές χρησιμοποιούν εικονικές μηχανές για να

αναπτύξουν και να δοκιμάσουν εφαρμογές σε διαφορετικά περιβάλλοντα.

Εκπαίδευση: Οι εικονικές μηχανές είναι ιδανικές για εκπαιδευτικούς σκοπούς, καθώς οι μαθητές μπορούν να πειραματιστούν με διαφορετικά λειτουργικά συστήματα και εφαρμογές χωρίς κίνδυνο.

Διάσωση δεδομένων: Μπορείς να δημιουργήσεις στιγμιότυπα (snapshots) μιας εικονικής μηχανής, επιτρέποντάς σου να επαναφέρεις το σύστημα σε μια προηγούμενη κατάσταση σε περίπτωση προβλήματος.

Ποια είναι τα **πλεονεκτήματα** των **εικονικών μηχανών**;

Ευελιξία: Μπορείς να δημιουργήσεις και να διαγράψεις εικονικές μηχανές εύκολα και γρήγορα.

Απομόνωση: Κάθε εικονική μηχανή είναι απομονωμένη από τις άλλες, προσφέροντας μεγαλύτερη ασφάλεια.

Εξοικονόμηση πόρων: Οι εικονικές μηχανές μπορούν να μοιράζονται τους πόρους του φυσικού υπολογιστή, βελτιστοποιώντας τη χρήση του hardware.

Φορητότητα: Μπορείς να μεταφέρεις μια εικονική μηχανή σε άλλον υπολογιστή.



Virtual Machines

Εισαγωγή στην Εικονικοποίηση 2/2

Δημοφιλή λογισμικά για εικονικές μηχανές:

VirtualBox: Δωρεάν και ανοιχτού κώδικα, προσφέροντας μια ευέλικτη λύση για οικιακούς χρήστες και προγραμματιστές.

VMware Workstation: Εμπορικό λογισμικό με περισσότερες προηγμένες λειτουργίες.

Hyper-V: Ενσωματωμένο στο Windows, προσφέροντας μια ολοκληρωμένη λύση για εικονικοποίηση.

Σε τι διαφέρουν οι εικονικές μηχανές από τα containers;

Ενώ **οι εικονικές μηχανές εμπεριέχουν ένα ολόκληρο λειτουργικό σύστημα**, τα **containers μοιράζονται τον ίδιο πυρήνα λειτουργικού συστήματος με τον υπολογιστή-ξενιστή**, μειώνοντας έτσι το μέγεθος και τους πόρους που καταναλώνουν.

Σε ποια περίπτωση θα χρειαστείς μια εικονική μηχανή;

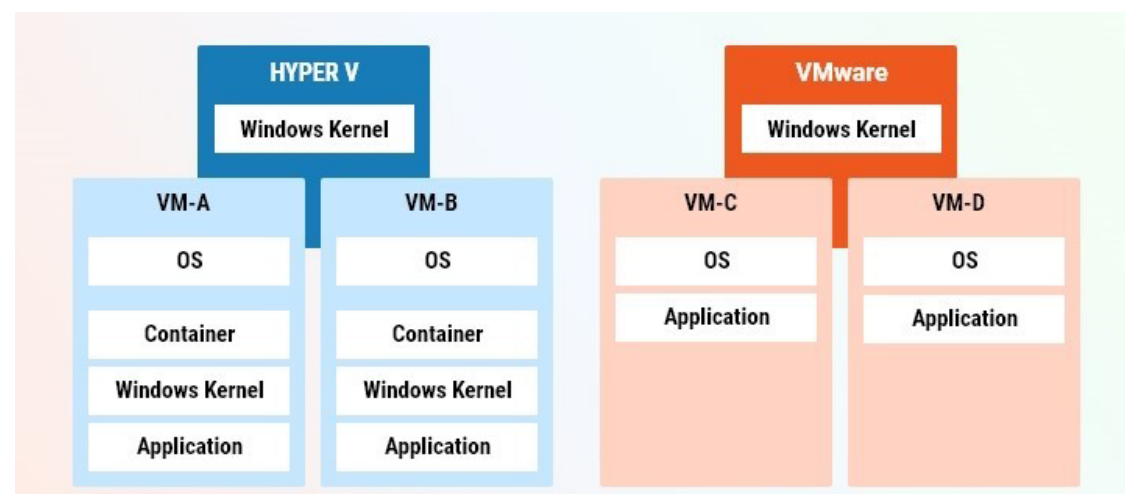
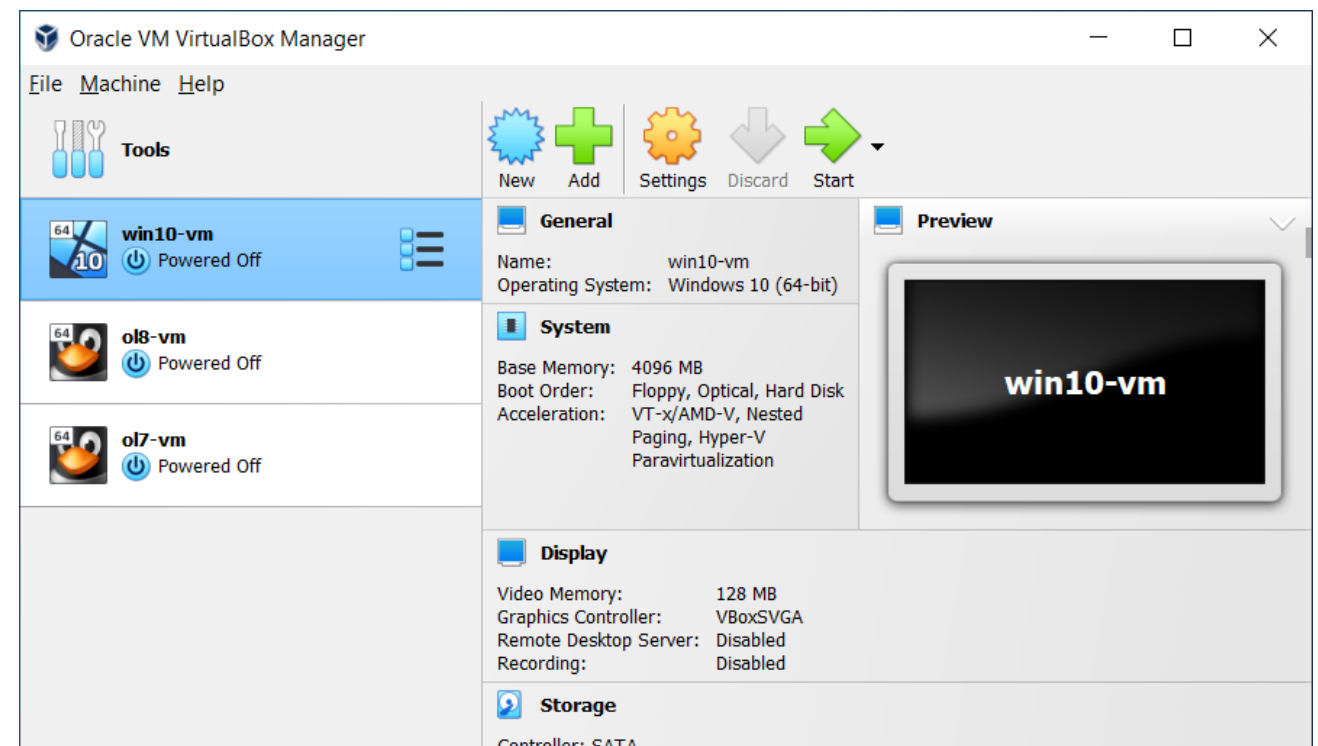
Αν θέλεις να **δοκιμάσεις** ένα νέο λειτουργικό σύστημα.

Αν χρειάζεσαι να **τρέξεις πολλές εφαρμογές** ταυτόχρονα.

Αν θέλεις να **αναπτύξεις** και να **δοκιμάσεις** εφαρμογές σε ένα απομονωμένο περιβάλλον.

Αν θέλεις να **δημιουργήσεις** ένα **εργαστήριο** για **εκπαιδευτικούς** σκοπούς.

Σε γενικές γραμμές, οι εικονικές μηχανές προσφέρουν έναν ευέλικτο και ασφαλή τρόπο για να εκτελέσεις πολλαπλά λειτουργικά συστήματα και εφαρμογές σε έναν μόνο υπολογιστή.



Hyper - V VS vmware

Τύποι Hypervisors

Οι **hypervisors** είναι λογισμικά που επιτρέπουν τη δημιουργία και τη διαχείριση εικονικών μηχανών. Διαχωρίζονται σε δύο κύριους τύπους:

Type 1 Hypervisors (**Bare-Metal**)

Οι Type 1 Hypervisors λειτουργούν απευθείας στο υλικό του διακομιστή, χωρίς την ανάγκη ενδιάμεσου λειτουργικού συστήματος. Είναι οι πιο αποδοτικοί και χρησιμοποιούνται σε περιβάλλοντα που απαιτούν υψηλή απόδοση και αξιοπιστία.

Παραδείγματα: VMware ESXi, Microsoft Hyper-V, Xen.

Πλεονεκτήματα: Υψηλή απόδοση, καλύτερη ασφάλεια, και άμεση πρόσβαση στο υλικό.

Χρήση: Ιδανικοί για data centers και μεγάλα περιβάλλοντα cloud.

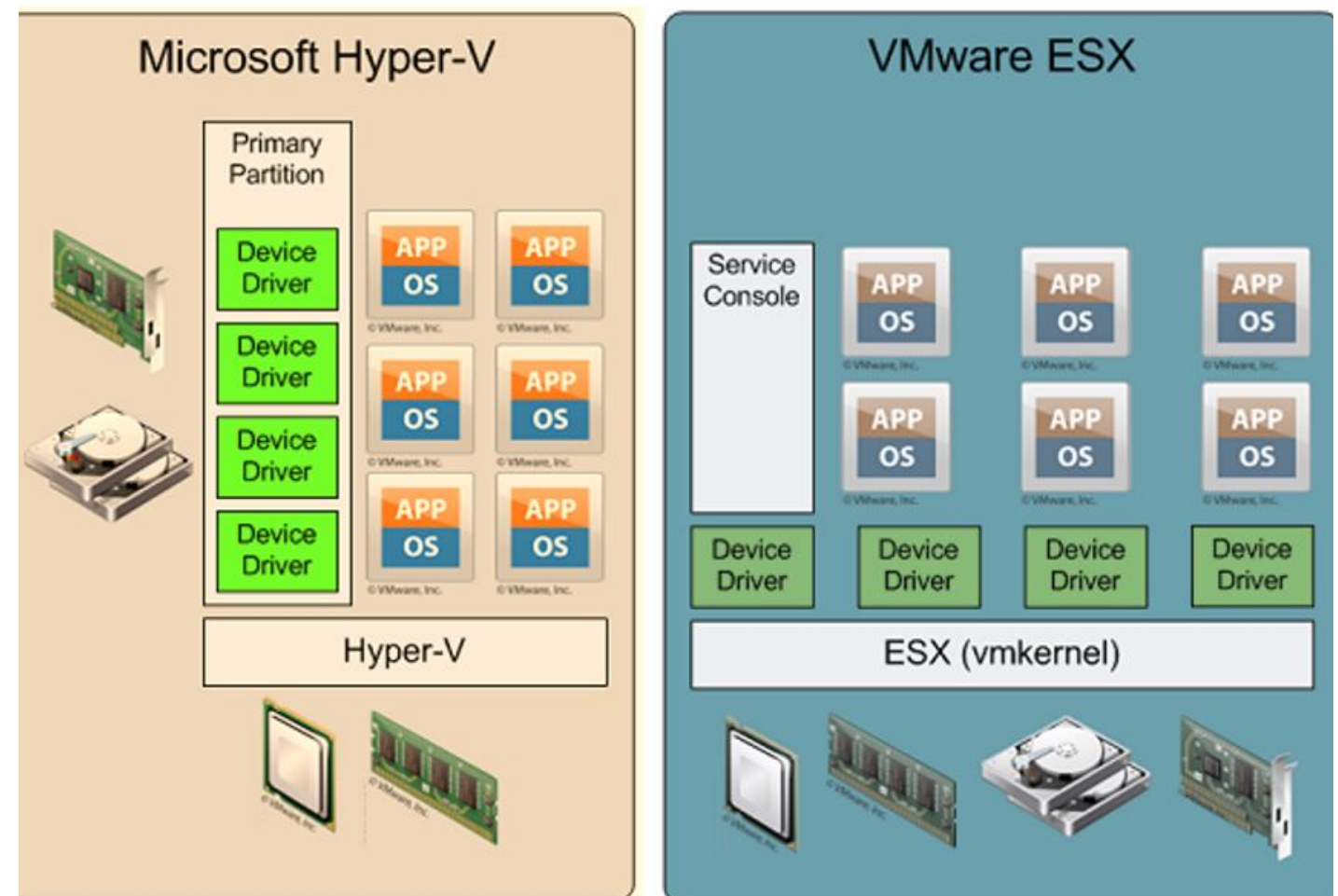
Type 2 Hypervisors (**Hosted**)

Οι Type 2 Hypervisors λειτουργούν πάνω από ένα λειτουργικό σύστημα, χρησιμοποιώντας το για την επικοινωνία με το υλικό του διακομιστή.

Παραδείγματα: Oracle VirtualBox, VMware Workstation.

Πλεονεκτήματα: Ευκολία στη ρύθμιση και στη χρήση, κατάλληλοι για περιβάλλοντα ανάπτυξης και δοκιμών.

Χρήση: Συνήθως χρησιμοποιούνται σε μικρότερα περιβάλλοντα ή για προσωπική χρήση από προγραμματιστές.



Hyper - V VS vmware

Τεχνικές Διαχείρισης Εικονικών Μηχανών

Η διαχείριση των **εικονικών μηχανών** είναι κρίσιμη για τη διασφάλιση της αποδοτικότητας και της αξιοπιστίας των πόρων του data center.

Κατανομή και Βελτιστοποίηση Πόρων

-**Dynamic Resource Allocation:** Η αυτόματη προσαρμογή των πόρων (π.χ., CPU, RAM) ανάλογα με τις ανάγκες κάθε VM.

-**Overcommitment:** Η δυνατότητα να διατεθούν περισσότεροι πόροι από αυτούς που είναι διαθέσιμοι φυσικά, αυξάνοντας την αποδοτικότητα.

Snapshots και Backups

Τα snapshots επιτρέπουν τη δημιουργία σημείων επαναφοράς για τις VMs, διευκολύνοντας την ανάκτηση από σφάλματα ή την επιστροφή σε προηγούμενη κατάσταση.

-**Διαφορικά Snapshots:** Αποθηκεύουν μόνο τις αλλαγές από το προηγούμενο snapshot, εξοικονομώντας χώρο.

-**Full Backups:** Αντίγραφα ασφαλείας ολόκληρης της VM, χρήσιμα για αποκατάσταση μετά από καταστροφές.

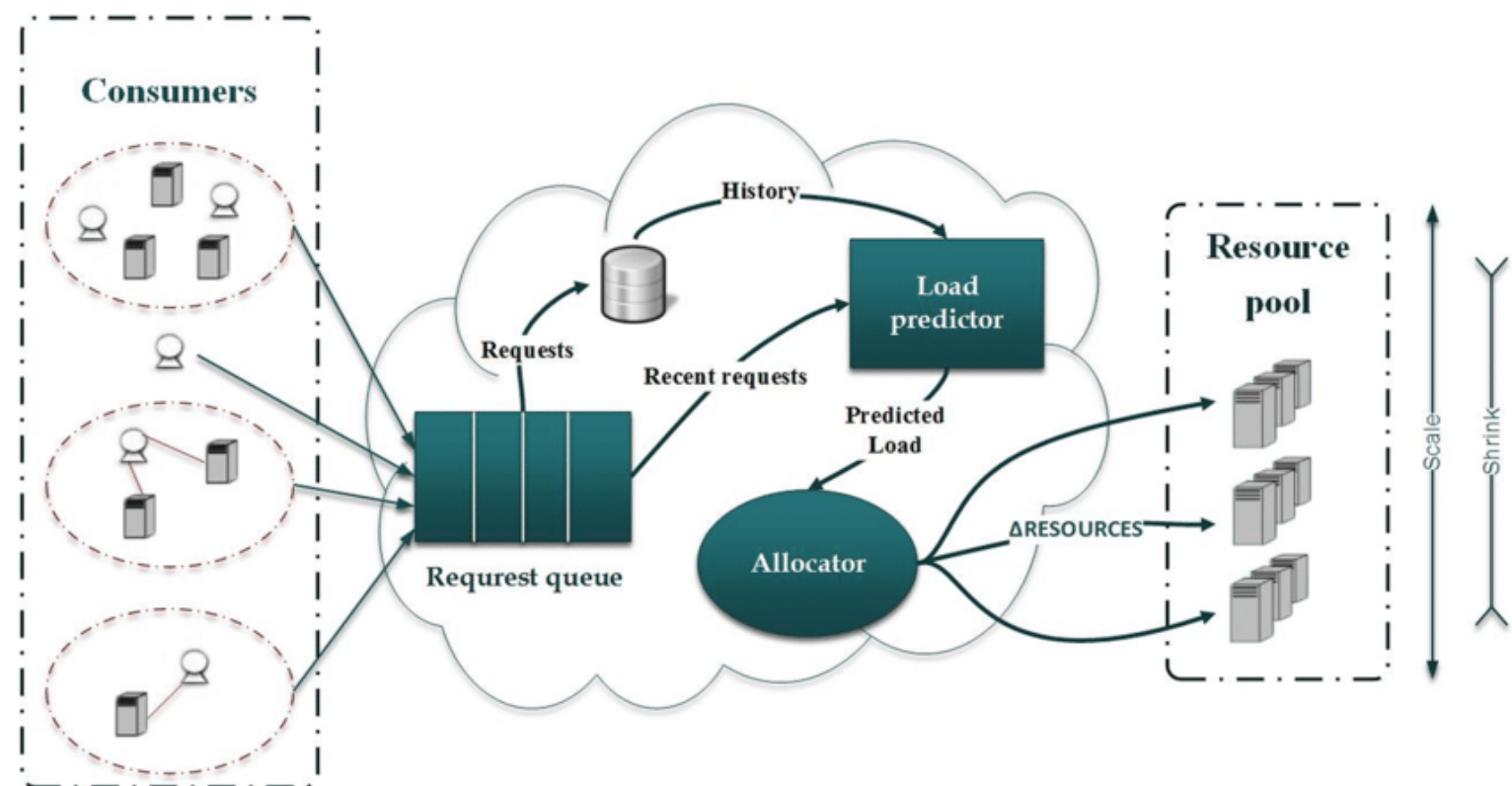
Live Migration

Η live migration επιτρέπει τη μετακίνηση μιας VM από έναν διακομιστή σε άλλον

χωρίς να διακόπτεται η λειτουργία της. Αυτή η τεχνική είναι ιδιαίτερα χρήσιμη για:

Load Balancing: Μετακίνηση VMs ώστε να εξισορροπηθεί η χρήση πόρων.

Αποφυγή Συντήρησης: Μεταφορά VMs πριν την εκτέλεση συντήρησης σε έναν διακομιστή.



Dynamic resource provisioning

Ασφάλεια στις Εικονικές Μηχανές

Η ασφάλεια στις εικονικές μηχανές είναι κρίσιμη, καθώς μια παραβίαση σε μία VM μπορεί να επηρεάσει ολόκληρο το σύστημα.

Απομόνωση και Προστασία Πόρων

Η απομόνωση των VMs είναι σημαντική για την αποτροπή διαρροής δεδομένων και την προστασία από κακόβουλο λογισμικό.

Sandboxing: Οι VMs λειτουργούν σε απομονωμένα περιβάλλοντα, μειώνοντας



Sandboxing

την πιθανότητα διάχυσης επιθέσεων.

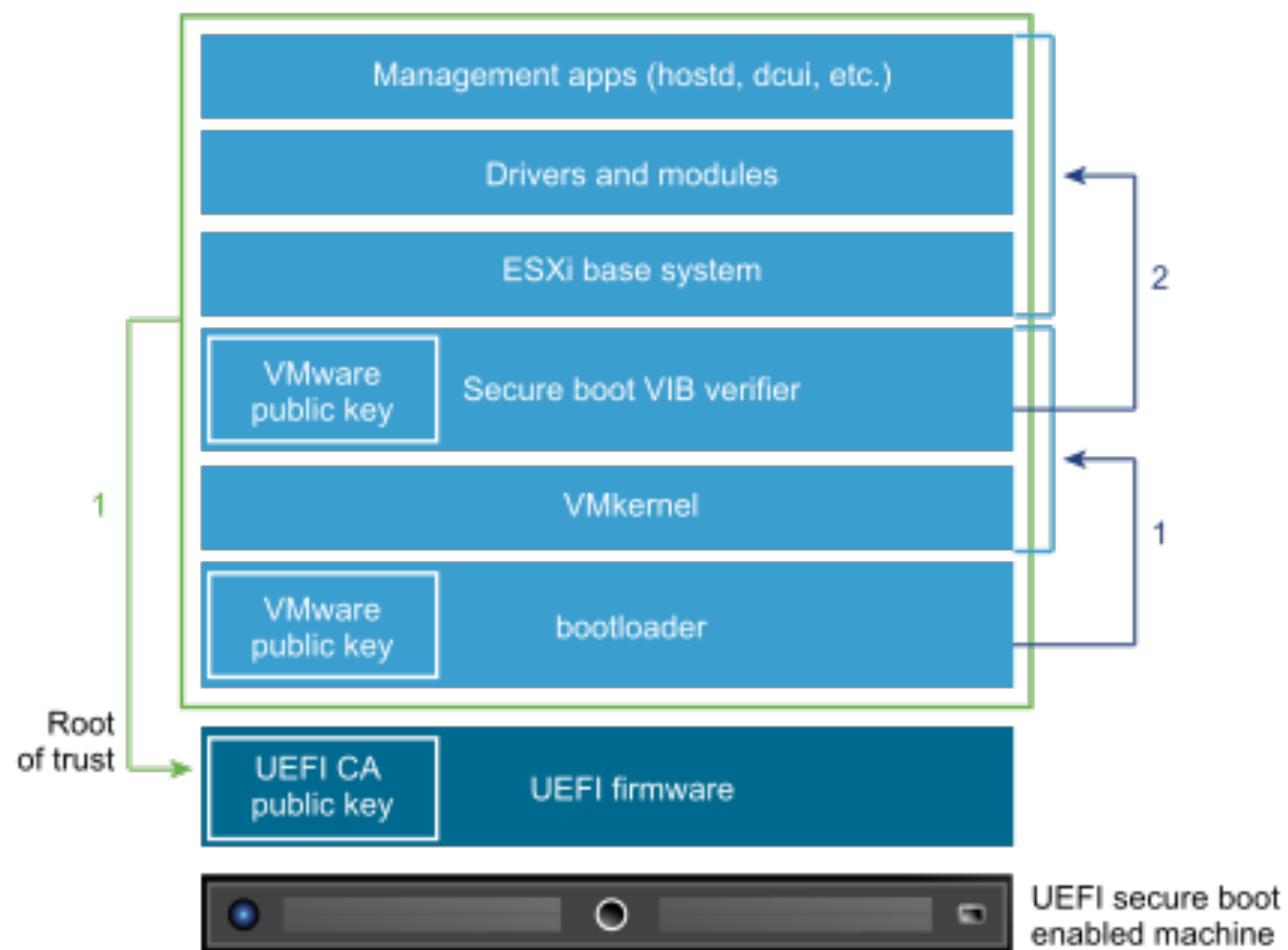
Secure Boot: Διασφαλίζει ότι οι VMs εκκινούν μόνο με έγκυρο και πιστοποιημένο λογισμικό.

Ασφάλεια Hypervisor

Οι hypervisors αποτελούν κρίσιμο σημείο στην αλυσίδα ασφάλειας και πρέπει να είναι σωστά προστατευμένοι.

Regular Patching: Εφαρμογή ενημερώσεων για την αντιμετώπιση γνωστών ευπαθειών.

Access Control: Περιορισμός της πρόσβασης στη διαχείριση των hypervisors σε εξουσιοδοτημένους χρήστες.



UEFI Secure boot

Σύγχρονα Εργαλεία και Τεχνολογίες Διαχείρισης

Υπάρχουν πολλά εργαλεία που διευκολύνουν τη διαχείριση των εικονικών μηχανών και την κλιμάκωση των πόρων.

VMware vSphere

Το VMware vSphere είναι από τα πιο δημοφιλή εργαλεία διαχείρισης εικονικών υποδομών, επιτρέποντας την κεντρική διαχείριση πολλαπλών ESXi hosts.

-vCenter Server: Παρέχει μια κεντρική κονσόλα για τη διαχείριση των VMs και των hosts.

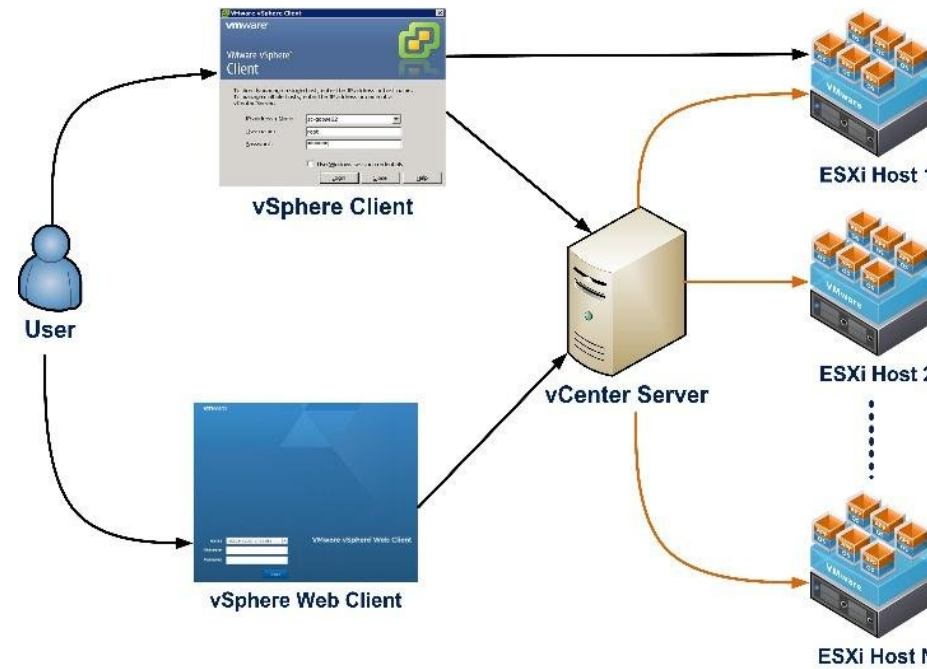
-High Availability (HA): Διασφαλίζει τη συνεχή λειτουργία των VMs σε περίπτωση σφάλματος hardware.

KVM (Kernel-based Virtual Machine)

Το KVM είναι ένα hypervisor ανοιχτού κώδικα που είναι ενσωματωμένο στον πυρήνα του Linux και χρησιμοποιείται ευρέως σε διανομές όπως το **Ubuntu** και το **Red Hat**.

Πλεονεκτήματα: Ευελιξία, δυνατότητα κλιμάκωσης και ενσωμάτωση με εργαλεία διαχείρισης όπως το OpenStack.

Χρήση: Ιδανικό για περιβάλλοντα που βασίζονται σε Linux και απαιτούν κλιμάκωση.



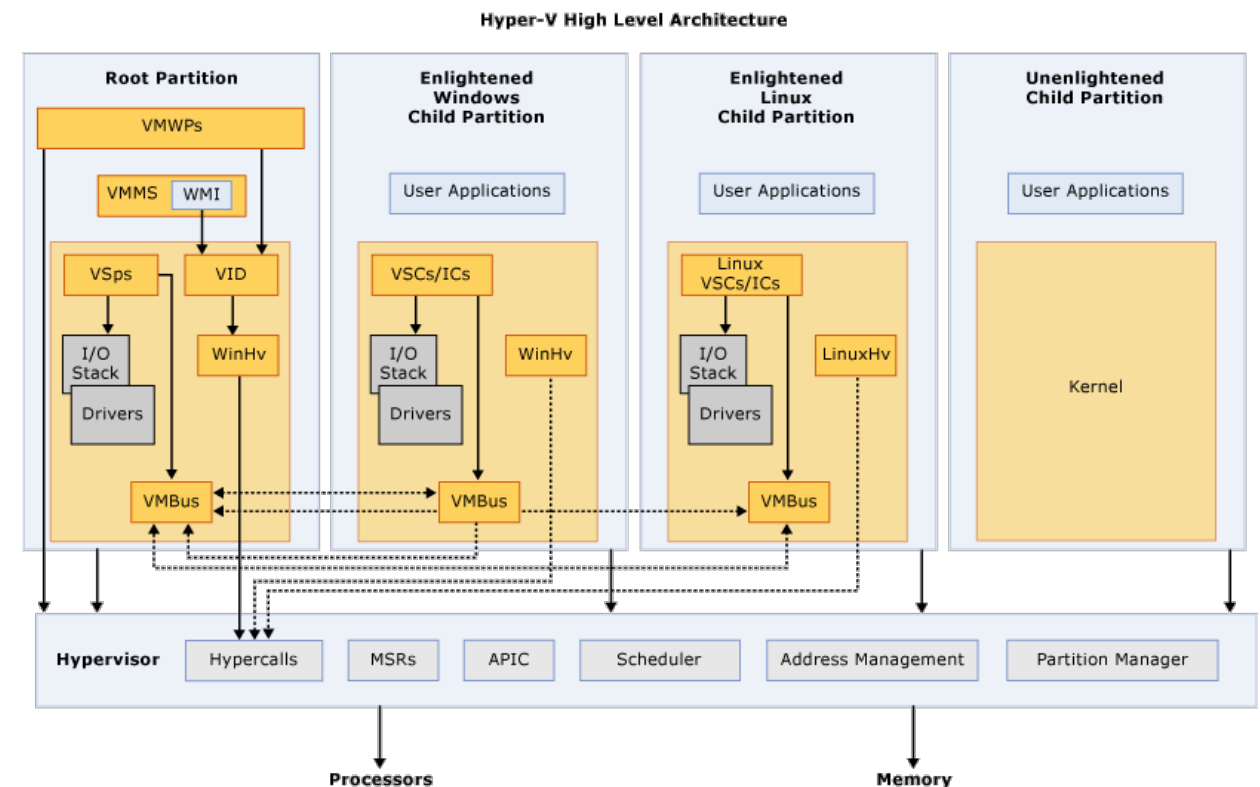
vSphere on Server setting VMs

Πλεονεκτήματα: Εύκολη ενσωμάτωση με προϊόντα Microsoft, όπως το Azure και το Active Directory.

Χρήση: Ιδανικό για επιχειρήσεις που χρησιμοποιούν ήδη το οικοσύστημα της Microsoft.

Microsoft Hyper-V

Το **Hyper-V** είναι η λύση της Microsoft για την εικονικοποίηση, ενσωματωμένη στα λειτουργικά συστήματα Windows Server.



Microsoft Hyper - V

KVM (Kernel-based Virtual Machine)

Το KVM αξιοποιεί τις δυνατότητες του υλικού για την εικονικοποίηση (όπως οι τεχνολογίες VT-x της Intel ή AMD-V) για να δημιουργήσει και να διαχειριστεί εικονικές μηχανές. Ο πυρήνας του Linux παρακολουθεί και διαχειρίζεται αυτές τις εικονικές μηχανές, διασφαλίζοντας ότι καθεμία λαμβάνει τους πόρους που της αναλογούν.

Σύγκριση με άλλες τεχνολογίες εικονικοποίησης:

Το KVM είναι μια από τις πιο δημοφιλείς τεχνολογίες εικονικοποίησης για Linux. Άλλες δημοφιλείς επιλογές περιλαμβάνουν το VMware και το VirtualBox. Ωστόσο, **το KVM ξεχωρίζει για την απόδοσή του, την ευελιξία του και το γεγονός ότι είναι ανοιχτού κώδικα.**

Σε **ποιους απευθύνεται το KVM;**

Διαχειριστές συστημάτων: Για τη διαχείριση πολλαπλών λειτουργικών συστημάτων σε έναν μόνο υπολογιστή.

Προγραμματιστές: Για την ανάπτυξη και τη δοκιμή εφαρμογών.

Εκπαιδευτικοί: Για τη δημιουργία εικονικών εργαστηρίων.

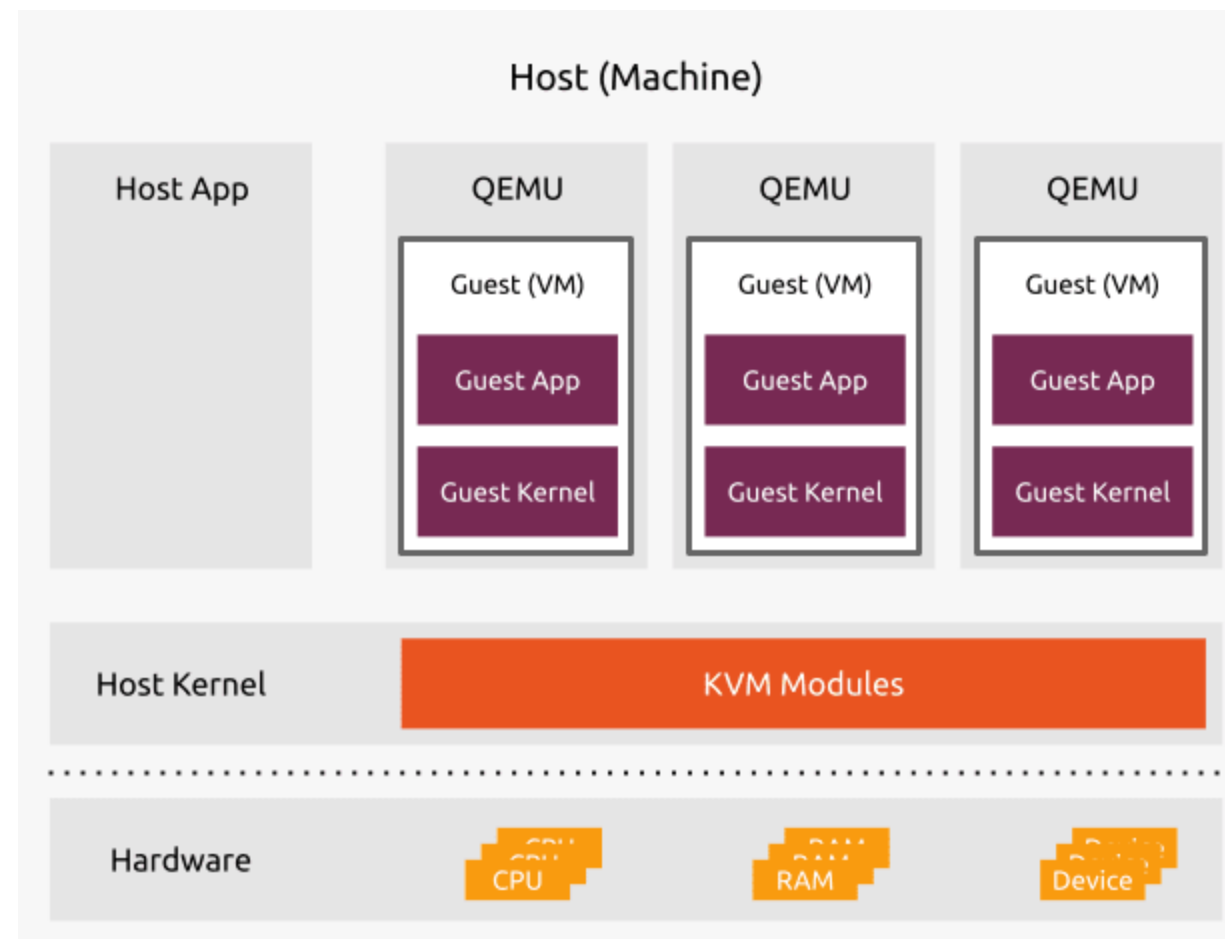
Οποιοσδήποτε θέλει να αξιοποιήσει στο έπακρο τους πόρους του υπολογιστή του.

Συνοπτικά:

Το KVM είναι μια ισχυρή και ευέλικτη τεχνολογία εικονικοποίησης που προσφέρει πολλές δυνατότητες. Είτε είσαι διαχειριστής συστήματος, προγραμματιστής ή απλώς ένας χρήστης που θέλει να πειραματιστεί, το KVM μπορεί να είναι ένα πολύτιμο εργαλείο για σένα.

Μετάβαση σε Containers

Η ανάπτυξη των containers προσφέρει μια πιο **ελαφριά** και **ευέλικτη** μορφή **εικονικοποίησης**. Ωστόσο, η μετάβαση από τις παραδοσιακές VMs στα containers μπορεί να απαιτεί σημαντική **αναδιοργάνωση των υποδομών**.



Kernel - based VM

Διάλεξη 11

Διαχείριση Πόρων στο Νέφος (Cloud Resource Management)

Στόχοι:

- Κατανόηση των βασικών αρχών της διαχείρισης πόρων στο cloud.
- Ανάλυση των τεχνικών δυναμικής κατανομής πόρων και αυτόματης κλιμάκωσης.
- Εξέταση της παρακολούθησης της απόδοσης και της βελτιστοποίησης χρήσης πόρων.
- Παρουσίαση των προκλήσεων και των βέλτιστων πρακτικών για τη διαχείριση πόρων.
- Προκλήσεις και διαχείριση πόρων.

Εισαγωγή στη Διαχείριση Πόρων στο Νέφος / Δυναμική Κατανομή Πόρων

Η διαχείριση πόρων στο νέφος

αναφέρεται στη διαδικασία βέλτιστης κατανομής και χρήσης των υπολογιστικών πόρων, όπως η CPU, η μνήμη, ο αποθηκευτικός χώρος και το δίκτυο, σε ένα περιβάλλον cloud. Ο στόχος είναι να εξασφαλιστεί ότι οι διαθέσιμοι πόροι χρησιμοποιούνται αποδοτικά, παρέχοντας τη μέγιστη δυνατή απόδοση με το ελάχιστο κόστος.

Σημασία της Διαχείρισης Πόρων στο Cloud

Η σωστή διαχείριση των πόρων είναι κρίσιμη για τη λειτουργία των εφαρμογών στο cloud, καθώς εξασφαλίζει:

Αποδοτικότητα: Μέγιστη χρήση των διαθέσιμων πόρων με την ελάχιστη κατανάλωση.

Κλιμακωσιμότητα: Δυνατότητα προσαρμογής της χρήσης πόρων ανάλογα με τη ζήτηση.

Οικονομική Βιωσιμότητα: Μείωση του κόστους λειτουργίας μέσω της ορθολογικής χρήσης πόρων.

Υποδομές Cloud και Κατανομή Πόρων

Οι υποδομές cloud χωρίζονται σε τρεις κύριες κατηγορίες: IaaS, PaaS, και SaaS, οι οποίες απαιτούν διαφορετικές προσεγγίσεις για τη διαχείριση των πόρων.

Η **δυναμική κατανομή** πόρων είναι η δυνατότητα προσαρμογής των πόρων που διατίθενται σε εφαρμογές ανάλογα με τις ανάγκες τους, επιτρέποντας την αποφυγή σπατάλης και την αποδοτική χρήση.

Κατανομή Πόρων σε Υποδομές IaaS

-Elasticity: Η δυνατότητα προσθήκης ή αφαίρεσης πόρων σε πραγματικό χρόνο για την προσαρμογή στη ζήτηση.

-Scaling Up/Down: Προσαρμογή των πόρων σε μία VM (π.χ., προσθήκη CPU).

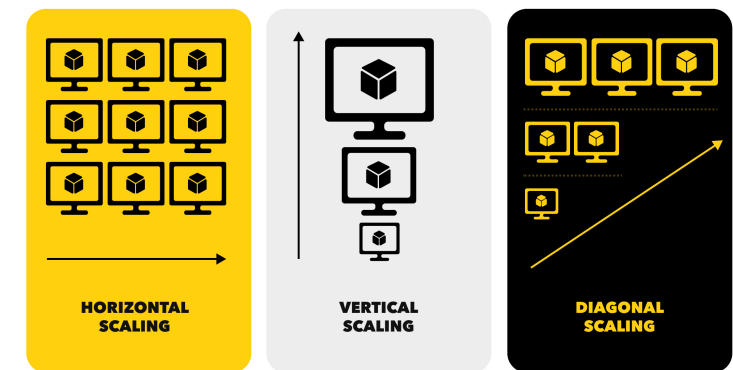
-Scaling Out/In: Προσθήκη ή αφαίρεση VMs για την κάλυψη της αυξανόμενης ζήτησης.

Αυτόματη Κλιμάκωση (Autoscaling)

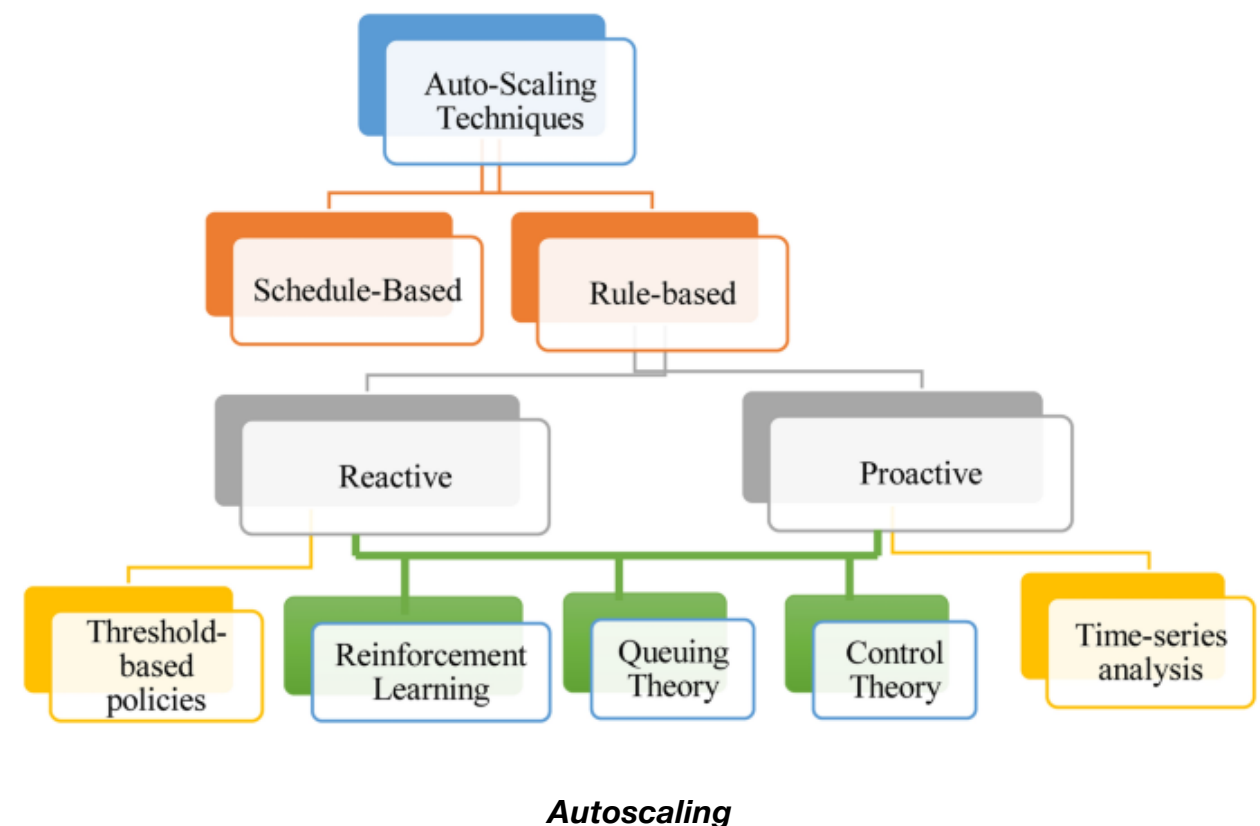
Η αυτόματη κλιμάκωση επιτρέπει στις εφαρμογές να προσαρμόζουν τους πόρους τους δυναμικά, ανάλογα με τις μεταβολές στη χρήση.

-Threshold-based Scaling:

Προσαρμόζει τους πόρους βάσει προκαθορισμένων ορίων χρήσης, όπως η χρήση CPU.



-Predictive Autoscaling: Χρησιμοποιεί αλγόριθμους μηχανικής μάθησης για να προβλέψει μελλοντική ζήτηση.



Παρακολούθηση και Βελτιστοποίηση της Απόδοσης

Η **παρακολούθηση** της **απόδοσης** είναι απαραίτητη για τη διαχείριση των πόρων στο cloud, καθώς επιτρέπει την αναγνώριση προβλημάτων και την προσαρμογή των ρυθμίσεων για καλύτερη απόδοση.

Εργαλεία Παρακολούθησης

Υπάρχουν πολλά εργαλεία που βοηθούν στη συνεχή παρακολούθηση των πόρων και της απόδοσης των εφαρμογών στο cloud:

Amazon CloudWatch: Παρέχει λεπτομερείς μετρήσεις και ειδοποιήσεις για την απόδοση των πόρων στο AWS.

Prometheus: Ανοιχτού κώδικα εργαλείο για την παρακολούθηση των μετρήσεων και τη συλλογή δεδομένων.

Azure Monitor: Προσφέρει πλήρη ορατότητα στις υπηρεσίες που εκτελούνται στο Microsoft Azure.

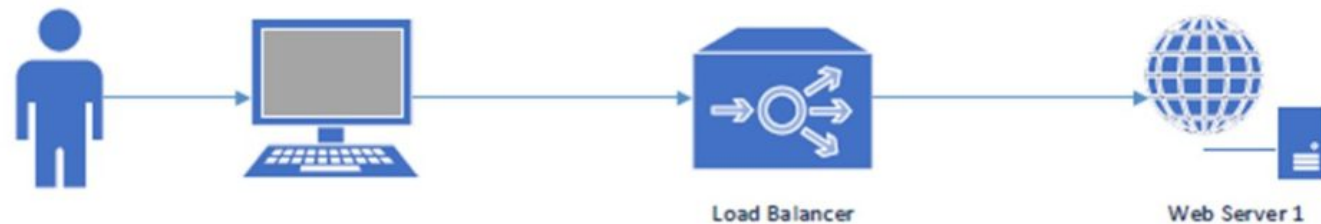
Βελτιστοποίηση Χρήσης Πόρων

Η **βελτιστοποίηση της χρήσης πόρων** περιλαμβάνει τη διαδικασία αναγνώρισης και εξάλειψης των πλεονασμάτων, όπως η απενεργοποίηση μη χρησιμοποιούμενων VMs ή η προσαρμογή των ρυθμίσεων των containers.

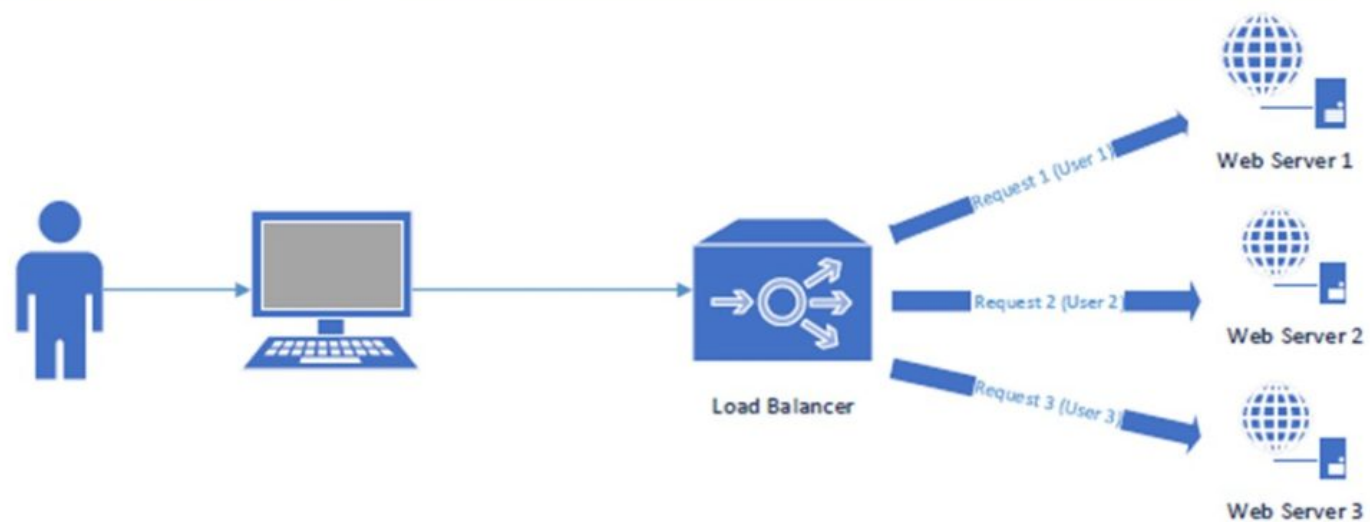
Right-sizing: Προσαρμογή των πόρων στις πραγματικές ανάγκες των εφαρμογών.

Load Balancing: Κατανομή των αιτημάτων μεταξύ πολλαπλών VMs για να αποφεύγεται η υπερφόρτωση.

Non Scalable: Single Server serving all the users. If the server cannot handle large number of users, it will not be able serve the subsequent requests.



Horizontally Scalable: Multiple Servers available to all the users. Depending on the load on each servers, request will be served by allocated server.



© 2016, NITRIX-Reloaded

Load balacing

Prometheus

Το **Prometheus** είναι ένα ισχυρό, ανοιχτού κώδικα εργαλείο που έχει σχεδιαστεί για να συλλέγει μετρήσεις από εφαρμογές και υποδομές σε πραγματικό χρόνο. Είναι ιδανικό για την παρακολούθηση της απόδοσης συστημάτων, την ανίχνευση προβλημάτων και την προειδοποίηση για πιθανές αστοχίες. Σκέψου το σαν έναν συνεχή παλμό που λαμβάνει πληροφορίες από όλο σου το σύστημα, επιτρέποντάς σου να έχεις μια ολοκληρωμένη εικόνα της κατάστασής του.

Βασικά Χαρακτηριστικά:

Συλλογή μετρήσεων: Το Prometheus συλλέγει μετρήσεις από διάφορες πηγές, όπως εφαρμογές, συστήματα λειτουργίας, βάσεις δεδομένων και υπηρεσίες cloud. Αυτές οι μετρήσεις μπορούν να είναι αριθμητικές τιμές (π.χ. χρήση CPU, μνήμη), boolean τιμές (π.χ. υπηρεσία σε λειτουργία ή όχι) ή ακόμη και κείμενο.

Αποθήκευση δεδομένων: Τα δεδομένα που συλλέγονται αποθηκεύονται σε μια βάση δεδομένων χρόνου σειράς, επιτρέποντας την εύκολη αναζήτηση και ανάλυση ιστορικών δεδομένων.

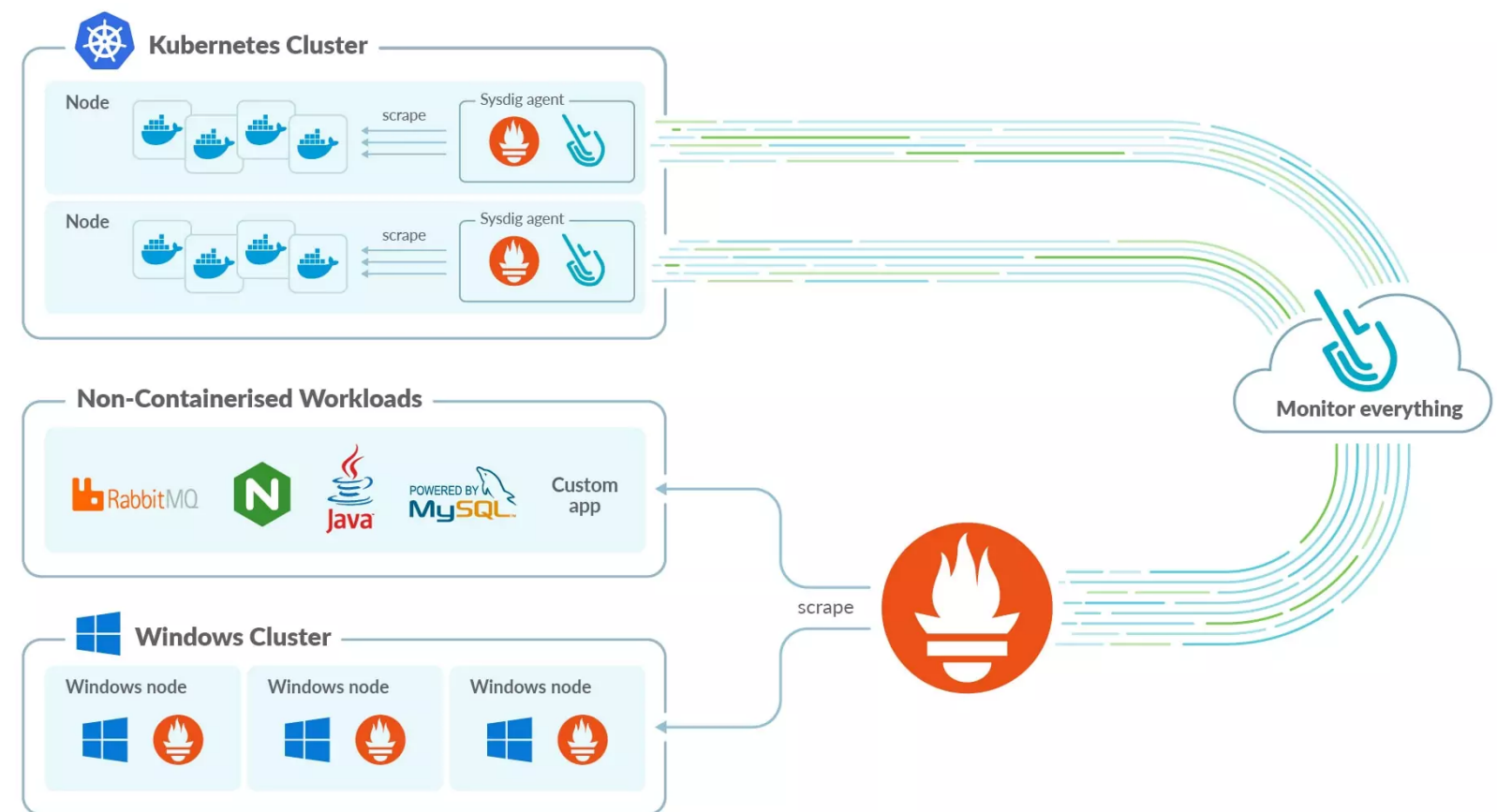
Ερωτήματα: Μια ισχυρή γλώσσα

ερωτημάτων, η PromQL, επιτρέπει στους χρήστες να αναζητούν, να φιλτράρουν και να οπτικοποιούν τα δεδομένα που έχουν συλλεχθεί.

Ειδοποιήσεις: Το Prometheus μπορεί να δημιουργήσει αυτόματα ειδοποιήσεις όταν εντοπιστούν προβλήματα, όπως η

υπερβολική χρήση πόρων ή η αποτυχία μιας υπηρεσίας.

Οπτικοποίηση: Τα δεδομένα μπορούν να οπτικοποιηθούν με τη χρήση γραφημάτων και πινάκων, καθιστώντας εύκολη την κατανόηση των τάσεων και των προβλημάτων.



Prometheus

Προκλήσεις στη Διαχείριση Πόρων στο Νέφος

Παρά τα οφέλη, η διαχείριση πόρων στο cloud συνοδεύεται από προκλήσεις που οι διαχειριστές πρέπει να αντιμετωπίσουν.

Πολυπλοκότητα της Διαχείρισης

Η πολυπλοκότητα στη διαχείριση πολλών διαφορετικών περιβαλλόντων και η χρήση πολλαπλών εργαλείων παρακολούθησης μπορεί να αυξήσει την πιθανότητα σφαλμάτων.

Ασφάλεια και Συμμόρφωση

Η εξασφάλιση της συμμόρφωσης με κανονισμούς, όπως το GDPR, είναι απαραίτητη για την προστασία των προσωπικών δεδομένων των χρηστών.

-Data Encryption: Κρυπτογράφηση δεδομένων για την προστασία τους από μη εξουσιοδοτημένη πρόσβαση.

-Identity and Access Management (IAM): Χρήση πολιτικών πρόσβασης για τον έλεγχο ποιος έχει πρόσβαση σε δεδομένα και πόρους.

Πρόβλεψη και Σχεδιασμός Χωρητικότητας

Ο **σχεδιασμός χωρητικότητας** είναι απαραίτητος για την πρόβλεψη των αναγκών σε πόρους και την αποφυγή bottlenecks.

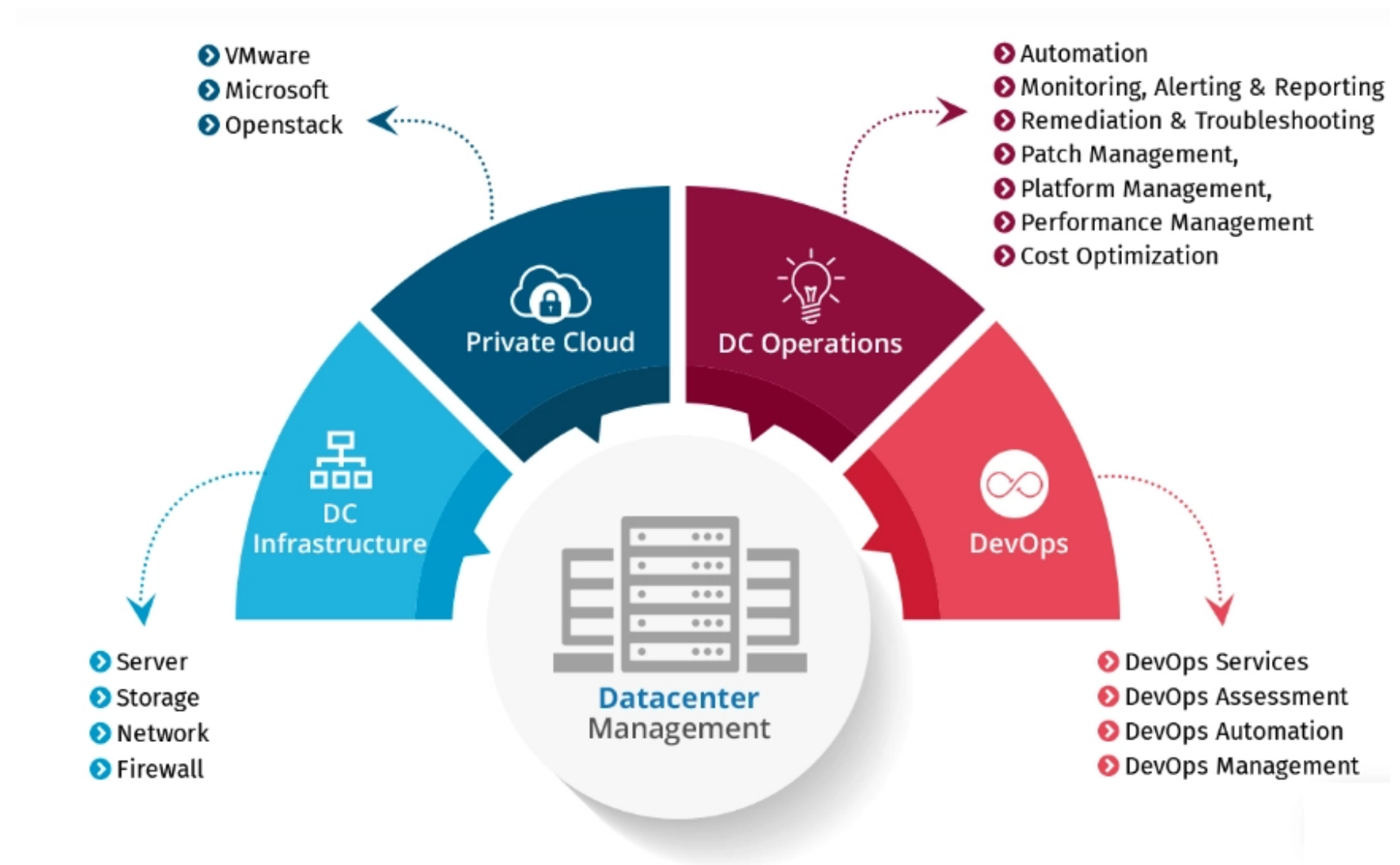
Capacity Planning: Ανάλυση των

τάσεων χρήσης πόρων για τη διασφάλιση ότι οι υποδομές μπορούν να ανταπεξέλθουν στην αυξανόμενη ζήτηση.

Συμπεράσματα

Η διαχείριση πόρων στο νέφος είναι ένα δυναμικό πεδίο που απαιτεί συνδυασμό

τεχνικών γνώσεων και στρατηγικής σκέψης. Οι σύγχρονες τεχνολογίες και τα εργαλεία που υπάρχουν επιτρέπουν την αποδοτική κατανομή και κλιμάκωση των πόρων, εξασφαλίζοντας την οικονομική βιωσιμότητα των υπηρεσιών στο cloud. Η σωστή κατανόηση και εφαρμογή των



Data Center Capacity PLanning

Διάλεξη 12

Κυβερνοασφάλεια στο Νέφος (Cloud Security)

Στόχοι:

- Κατανόηση των βασικών εννοιών της ασφάλειας στο cloud.
- Ανάλυση των απειλών και των προκλήσεων που συνοδεύουν την ασφάλεια στο νέφος.
- Παρουσίαση των τεχνικών προστασίας δεδομένων και των πρακτικών ασφάλειας.
- Εξέταση της συμμόρφωσης με κανονισμούς και της διαχείρισης της πρόσβασης.
- Ανάλυση των βέλτιστων πρακτικών για την αποτροπή επιθέσεων και την ανάκαμψη από παραβιάσεις.

Εισαγωγή στην Κυβερνοασφάλεια στο Cloud

Η κυβερνοασφάλεια στο νέφος

αναφέρεται στη διασφάλιση των δεδομένων και των υποδομών που φιλοξενούνται σε περιβάλλοντα cloud από απειλές και μη εξουσιοδοτημένη πρόσβαση. Η μετάβαση στο cloud φέρνει σημαντικά οφέλη, όπως την ευελιξία και την κλιμάκωση, αλλά επίσης φέρνει νέες προκλήσεις όσον αφορά την ασφάλεια.

Σημασία της Ασφάλειας στο Cloud

Η ασφαλής διαχείριση των δεδομένων και των εφαρμογών είναι κρίσιμη, καθώς οι πάροχοι cloud ενδέχεται να φιλοξενούν τα δεδομένα πολλών πελατών στον ίδιο φυσικό χώρο.

-Εμπιστευτικότητα:

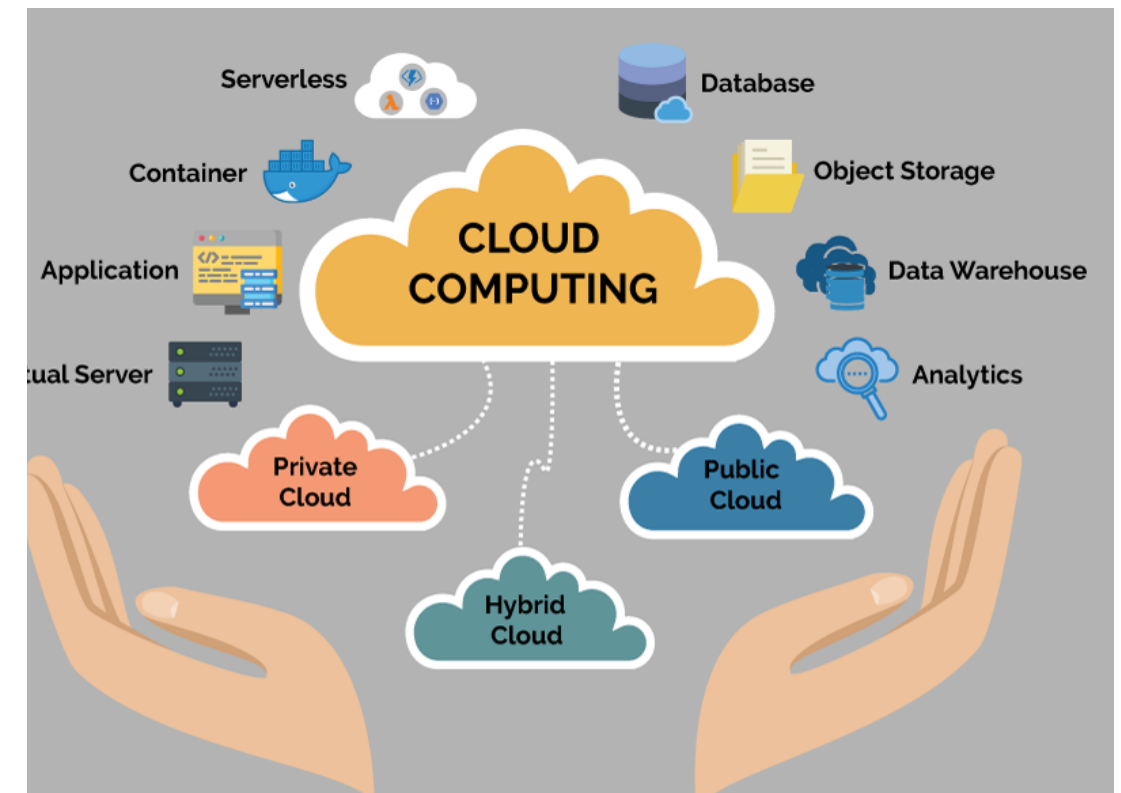
Διασφαλίζει ότι μόνο εξουσιοδοτημένα μέρη έχουν πρόσβαση στα δεδομένα.

-Ακεραιότητα:

Εξασφαλίζει ότι τα δεδομένα παραμένουν αμετάβλητα κατά τη μεταφορά ή την αποθήκευση.

-Διαθεσιμότητα:

Εξασφαλίζει ότι οι πόροι είναι διαθέσιμοι ανά πάσα στιγμή στους

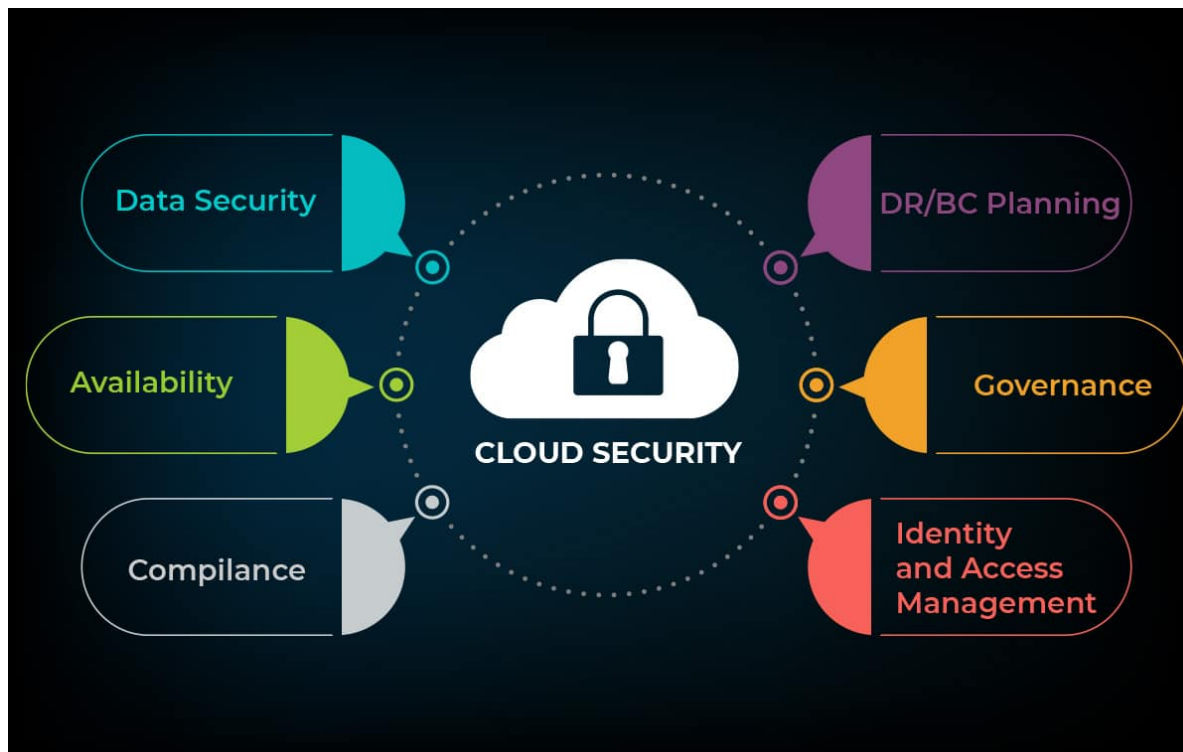


Cloud computing

εξουσιοδοτημένους χρήστες.

Προκλήσεις Ασφάλειας στο Cloud

Οι κύριες προκλήσεις περιλαμβάνουν τη διαχείριση της πρόσβασης, την προστασία των δεδομένων, την αποφυγή διαρροών, και τη συμμόρφωση με διεθνείς κανονισμούς.



Cloud Security

Κύριες Απειλές στο Cloud

Οι υποδομές cloud είναι ευάλωτες σε διάφορες μορφές επιθέσεων, οι οποίες μπορούν να επηρεάσουν την εμπιστευτικότητα και την ακεραιότητα των δεδομένων.

Επιθέσεις DDoS (Distributed Denial of Service)

Οι επιθέσεις DDoS στοχεύουν στην εξάντληση των πόρων του cloud, καθιστώντας τις υπηρεσίες μη διαθέσιμες στους χρήστες.

-Προστασία DDoS: Περιλαμβάνει τη χρήση υπηρεσιών από παρόχους cloud που φιλτράρουν την κυκλοφορία και εντοπίζουν ανώμαλα πρότυπα.

Επιθέσεις Man-in-the-Middle (MitM)

Στις επιθέσεις MitM, ο επιτιθέμενος παρεμβαίνει στη σύνδεση μεταξύ δύο μερών, αποκτώντας πρόσβαση σε δεδομένα.

-TLS/SSL: Η χρήση κρυπτογραφημένων συνδέσεων μπορεί να αποτρέψει τέτοιες επιθέσεις, διασφαλίζοντας ότι τα δεδομένα παραμένουν προστατευμένα κατά τη μεταφορά.

Ransomware στο Cloud

Το ransomware μπορεί να στοχεύσει τα δεδομένα που αποθηκεύονται στο cloud,

κρυπτογραφώντας τα και ζητώντας λύτρα για την επαναφορά τους.

-Backups και Snapshots: Η τακτική δημιουργία αντιγράφων ασφαλείας μπορεί να μειώσει τον αντίκτυπο αυτών των επιθέσεων.

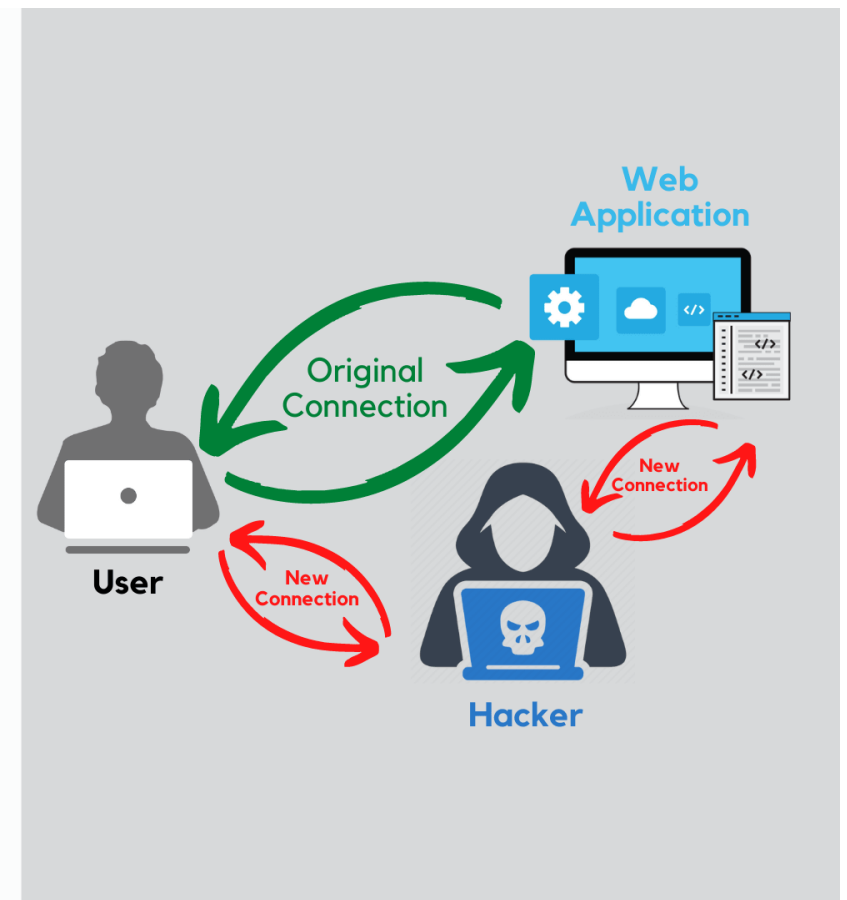
Διαρροές Δεδομένων (Data Breaches)

Οι διαρροές δεδομένων συμβαίνουν όταν μη εξουσιοδοτημένοι χρήστες αποκτούν πρόσβαση σε ευαίσθητες πληροφορίες.

-Encryption at Rest: Κρυπτογράφηση των δεδομένων ακόμα και όταν δεν είναι σε χρήση, για να διασφαλίζεται η εμπιστευτικότητα.

TYPES OF MAN IN THE MIDDLE ATTACKS

- Wi-Fi Eavesdropping
- DNS Spoofing
- Session Hijacking
- ARP Cache Poisoning
- Man in the Browser
- IP Spoofing
- Secure Sockets Layer (SSL) Hijacking
- Email Hijacking



Τεχνικές Προστασίας Δεδομένων

Η προστασία των δεδομένων στο cloud είναι κρίσιμη για την αποτροπή διαρροών και την προστασία των χρηστών.

Κρυπτογράφηση Δεδομένων

-Encryption in Transit: Κρυπτογράφηση των δεδομένων κατά τη μεταφορά μέσω TLS/SSL.

-Encryption at Rest: Κρυπτογράφηση των δεδομένων όταν αποθηκεύονται σε δίσκους ή βάσεις δεδομένων στο cloud.

Διαχείριση Ταυτοτήτων και Πρόσβασης (IAM)

Το IAM διασφαλίζει ότι μόνο εξουσιοδοτημένοι χρήστες έχουν πρόσβαση στους πόρους.

-Role-Based Access Control (RBAC):

Ορισμός ρόλων για την απλοποίηση της διαχείρισης πρόσβασης.

-Multi-Factor Authentication (MFA):

Προσθήκη ενός δεύτερου επιπέδου αυθεντικοποίησης για αυξημένη ασφάλεια.

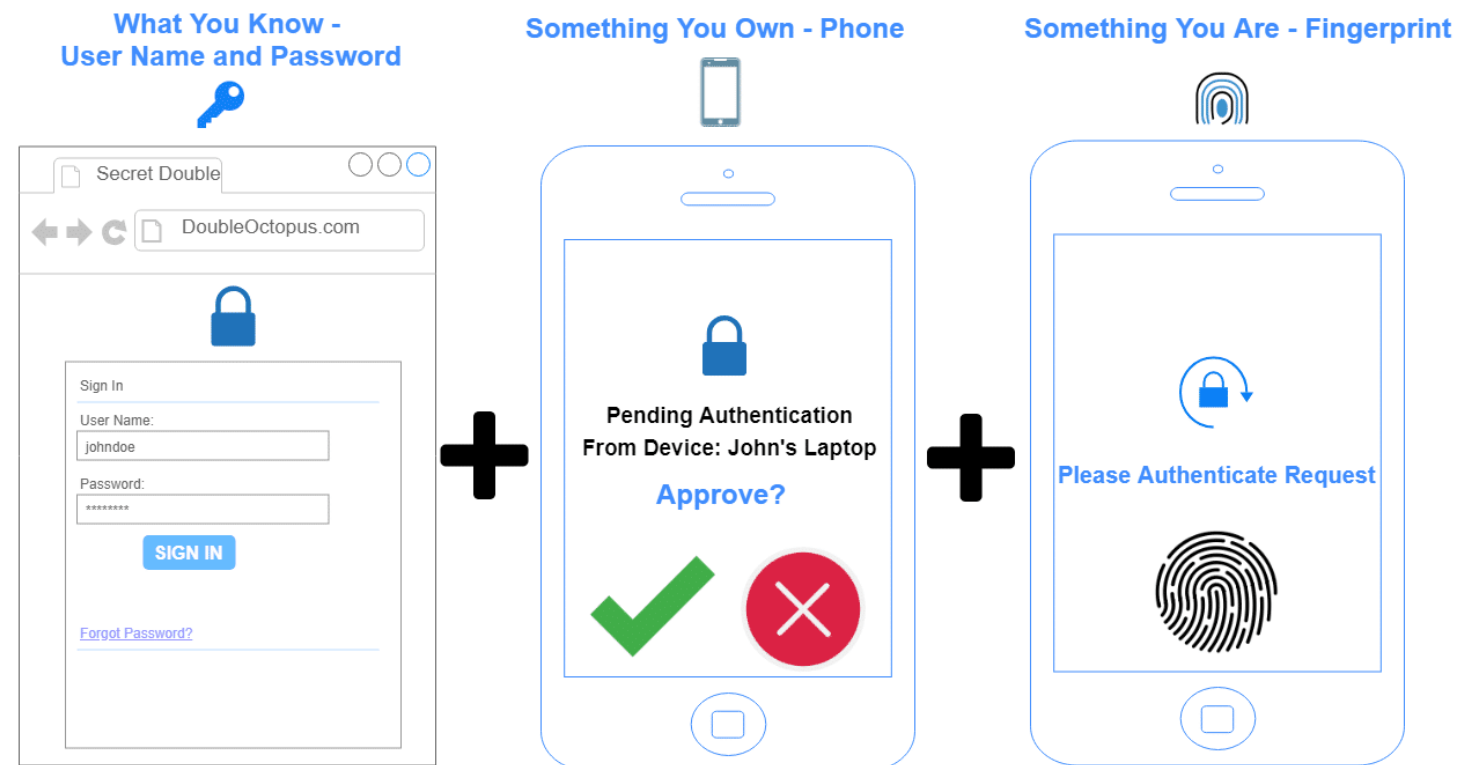
Ασφάλεια των API

Τα API είναι ευάλωτα σε επιθέσεις και πρέπει να προστατεύονται κατάλληλα.

-OAuth 2.0 και OpenID Connect:

Πρότυπα για τη διαχείριση της πρόσβασης στα API.

Example of Multi Factor Authentication



Multi Factor Authentication

-Rate Limiting: Περιορισμός του αριθμού αιτημάτων για αποτροπή κακόβουλων επιθέσεων.

Βέλτιστες Πρακτικές Ασφάλειας στο Cloud

Οι βέλτιστες πρακτικές βοηθούν στην ενίσχυση της ασφάλειας και στην αποτροπή παραβιάσεων.

Intrusion Detection Systems (IDS) και SIEM

-Intrusion Detection Systems (IDS):

Παρακολουθούν την κυκλοφορία και εντοπίζουν ύποπτες δραστηριότητες.

-Security Information and Event Management (SIEM): Συγκεντρώνει δεδομένα ασφαλείας και παρέχει πληροφορίες για ανάλυση απειλών.

Πρακτικές Ανάκαμψης από Παραβιάσεις

-Disaster Recovery Plans (DRP): Σχέδια για την ανάκαμψη από καταστροφικές επιθέσεις.

-Incident Response: Καθορίζει τα βήματα που πρέπει να ακολουθηθούν σε περίπτωση παραβίασης.

Ασφάλεια σε Περιβάλλον Multi-Cloud

Η χρήση πολλών παρόχων cloud προσφέρει πλεονεκτήματα, αλλά απαιτεί σωστή διαχείριση της ασφάλειας.

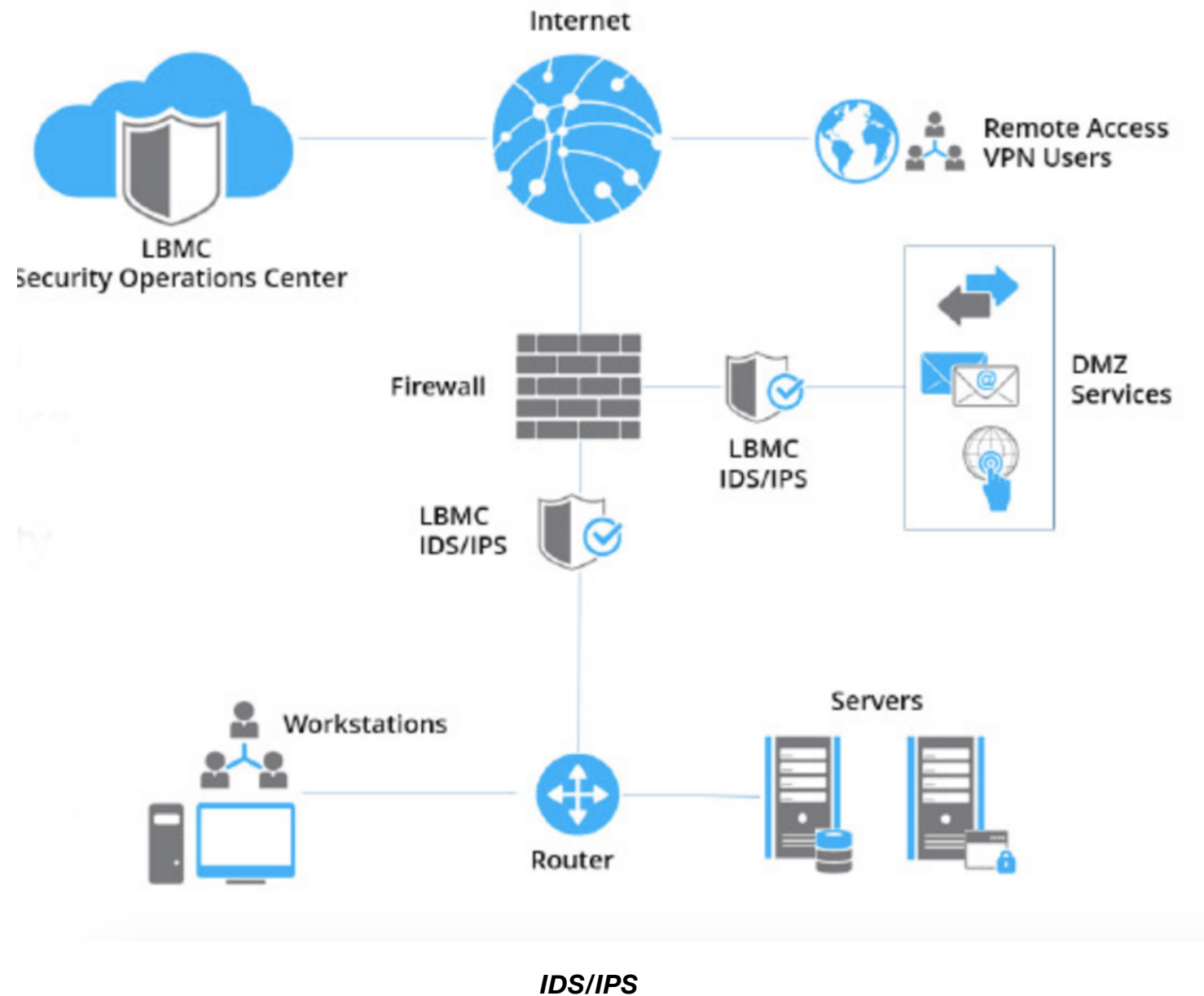
-Cross-Cloud Identity Management:

Ενιαία διαχείριση ταυτοτήτων και πρόσβασης σε πολλαπλές πλατφόρμες.

-Data Encryption Across Clouds:

Κρυπτογράφηση δεδομένων κατά τη

μεταφορά τους μεταξύ διαφορετικών παρόχων.



Προκλήσεις και Μέλλον της Ασφάλειας στο Cloud

Η κυβερνοασφάλεια στο cloud αντιμετωπίζει νέες προκλήσεις, όπως η προστασία των συσκευών IoT και η διαχείριση των επιθέσεων zero-day.

Προστασία Συστημάτων IoT

Τα **συστήματα IoT** συνδέονται με το cloud για την επεξεργασία δεδομένων και είναι ιδιαίτερα ευάλωτα σε επιθέσεις.

-Edge Computing: Μεταφέρει την επεξεργασία κοντά στις συσκευές για να μειωθεί η καθυστέρηση και να αυξηθεί η ασφάλεια.

Αντιμετώπιση Επιθέσεων Zero-Day

Οι επιθέσεις zero-day εκμεταλλεύονται άγνωστες ευπάθειες και είναι ιδιαίτερα επικίνδυνες.

-Threat Intelligence: Χρήση δεδομένων για την πρόβλεψη και αποτροπή νέων επιθέσεων.

Τάσεις στο Cloud Security

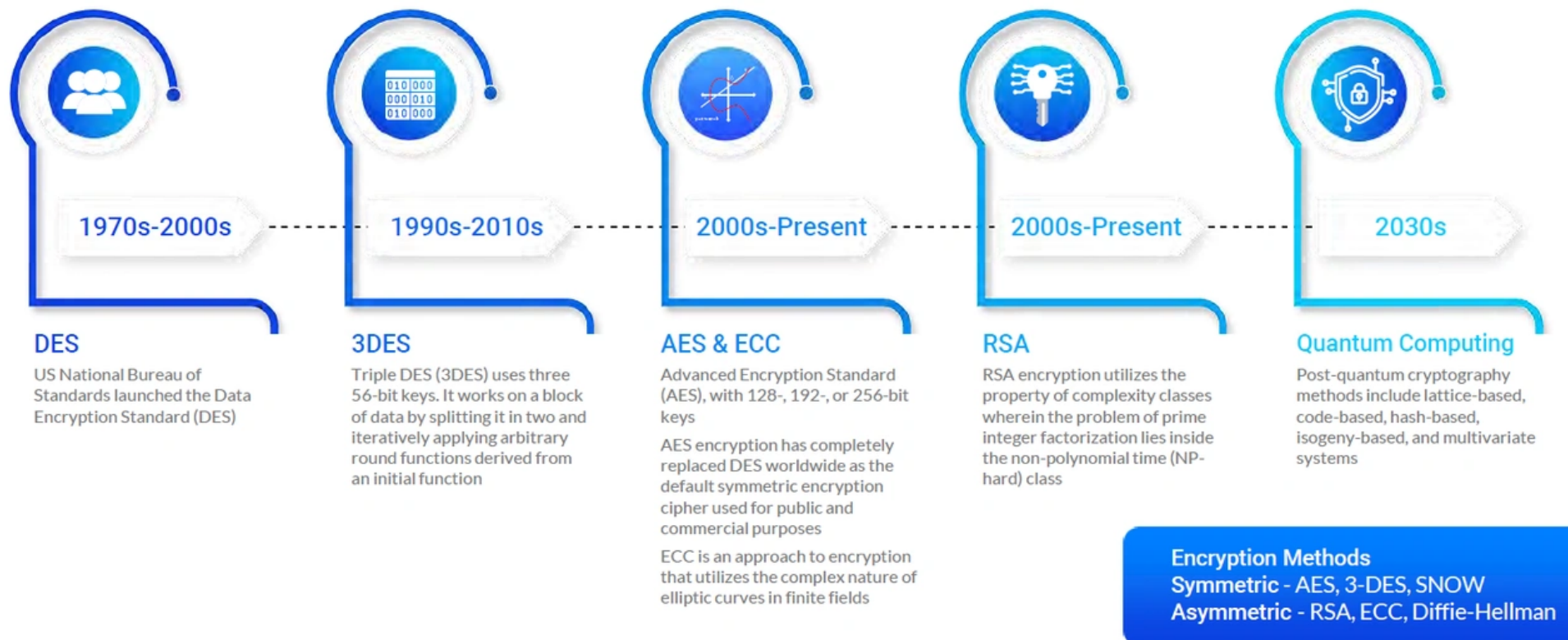
Η ανάπτυξη νέων τεχνολογιών, όπως το quantum computing, μπορεί να φέρει αλλαγές στην ασφάλεια κρυπτογράφησης.

-Post-Quantum Encryption: Προετοιμασία για την αντιμετώπιση των προκλήσεων που θα φέρει η κβαντική υπολογιστική.

Συμπερασματικά:

Η **κυβερνοασφάλεια** στο νέφος είναι ένα **πολυσύνθετο πεδίο** που απαιτεί τη συνεχή παρακολούθηση και την εφαρμογή των καλύτερων πρακτικών για τη διασφάλιση της εμπιστευτικότητας και της ακεραιότητας των δεδομένων. Η προστασία των συστημάτων από κακόβουλες επιθέσεις και η συμμόρφωση με τους κανονισμούς είναι κρίσιμη για την

επιτυχία των οργανισμών που χρησιμοποιούν το cloud.



Quantum Computing

Διάλεξη 13

Τάσεις και Μέλλον των Λειτουργικών Συστημάτων

Στόχοι:

- Κατανόηση των σύγχρονων τάσεων στα λειτουργικά συστήματα.
- Ανάλυση των επιπτώσεων των νέων τεχνολογιών όπως το cloud και το IoT.
- Εξέταση της ασφάλειας και της αυτοματοποίησης στα λειτουργικά συστήματα.
- Παρουσίαση των προκλήσεων και των προοπτικών για το μέλλον, συμπεριλαμβανομένου του quantum computing.

Σύγχρονες Τάσεις στα Λειτουργικά Συστήματα

Τα λειτουργικά συστήματα συνεχίζουν να εξελίσσονται παράλληλα με τις τεχνολογικές τάσεις, προσαρμόζοντας τις δυνατότητές τους για να καλύψουν τις νέες ανάγκες των χρηστών και των επιχειρήσεων. Από την εποχή των μονολιθικών λειτουργικών συστημάτων μέχρι την υιοθέτηση του cloud computing και των συστημάτων εικονικοποίησης, οι καινοτομίες δεν σταματούν να επηρεάζουν τον σχεδιασμό και την αρχιτεκτονική των λειτουργικών συστημάτων.

Ενοποίηση με το Cloud Computing

Το cloud computing έχει επηρεάσει σημαντικά τον σχεδιασμό των λειτουργικών συστημάτων, οδηγώντας στη βελτιστοποίηση της απόδοσης και της κλιμάκωσης.

-Serverless Computing: Οι προγραμματιστές μπορούν να εκτελούν κώδικα χωρίς να διαχειρίζονται διακομιστές, χάρη στις cloud πλατφόρμες που αναλαμβάνουν την αυτόματη κατανομή πόρων.

-Containerization: Οι πλατφόρμες όπως το Docker και το Kubernetes επιτρέπουν την ανάπτυξη ελαφρών containers,

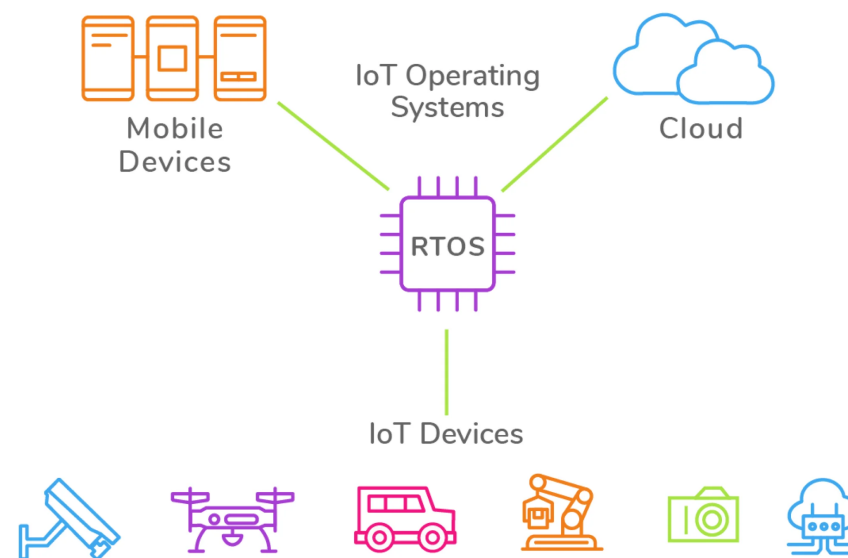
που λειτουργούν ανεξάρτητα από το λειτουργικό σύστημα του κεντρικού server.

Ανάπτυξη στα Συστήματα IoT

Τα λειτουργικά συστήματα σχεδιάζονται πλέον με στόχο την υποστήριξη του Internet of Things (IoT), παρέχοντας συμβατότητα και αποδοτική διαχείριση για συσκευές με περιορισμένους πόρους.

-Real-Time Operating Systems (RTOS):

Λειτουργικά συστήματα που εξυπηρετούν την ανάγκη των IoT συσκευών για ταχεία επεξεργασία και απόκριση.



Real Time OS - Internet of Things

-Edge Computing: Η μεταφορά της επεξεργασίας δεδομένων κοντά στις IoT συσκευές μειώνει την καθυστέρηση και βελτιώνει την απόκριση, απαιτώντας την προσαρμογή των λειτουργικών συστημάτων για τη διαχείριση αυτών των διαδικασιών.

Ασφάλεια και Αυτοματοποίηση

Η ασφάλεια αποτελεί κρίσιμη πτυχή για τα σύγχρονα λειτουργικά συστήματα, ειδικά με τις αυξανόμενες απειλές από κακόβουλο λογισμικό και επιθέσεις.

-Zero Trust Security:

Η υιοθέτηση μοντέλων όπου κάθε πρόσβαση θεωρείται επικίνδυνη μέχρι να αποδειχθεί το αντίθετο.

-Automation in Security

Updates: Τα λειτουργικά συστήματα, όπως το Windows και οι διανομές Linux, ενσωματώνουν αυτοματοποιημένα συστήματα ενημερώσεων για να μειώσουν τις ευπάθειες και να διατηρούν τα συστήματα ασφαλή.

Καινοτομίες και Τεχνολογικές Εξελίξεις

Αναβάθμιση της Εικονικοποίησης

Η εικονικοποίηση εξακολουθεί να παίζει κεντρικό ρόλο στον σχεδιασμό των λειτουργικών συστημάτων, αλλά εξελίσσεται με νέες μορφές.

-MicroVMs: Ελαφρύτερες εκδόσεις VMs, όπως τα Firecracker της AWS, προσφέρουν ταχύτητα και αποδοτικότητα, καθιστώντας τις ιδανικές για serverless εφαρμογές.

-Containers vs. VMs: Η συζήτηση μεταξύ containers και VMs συνεχίζει να επηρεάζει τον τρόπο με τον οποίο οι επιχειρήσεις υιοθετούν και διαχειρίζονται τις εφαρμογές τους.

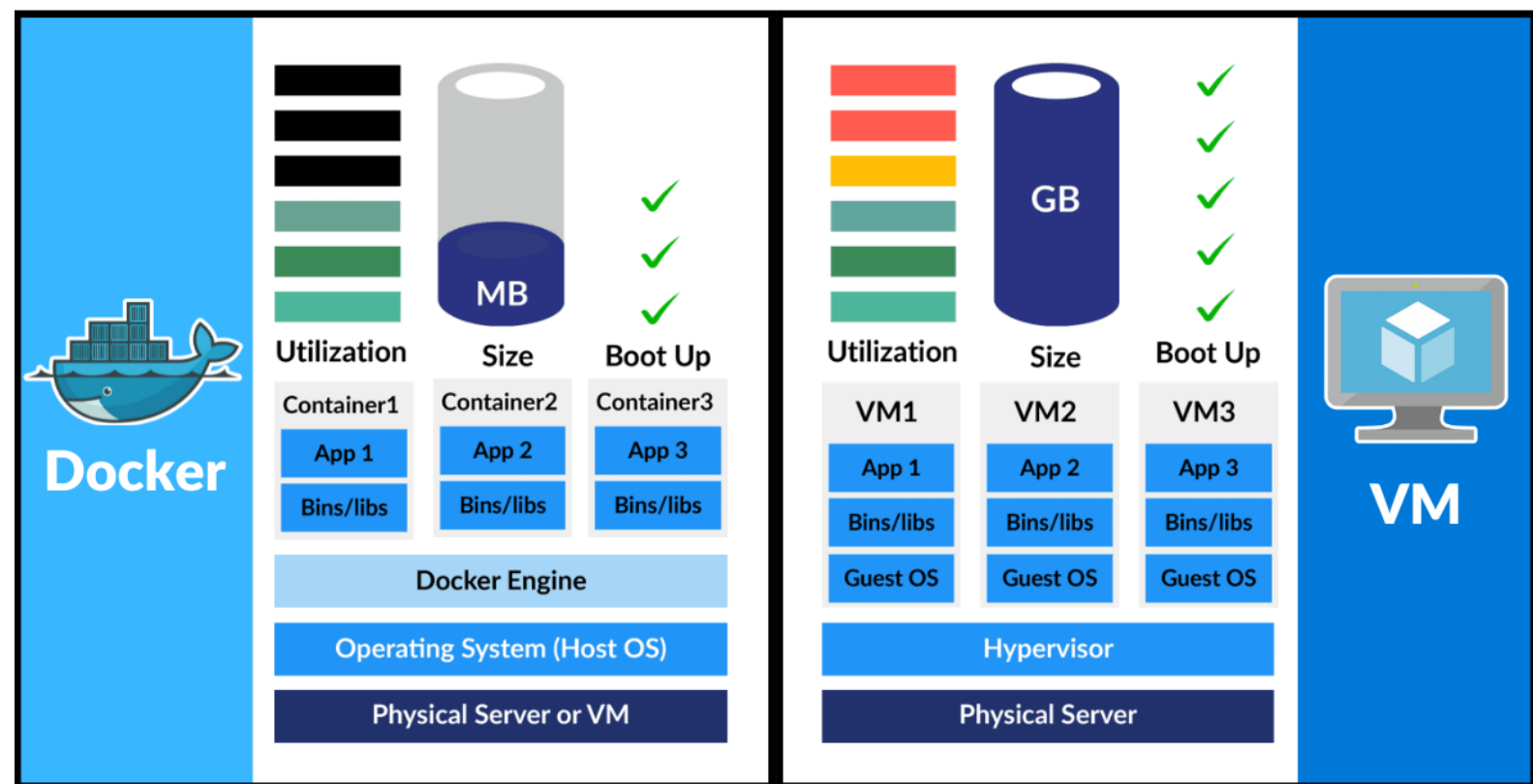
Artificial Intelligence και Machine Learning

Η ενσωμάτωση της τεχνητής νοημοσύνης (AI) και της μηχανικής μάθησης (ML) στις λειτουργίες των λειτουργικών συστημάτων βοηθά στην αυτοματοποίηση διαδικασιών, όπως η ανίχνευση απειλών και η διαχείριση πόρων.

-Predictive Maintenance: Χρήση AI για την πρόβλεψη πιθανών αποτυχιών σε επίπεδο hardware και την αποτροπή τους μέσω αυτόματων ενημερώσεων και επιδιορθώσεων.

-Enhanced User Experience:

Προσαρμογή της διεπαφής χρήστη με βάση τις προτιμήσεις και τις συνήθειες μέσω ανάλυσης δεδομένων χρήσης.



Containers vs VMs

Προοπτικές για το Μέλλον των Λειτουργικών Συστημάτων

Οι προκλήσεις είναι η ενοποίηση, τα κβάντα και η αύξηση της ασφάλειας.

Ενοποίηση Λειτουργικών Συστημάτων και Πλατφορμών

Η τάση για ενοποίηση των λειτουργικών συστημάτων προσπαθεί να γεφυρώσει το χάσμα μεταξύ **desktop**, **mobile**, και **IoT** πλατφορμών.

-Universal OS (π.χ., Fuchsia): Το Fuchsia της Google επιχειρεί να δημιουργήσει ένα ενιαίο λειτουργικό σύστημα για όλες τις πλατφόρμες, προσφέροντας ομοιομορφία και καλύτερη ενσωμάτωση.

-Cross-Platform Development: Εργαλεία όπως το Flutter επιτρέπουν στους προγραμματιστές να δημιουργούν εφαρμογές που λειτουργούν σε πολλαπλές πλατφόρμες με ένα κοινό κώδικα.

Quantum Computing και Λειτουργικά Συστήματα

Η έλευση της κβαντικής υπολογιστικής φέρνει νέες προκλήσεις και ευκαιρίες για τα λειτουργικά συστήματα.

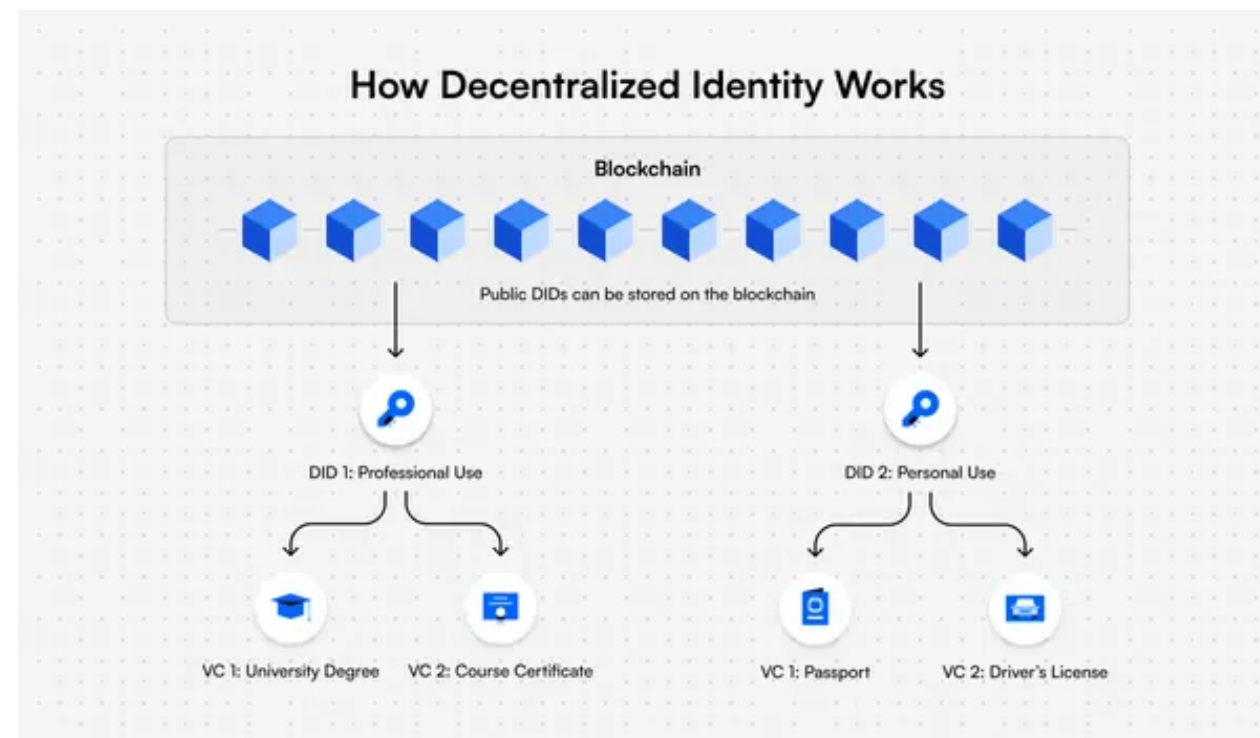
-Quantum Operating Systems (QOS): Η ανάγκη για λειτουργικά συστήματα που μπορούν να διαχειριστούν κβαντικούς επεξεργαστές είναι μια νέα περιοχή έρευνας.

-Hybrid Computing Models: Η ενσωμάτωση κλασικών και κβαντικών υπολογισμών σε ένα ενιαίο περιβάλλον θα απαιτήσει προσαρμογές στα υπάρχοντα λειτουργικά συστήματα.

Αύξηση της Ασφάλειας με τη Χρήση Blockchain

-Decentralized Authentication: Τα λειτουργικά συστήματα μπορούν να αξιοποιήσουν την κατακεταμμένη φύση του blockchain για την ενίσχυση της ασφάλειας ταυτοποίησης και την αποφυγή κεντρικών σημείων ευπάθειας.

-Data Integrity: Το blockchain μπορεί να



Decentralized Identity Authentication

Η χρήση του blockchain και άλλων τεχνολογιών κατακεταμμένης λογιστικής μπορεί να ενισχύσει την ασφάλεια των λειτουργικών συστημάτων.

χρησιμοποιηθεί για την προστασία της ακεραιότητας των δεδομένων, διασφαλίζοντας ότι τα δεδομένα παραμένουν αμετάβλητα.

Αναφορές λειτουργικών συστημάτων

A. Βασικότερες αναφορές

Silberschatz, Galvin, Gagne - Operating System Concepts

Remzi και Andrea Arpaci-Dusseau - Operating Systems: Three Easy Pieces

Tanenbaum και Bos - Modern Operating Systems

B. Το UEFI

<https://www.spiceworks.com/tech/tech-general/articles/what-is-unified-extensible-firmware-interface/>

<https://en.wikipedia.org/wiki/UEFI>

C. Λειτουργικά συστήματα σήμερα

<http://www.aclals.org/what-are-operating-systems/>

https://en.wikipedia.org/wiki/Comparison_of_operating_systems

D. Διαχείριση διεργασιών

<https://www.enjoyalgorithms.com/blog/process-management-in-operating-system>

<https://datatrained.com/post/process-management-os/>

E. Διαχείριση μνήμης

<https://phoenixnap.com/kb/memory-management>

[https://en.wikipedia.org/wiki/Memory_management_\(operating_systems\)](https://en.wikipedia.org/wiki/Memory_management_(operating_systems))

https://en.wikipedia.org/wiki/Virtual_memory

F. Βιβλία

Silberschatz, Galvin, Gagne - Operating System Concepts.

Remzi και Andrea Arpaci-Dusseau - Operating Systems: Three Easy Pieces.

Tanenbaum και Bos - Modern Operating Systems.