



## Διάλεξη 1

# Εισαγωγή

Δημήτρης Ζεϊναλιπούρ

<https://www.cs.ucy.ac.cy/~dzeina>



# Επαναληπτικά

- **Λογισμικό Εφαρμογών (Application Software):**
  - Περιλαμβάνει προγράμματα που επιτρέπουν την **εκτέλεση συγκεκριμένων εργασιών** των χρηστών π.χ την δημιουργία κειμένων, την σχεδίαση γραφικών, την οργάνωση δεδομένων κλπ.
- **Λογισμικό Συστήματος (System Software):**
  - Περιλαμβάνει το σύνολο των προγραμμάτων που ελέγχουν διαχειρίζονται και **συντονίζουν** τους **πόρους των Η/Υ**.
  - Λειτουργεί σε **καθεστώς ανεξαρτησίας** από συγκεκριμένες εφαρμογές.
  - Η λειτουργία του **ΔΕΝ** είναι **άμεσα αντιληπτή** από τον απλό χρήστη.
- **Προγραμματισμός Συστημάτων;**
  - Η διαδικασία συγγραφής λογισμικού συστήματος.



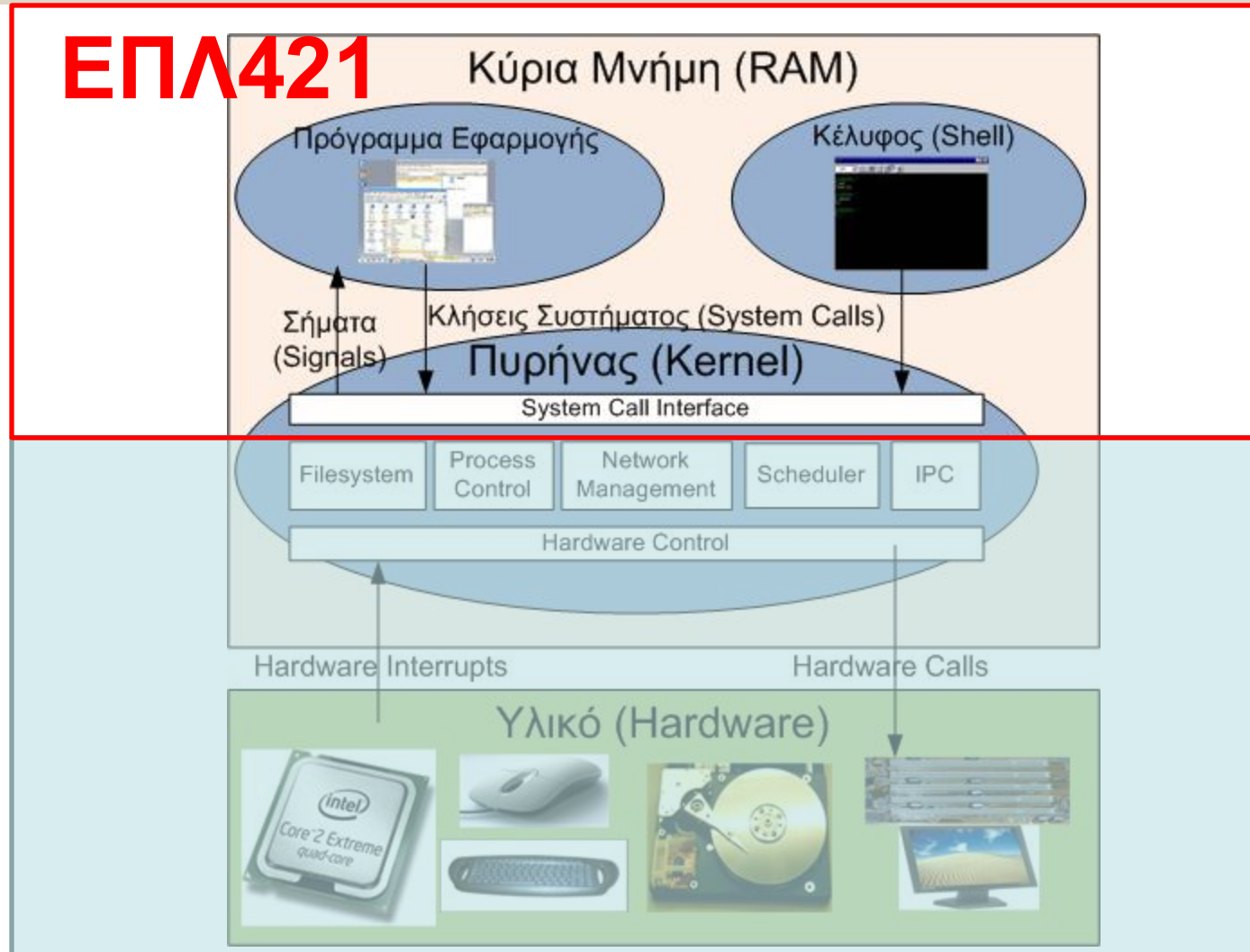
# Τι καλύπτει το ΕΠΛ421;

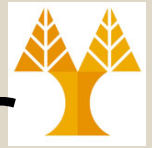
## Βασικές και Προχωρημένες έννοιες Προγραμματισμού Συστημάτων

**Στόχος:** Να κατανοήσουν οι φοιτητές/τριες τους μηχανισμούς με τους οποίους επιτυγχάνεται η πρόσβαση στις ρουτίνες (βιβλιοθήκες / μονάδες) ενός Λειτουργικού Συστήματος (UNIX) με τη χρήση της γλώσσας προγραμματισμού C, δημοφιλών κελυφών και εργαλείων ωφελιμότητας.



# Τι καλύπτει το ΕΠΛ421;





# Αναλυτικοί Στόχοι Μαθήματος

- **Εκμάθηση βασικών και προχωρημένων τεχνικών διαχείρισης και προγραμματισμού στο Λειτουργικό Σύστημα Unix (Shell commands and Bash Shell programming).**
- **Εκμάθηση προχωρημένων τεχνικών προγραμματισμού στην C & Unix:**
  1. Συστήματα Αρχείων και Διαχείριση Μνήμης;
  2. Προγραμματισμός Διεργασιών (Processes – Πολυδιεργασιακός Προγραμματισμός) και Νημάτων (Threads – Πολυνηματικός Προγραμματισμός);
  3. Επικοινωνία Διεργασιών (Διαδιεργασιακή Επικοινωνία);
  4. Προγραμματισμός Δικτύου (Socket programming);
  5. Ασφάλεια Λογισμικού Συστημάτων (security)



# Συμβόλαιο Μαθήματος

- **Επίπεδο:** Προπτυχιακό
  - Περιορισμένη Επιλογή
- **Πίστωση:** 7.5 μονάδες ECTS
- **Προαπαιτούμενα:**
  - ΕΠΛ222: Λειτουργικά Συστήματα
    - ΕΠΛ232: Προγραμματιστικές Τεχνικές και Εργαλεία
- **Μέθοδοι Διδασκαλίας**
  - Διαλέξεις (3 ώρες εβδομαδιαίως)
  - Φροντιστήριο (1 ώρα εβδομαδιαίως)
  - Εργαστήριο (1 1/2 ώρα εβδομαδιαίως)



# Συμβόλαιο Μαθήματος

- **Αξιολόγηση**

- 50% Τελική Εξέταση

- 25% Ενδιάμεση Εξέταση-

- **Παρασκευή, 25 Οκτωβρίου 2024 (8<sup>η</sup> Εβδ.)**

- 25% Ασκήσεις ή/και Quizzes

- 10% **Προγραμματιστικές Ασκήσεις**

- 10% Project – **Υλοποίηση Συστήματος**

- 5% Παρουσίαση κάποιας **γλώσσας** ή **βιβλιοθήκης**, με έμφαση στο προγραμματισμό συστημάτων (π.χ., Systems programming in windows, scripting languages, emerging systems in mobile, clouds, etc.).

# Αναγνώριση Εξειδίκευσης



<https://www.cs.ucy.ac.cy/docs/prospectus.pdf>

- 4/7 Κατευθύνσεις!
  - Δικτύων Υπολογιστών
  - Μεγάλα Δεδομένα και Διαδικτυακός Υπολογισμός
  - Τεχνολογία Λογισμικού
  - Ψηφιακά και Ενσωματωμένα Συστήματα

## Μαθήματα Περιορισμένης Επιλογής Εξειδίκευσης Μεγάλα Δεδομένα και Διαδικτυακός Υπολογισμός\*

ΕΠΑ344	Τεχνολογίες Διαδικτύου
ΕΠΑ421	Προγραμματισμός Συστημάτων
ΕΠΑ446	Προχωρημένες Βάσεις Δεδομένων
ΕΠΑ448	Εξόρυξη Δεδομένων στον Παγκόσμιο Ιστό
ΕΠΑ450	Εικονικοποίηση και Διαχείριση Δικτύου
ΕΠΑ481	Τεχνολογία Λογισμικού για Λογισμικό ως Υπηρεσία
ΜΑΣ458	Στατιστική Ανάλυση Δεδομένων

## Μαθήματα Περιορισμένης Επιλογής Εξειδίκευσης Ψηφιακά και Ενσωματωμένα Συστήματα

ΕΠΑ420	Αρχιτεκτονική Υπολογιστών
ΕΠΑ421	Προγραμματισμός Συστημάτων
ΕΠΑ428	Διαδίκτυο των Πραγμάτων: Προγραμματισμός και Εφαρμογές
ΕΠΑ429	Θεωρία και Πρακτική Μεταγλωττιστών
ΕΠΑ445	Ψηφιακή Επεξεργασία Εικόνων
ΕΠΑ481	Τεχνολογία Λογισμικού για Λογισμικό ως Υπηρεσία

## Μαθήματα Περιορισμένης Επιλογής Εξειδίκευσης Τεχνολογία Λογισμικού

ΕΠΑ344	Τεχνολογίες Διαδικτύου
ΕΠΑ421	Προγραμματισμός Συστημάτων
ΕΠΑ435	Αλληλεπίδραση Ανθρώπου Υπολογιστή
ΕΠΑ441	Προχωρημένη Τεχνολογία Λογισμικού
ΕΠΑ443	Επαναχρησιμοποίηση Λογισμικού
ΕΠΑ449	Επαγγελματική Πρακτική Τεχνολογίας Λογισμικού
ΕΠΑ451	Ανάλυση Λογισμικού
ΕΠΑ481	Τεχνολογία Λογισμικού για Λογισμικό ως Υπηρεσία

## Μαθήματα Περιορισμένης Επιλογής Εξειδίκευσης Δικτύων Υπολογιστών

ΕΠΑ344	Τεχνολογίες Διαδικτύου
ΕΠΑ421	Προγραμματισμός Συστημάτων
ΕΠΑ422	Προχωρημένα Δίκτυα
ΕΠΑ423	Ασφάλεια Δικτύων και Πληροφοριών
ΕΠΑ427	Κινητά Δίκτυα Υπολογιστών
ΕΠΑ428	Διαδίκτυο των Πραγμάτων: Προγραμματισμός και Εφαρμογές
ΕΠΑ432	Κατανεμημένοι Αλγόριθμοι
ΕΠΑ450	Εικονικοποίηση και Διαχείριση Δικτύου

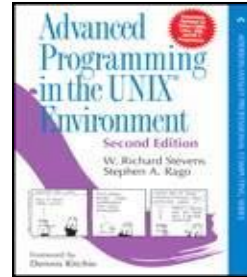


# Βιβλιογραφία



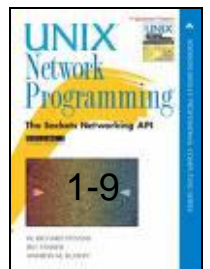
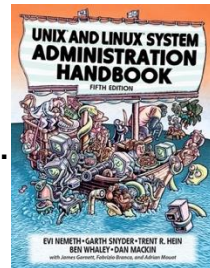
## Main Bibliography

- **Stevens & Rago** [Advanced Programming in the UNIX® Environment](#), Third Edition, Richard Stevens, Stephen A. Rago, Addison-Wesley, ISBN-10: 0321637739 | ISBN-13: 978-032163773, **2013**.
- **Das** [Your UNIX/Linux: The Ultimate Guide](#), 3rd Edition, Sumitabha Das, McGraw Hill, ISBN-13 9780073376202, 800 page, **2013**.
- **Das** [UNIX and Linux System Administration Handbook](#) 5th Edition, Evi Nemeth, Garth Snyder, Trent R. Hein, Ben Whaley, Dan Mackin, 1232 pages, Addison-Wesley, ISBN-10: 0134277554 | ISBN-13: 978-01342775545, **2017**.



## Supplementary Bibliography

- **CS:APP3e** [Computer Systems: A Programmer's Perspective, 3/E \(CS:APP3e\)](#), Third Edition, Randal E. Bryant and David R. O'Hallaron, Carnegie Mellon University, ISBN-13: 978-0134092669 | ISBN-10: 013409266X, 1128 pages, **2015**.
- **Robbins & Robbins** [UNIX Systems Programming: Communication, Concurrency and Threads](#), 2nd Edition, Kay Robbins and Steve Robbins, Prentice Hall, ISBN-10: 0134424077, ISBN-13: 978-0134424071, **2015**.
- **Stevens, Fenner & Rudoff** [Unix Network Programming, Volume 1: The Sockets Networking API](#), 3rd Edition, W. Richard Stevens, Bill Fenner, Andrew M. Rudoff, Addison Wesley Professional, ISBN-10:0131411551 | ISBN-13:9780131411555, **2004**.





# WWW

- Όλες οι πληροφορίες σχετικά με το μάθημα βρίσκονται στο ακόλουθο URL

<http://www.cs.ucy.ac.cy/courses/EPL421>

The screenshot shows the course page for EPL421: Systems Programming. The page header includes the University of Cyprus logo and navigation links: EPL421, News, Schedule, Labs, Assignments, Links, Presentations, Moodle, TA, and a search icon. The breadcrumb trail is 'Zeinalipour > Courses > EPL421'. The course title is 'EPL421: Systems Programming'. The instructor is Demetris Zeinalipour. The course type is 'Upper-division Undergraduate (Elective)'. The prerequisite is 'EPL222: Operating Systems'. The course is held on Tuesdays and Fridays from 16:30 to 18:00 in room XΩΔ01-#101. Recitations are on Wednesdays from 15:00 to 16:00 in room XΩΔ01-#004. The laboratory is on Fridays from 18:00 to 20:00 in room ΘΕΕ01-B103. The assistant is Pavlos Antoniou. An 'Overview' section follows, describing the course content: 'In this course, students will learn to develop complex system-level software in the C programming language while gaining an intimate understanding of the UNIX operating system (and all OS that belong to this family, such as Linux, the BSDs, and even Mac OS X) and its programming environment. Topics covered will include the user/kernel interface, fundamental concepts of UNIX, user authentication, basic and advanced I/O, filesystems, signals, process relationships, and interprocess communication. Fundamental concepts of software development and maintenance on UNIX systems will also be covered. The students are expected to have a good working knowledge of the C programming language (EPL232) and a good working knowledge of fundamental Operating System Concepts (EPL221).'



# WWW

- Για τις εκπαιδευτικές δραστηριότητες του μαθήματος (υποβολή εργασιών, φόρουμ ανακοινώσεων, ερωτηματολόγια, βαθμολογίες εργασιών, κτλ) θα χρησιμοποιηθεί το Moodle. <http://moodle.cs.ucy.ac.cy/>

The screenshot displays the Moodle interface for the course 'EPL421 - Systems Programming'. The top navigation bar includes the Moodle logo, the course name, and the user's name 'Demetris Zeinalipour'. The main content area is divided into three columns:

- Left Column (NAVIGATION):** A sidebar menu with options like Home, Dashboard, Site pages, My courses, and a detailed list of topics and sub-topics under EPL421.
- Middle Column (General):** The main course information, including the title 'EPL421: Systems Programming', the instructor's name 'Demetris Zeinalipour', and a link to the course page. It also features 'Important Notices' and links to 'Announcements' and 'Discussion Forum'.
- Right Column:** A sidebar with 'LATEST ANNOUNCEMENTS', a 'SEARCH FORUMS' box, and a 'CS COLLOQUIUM SERIES @ UCY' section featuring the UCY logo and details of a colloquium.



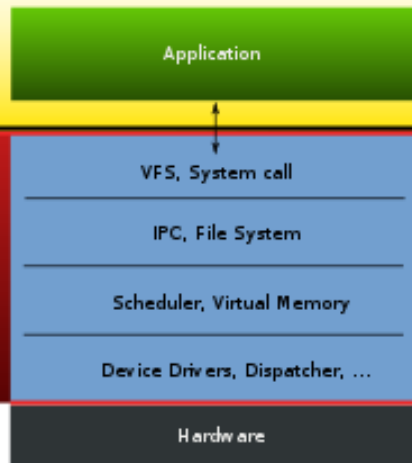
# Ανατομία του ΛΣ

- Γνωρίζουμε ότι ένα **Λειτουργικό Σύστημα (ΛΣ)** συντονίζει τους πόρους ενός Η.Υ. (υλικό, λογισμικό, διαχείριση χρηστών, κτλ.).
- Το ΛΣ αποτελείται τα εξής συστατικά:
  1. **Kernel (Πυρήνας)**: διεξάγει τις πιο σημαντικές λειτουργίες για την εύρυθμη λειτουργία του ΗΥ (χρονοδρομολόγηση διεργασιών, διαχείριση μνήμης, Δια-διεργασιακή επικοινωνία, κτλ.)
  2. **System Commands, Utilities and Libraries** (π.χ., ls, sed, awk, pthread.h,unistd.h, κτλ.) : Προγράμματα και βιβλιοθήκες τα οποία επιτρέπουν στα προγράμματα **εφαρμογών (user space)** να έχουν πρόσβαση στις ρουτίνες του **Πυρήνα (kernel space)** του Λειτουργικού Συστήματος



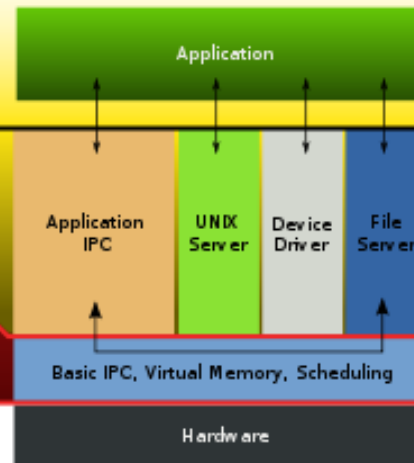
# Ανατομία του ΛΣ

Monolithic Kernel based Operating System



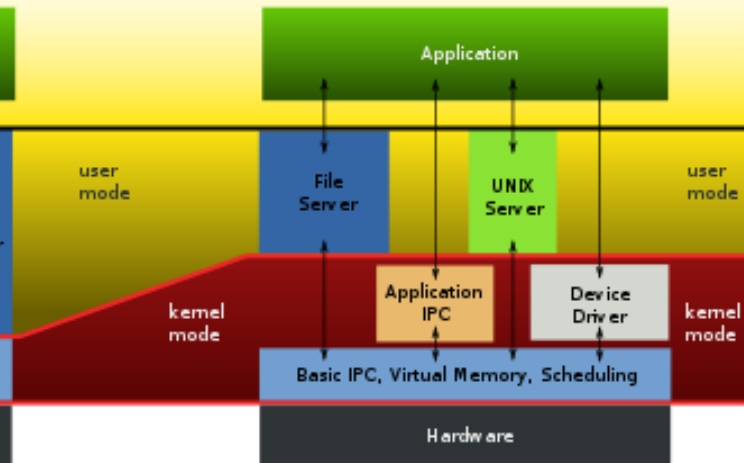
Most Modern Unixes (Linux, CentOS, BSD-like, Solaris, MacOS-8, AIX, Android, κτλ)

Microkernel based Operating System



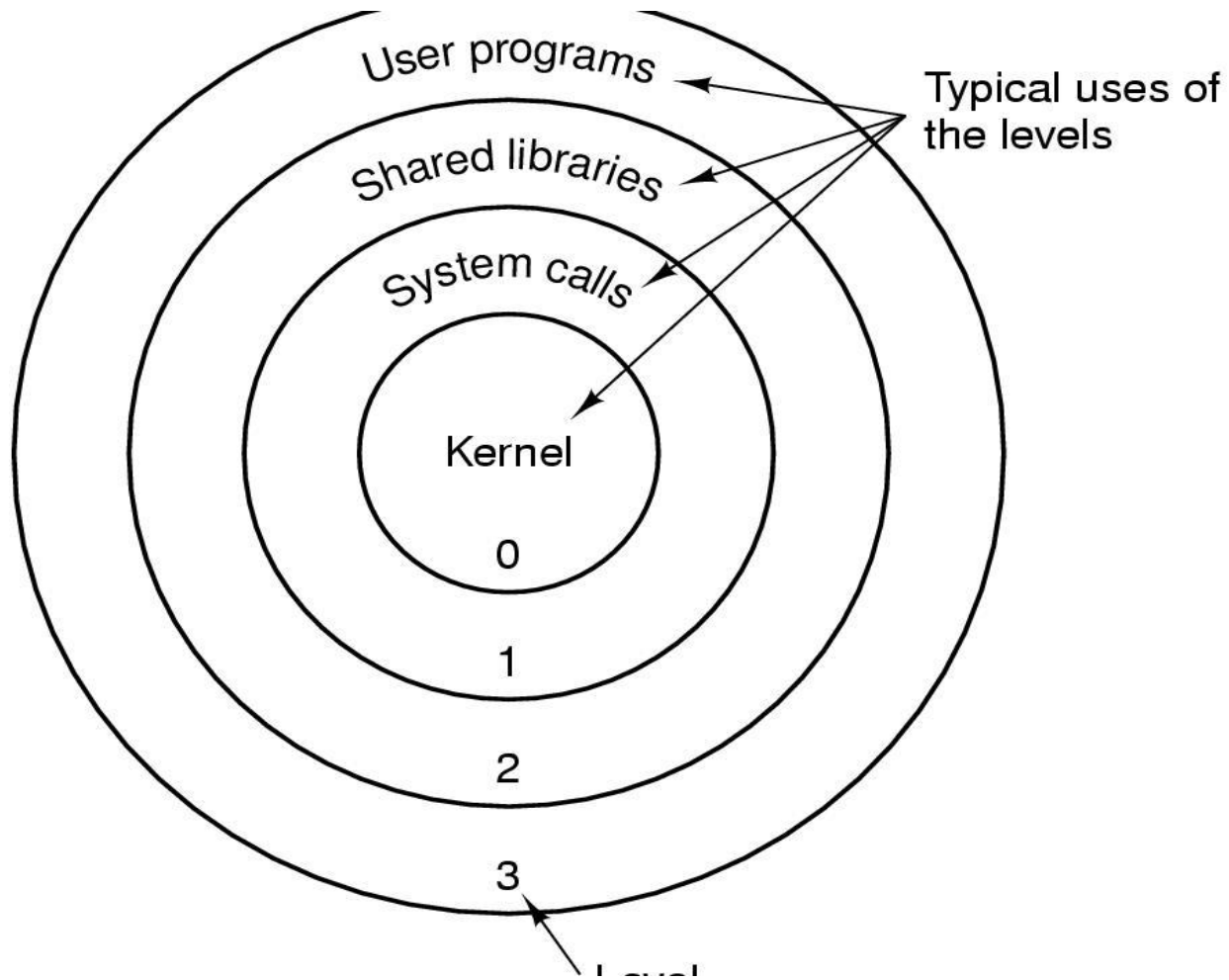
Π.χ., Early-UNIXes, MINIX-3 (Tanenbaum's) read <http://www.cs.vu.nl/~ast/reliable-os/>

"Hybrid kernel" based Operating System



Π.χ., MacOSX and NT-Kernel (i.e., Leopard, Windows NT, 2000, XP, Server 2003, Vista, Server 2008 and 7)

# Πολύ-επίπεδη Πρόσβαση στους Πόρους ενός ΗΥ



# Πυρήνας (Kernel)



- Το **kernel (πυρήνας)** είναι :
  - Ένα **πρόγραμμα** το οποίο φορτώνεται στην μνήμη ενός ΗΥ κατά την **διάρκεια της εκκίνησης** και παραμένει εκεί καθ' όλη την διάρκεια εκτέλεσης.
  - Έχει μεταξύ άλλων τα ακόλουθα υποσυστήματα:
    - **Διαχείριση Διεργασιών (Process management)**
      - Schedule processes to run on CPU, Inter-process communication (IPC)
    - **Διαχείριση Μνήμης (Memory management)**
      - Virtual memory (Paging/Swapping)
    - **Διαχείριση Εισόδου/Εξόδου (I/O system)**
      - File system, Device drivers, Buffer cache



# Κλήσεις Συστήματος (System Calls)

- Παρέχουν **διασύνδεση με το τον Πυρήνα**
- Στο **Linux** υπάρχουν περισσότερα από **1,000 system calls** και αυτά μπορούν κιόλας να επεκταθούν με **Loadable Kernel Modules (LKM)**
- Αυτά χωρίζονται σε τρεις βασικές κατηγορίες
  - File/device manipulation (Επεξεργασία Αρχείων)
    - e.g. `mkdir()`, `unlink()`
  - Process control (Διαχείριση Διεργασιών)
    - e.g. `fork()`, `execve()`, `nice()`
  - Information manipulation (Διαχείριση Μέτα-πληροφοριών), e.g. `getuid()`, `time()`



# Κέλυφος και Προγρ. Κελύφους / Shell and Shell Programming



- Στο ΕΠΛ421 θα δούμε πως αλληλεπιδράμε με τον πυρήνα: **α) Μέσω του Κελύφους (Bash) και Προγραμμάτων Ωφελιμότητας (System Utilities) και β) Μέσω Βιβλιοθηκών Συστημάτων (System Libraries)**

Χρήση Κέλυφους

```
who | awk '{print $1}' | sort | uniq
```

*Σημείωση:* Μας εκτυπώνει τα logins των χρηστών που είναι συνδεδεμένοι στο σύστημα μια δεδομένη στιγμή

- Βλέπουμε ότι με μια τέτοια γλώσσα (scripting, high level interpreted), μπορούμε να φτιάξουμε εύκολα σύνθετες λειτουργίες

who	755
awk	3,412
sort	2,614
uniq	302

7,424 γραμμές

κώδικα 1-17

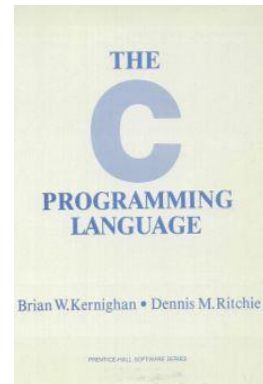


# Εισαγωγή στο Unix

- **1960:** Δημιουργία του Multics (mainframe timesharing /multitasking OS by MIT, GE, Bell Labs). Ο Thompson εργάζεται πάνω στον υπολογιστή PDP-7 στα Bell Labs
- **1969:** Ο Ken Thomson & ο Dennis Ritchie δημιουργούν ένα νέο ΛΣ για το υπολογιστή PDP-7 σε assembly το οποίο ονομάζουν Unics, το οποίο μετά ονομάζεται Unix!



- Ο Ritchie δημιουργεί την γλώσσα C για το Unix (**1972**) στα Bell Labs.
- ...και συγγραφέας του best-seller “The C Programming Language”!



Ken Thompson Dennis Ritchie



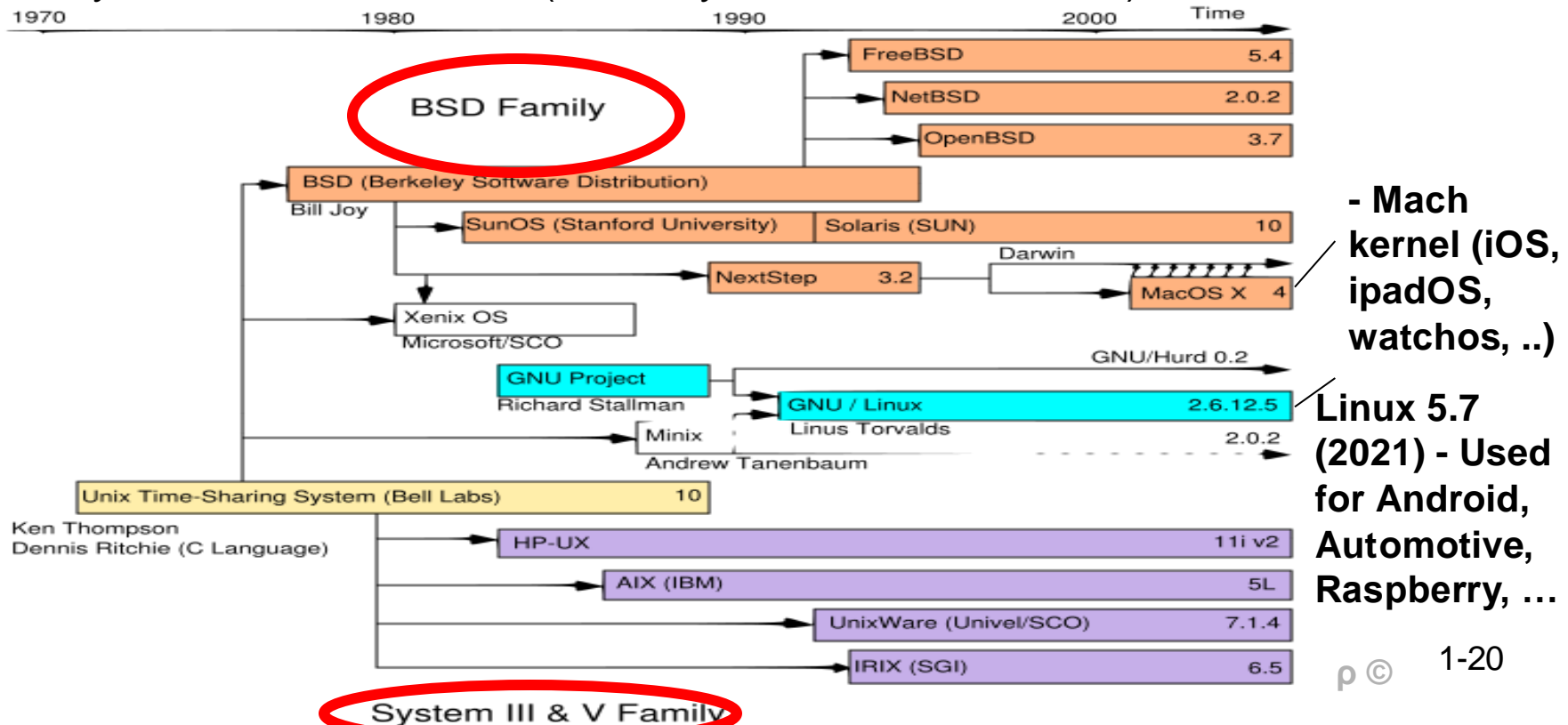
# To PDP-7 (1965)





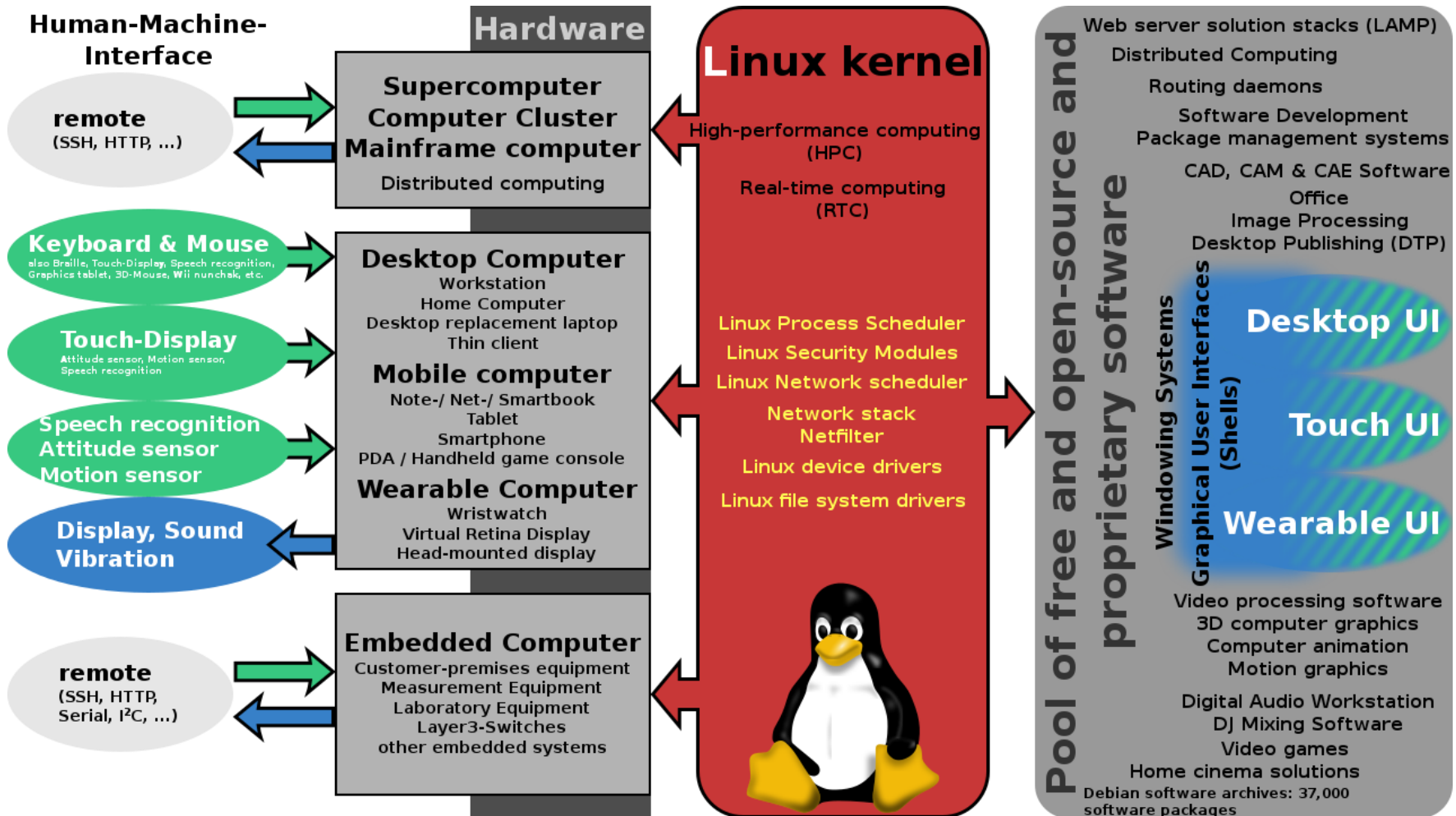
# Εισαγωγή στο Unix

- 1973: Το Unix γράφεται σε γλώσσα C.
- 1976 : Το κέλυφος Bourne Shell (bsh) αναπτύσσεται στην AT&T και επιτρέπει την αλληλεπίδραση με τον πυρήνα του Λ.Σ. μέσω εντολών .
- 80's- : Το Unix συνεχίζει να εξελίσσεται. Διακρίνονται δυο οικογένειες α) Το System III&V και το BSD (Berkeley Software Distribution).

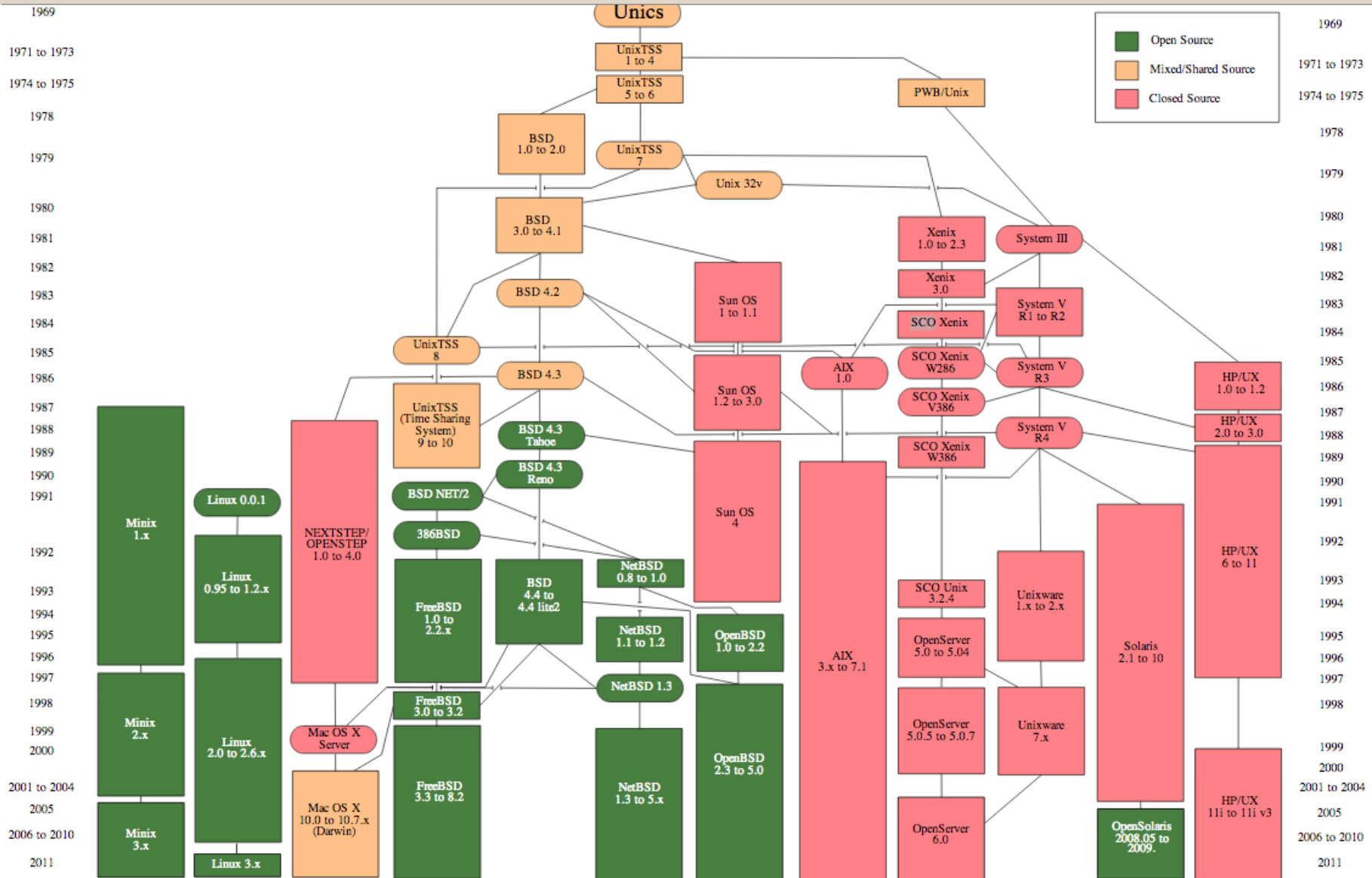




# Linux Everywhere









# Εισαγωγή στο Unix



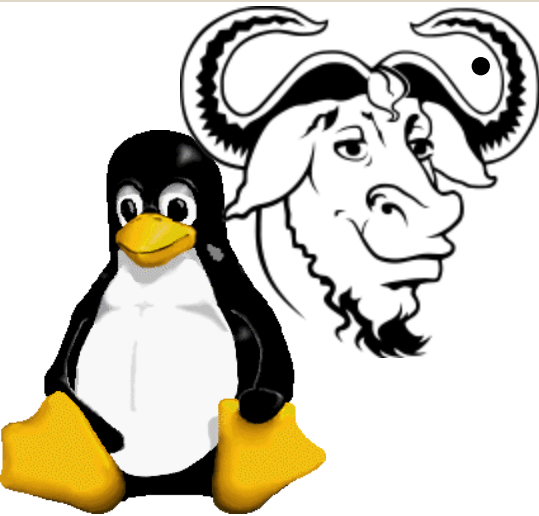
# Εισαγωγή στο Unix

## Εμπορική Επιτυχία



- AIX 
- SunOS -> Solaris (Sun -> Oracle) SUS-compliant 
- Ultrix, Digital Unix 
- HP-UX 
- Irix 
- UnixWare -> Novell -> SCO -> Caldera ->SCO' s  
Unixware
- Xenix:  -> SCO
- Standardization (**IEEE POSIX**: **P**ortable **O**perating **S**ystem **I**nterface for uni**X** (all known unices are POSIX-compliant), **SUS**: **S**ingle **U**nix **S**pecification (Austin), X/open (από διάφορες ευρωπαϊκές εταιρίες)

# Unix για Προσωπικούς Υπολογιστές PC



## (Gnu) Linux

- Γράφεται το 1991 από τον 21 ετών Φιλανδό Linus Torvalds
- Δημιουργεί ΛΣ UNIX για PCs
- Δωρεάν με **GNU license**
- **Τελευταία Έκδοση Kernel: 5.16 (01/2022)**

## • BSD Lite

---



- FreeBSD (1993, focus on PCs)
- NetBSD (1993, focus on portability)
- OpenBSD (1996, focus on security)
- Δωρεάν με **BSD license**



# Unix για Κινητές Συσκευές



## • Linux on Portable Devices

– Οι πυρήνες των πλείστων IoT συσκευων σήμερα στηρίζονται στο UNIX/Linux

- **Android** => uses and contributed to the Linux kernel (e.g., energy efficiency, drivers)

- <https://source.android.com/devices/architecture/kernel>

- **iOS (Apple)** => Darwin Kernel

  - A POSIX-compliant open source OS released by Apple in 2000

  - Darwin based on BSD -> Nextstep kernel (CMU)

- **Raspberry PI Linux**

– Linux RT: Android Automotive OS (Renault, Nissan and Mitsubishi), VW OS (VW/Audi/Skoda with Azure Backend), Mercedes (MB OS), Tesla OS (Tesla), Arene OS (Toyota/Subaru), Automotive Grade Linux ...



# UNIX for Artificial Intelligence



- Most tools in the modern AI era are founded on UNIX!
- Google Brain => Tensorflow
  - TensorFlow is a free and open-source software library for machine learning and artificial intelligence. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks.
  - *Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. **TensorFlow: a system for large-scale machine learning**. In **Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation (OSDI'16)**. USENIX Association, USA, 265–283.*

# UNIX for AI



Installing and instrumenting AI pipelines (especially at scale) requires UNIX skills.

- Let's see how!

## Download a package

Install TensorFlow with Python's *pip* package manager.

★ TensorFlow 2 packages require a *pip* version >19.0 (or >20.3 for macOS).

Official packages available for Ubuntu, Windows, and macOS.

[Read the pip install guide](#)

```
# Requires the latest pip
$ pip install --upgrade pip

# Current stable release for CPU and GPU
$ pip install tensorflow

# Or try the preview build (unstable)
$ pip install tf-nightly
```

## Run a TensorFlow container

The [TensorFlow Docker images](#) are already configured to run TensorFlow. A [Docker](#) container runs in a virtual environment and is the easiest way to set up GPU support.

```
$ docker pull tensorflow/tensorflow:latest # Download latest stable image
$ docker run -it -p 8888:8888 tensorflow/tensorflow:latest-jupyter # Start Jupyter server
```



# Unix for AI

- Jupyter Notebook Integration

```
sudo docker run -it -p 8888:8888 tensorflow/tensorflow:latest-jupyter
[I 06:58:28.960 NotebookApp] Writing notebook server cookie secret to /root/.local
t
jupyter_http_over_ws extension initialized. Listening on /http_over_websocket
[I 06:58:29.308 NotebookApp] Serving notebooks from local directory: /tf
[I 06:58:29.308 NotebookApp] Jupyter Notebook 6.5.2 is running at:
[I 06:58:29.309 NotebookApp] http://1b9f569b2c54:8888/?token=b29d7e2b312fb9bfec5f234a83c403e9a2ba7f244c2
[I 06:58:29.309 NotebookApp] or http://127.0.0.1:8888/?token=b29d7e2b312fb9bfec5f234a83c403e9a2ba7f244c2
[I 06:58:29.309 NotebookApp] Use Control-C to stop this server and shut down all
[C 06:58:29.314 NotebookApp]
```

To access the notebook, open this file in a browser:

`file:///root/.local/share/jupyter/runtime/nbserver-1-open.html`

Or copy and paste one of these URLs:

`http://1b9f569b2c54:8888/?token=b29d7e2b312fb9bfec5f234a83c403e9a2ba7f244c2`

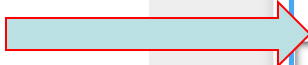
or `http://127.0.0.1:8888/?token=b29d7e2b312fb9bfec5f234a83c403e9a2ba7f244c2`



# UNIX for AI

- BASH Integration!

```
10.16.20.11:8888/notebooks/tensorflow-tutorials/Untitled.ipynb?kernel_name=python3
Jupyter Untitled Last Checkpoint: 7 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3
Run Code
In [1]: print("Hello, World!")
Hello, World!
In [8]: %%bash
curl -vvv https://www.ucy.ac.cy/
% Total % Received % Xferd Average Speed Time Time Time Current
Dload Upload Total Spent Left Speed
<IDOCTYPE html>
<html lang="el" class="no-js " >
<head>
<meta charset="UTF-8" />
<meta name="description" content="University of Cyprus | Holds the EFQM "Committed to Excellence" Award and acknowledged with the "HR Excellence in Research" logo"/>
<meta name="keywords" content="university public academic research academic study education excellence"/>
<style>
form#stickyelements-form input::-moz-placeholder{
color: #4F4F4F;
}
form#stickyelements-form input::-ms-input-placeholder{
color: #4F4F4F
}
form#stickyelements-form input::-webkit-input-placeholder{
color: #4F4F4F
}
In [10]: %%bash
ps -ef
UID PID PPID C STIME TTY TIME CMD
root 1 0 0 06:58 pts/0 00:00:06 /usr/bin/python3 /usr/local/bin/jupyter-notebook --notebook-dir=/tf -
-ip 0.0.0.0 --no-browser --allow-root
root 33 1 0 07:03 ? 00:00:01 /usr/bin/python3 -m ipykernel_launcher -f /root/.local/share/jupyter/
runtime/kernel-615c64fb-c728-439f-9939-537fed362cde.json
root 45 1 0 07:16 ? 00:00:01 /usr/bin/python3 -m ipykernel_launcher -f /root/.local/share/jupyter/
runtime/kernel-efe9befc-185a-4655-bac2-192ffea3a84.json
root 62 45 0 07:22 ? 00:00:00 bash
root 64 62 0 07:22 ? 00:00:00 ps -ef
```





# UNIX for Blockchain

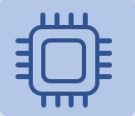
- Hyperledger Fabric (IBM) is a software stack for building Permissioned Blockchains.
  - Install:
    - `curl -sSL https://bit.ly/2ysbOFE | bash -s -- 2.2.2 1.4.9 # the url sends a BASH script and “bash –s” executes this.`
  - Usage
    - Requires lots of unix commands to manage the P2P network.
      - Check: [https://hyperledger-fabric.readthedocs.io/en/release-2.2/test\\_network.html#bring-up-the-network-with-certificate-authorities](https://hyperledger-fabric.readthedocs.io/en/release-2.2/test_network.html#bring-up-the-network-with-certificate-authorities)



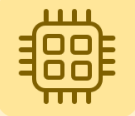
# Unix/C Support Everywhere



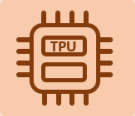
- **CPU (Intel, ARM, M)** => x64 or ARM-based
- **GPU (NVIDIA)** => Gaming, Ray Tracing, Machine Learning
- **APU (AMD Ryzen)** = GPU + CPU (e.g., PS5)
- **TPU Tensor Processing Unit** => Machine Learning (e.g., Google)
- **Future:**

 **CPU**


- Small models
- Small datasets
- Useful for design space exploration

 **GPU**

- Medium-to-large models, datasets
- Image, video processing
- Application on CUDA or OpenCL

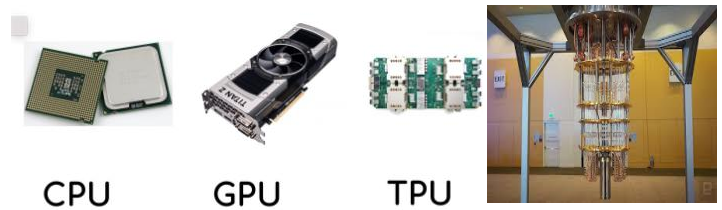
 **TPU**

- Matrix computations
- Dense vector processing
- No custom TensorFlow operations

 **FPGA**

- Large datasets, models
- Compute intensive applications
- High performance, high perf./cost ratio

- QPU => Quantum Processing Units (e.g., IBM Q accessible for exponential programs with ~50 Qubits only so far)



# Unix για Υποδομές Νεφέλης (Large Filesystems - HDFS)



**YAHOO!**

2010

	Target	Deployed
Capacity	10PB	14PB
Nodes	10,000	4000
Clients	100,000	15,000
Files	100,000,000	60,000,000

**facebook**

2010

- 21 PB of storage in a single HDFS cluster
- 2000 machines
- 12 TB per machine (a few machines have 24 TB each)
- 1200 machines with 8 cores each + 800 machines with 16 cores each
- 32 GB of RAM per machine
- 15 map-reduce tasks per machine

**TABLE 1: TARGETS FOR HDFS VS. ACTUALLY DEPLOYED VALUES  
AS OF 2009**

**HDFS scalability: the limits to growth**

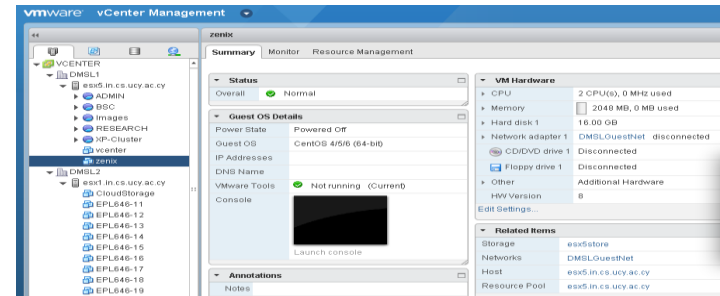
<http://static.usenix.org/publications/login/2010-04/openpdfs/shvachko.pdf>



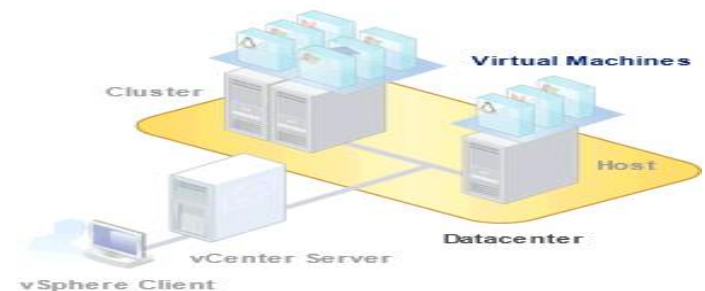


# Εικονικοποίηση (Virtualization)

- **Εικονικοποίηση (virtualization)** αναφέρεται στη διαδικασία δημιουργίας μιας νοητής, πάρα πραγματικής έκδοσης, κάποιου μέσου (π.χ., Η/Υ, μνήμης, δικτύου, υπηρεσίας, Λ.Σ., εφαρμογών, κτλ.)



Εικονικές Μηχανές



Πραγματικό Υλικό

# Εικονικοποίηση (Virtualization)



## Google's Datacenter in Oregon



Πραγματικό Υλικό



Εικονική Υπηρεσία  
(Virtualized Service)



# Video of SAP Datacenter



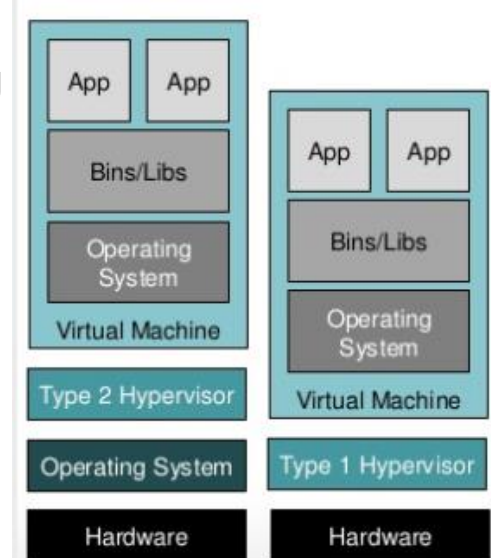
## SAP's Data Centers – A Safe Place for Data

Learn how SAP's data centers ensure the safety and accessibility of your data 24/7 based on the highest standards of the industry with the latest technology.

# Hypervisors (Επτόπτες) και Virtual Machines (VMs)



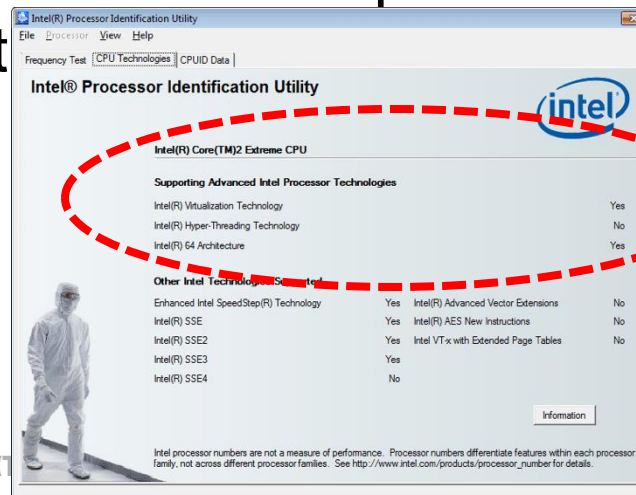
- **Hypervisors:** πρόγραμμα το οποίο δημιουργεί και εκτελεί **εικονικές μηχανές (virtual machines)**.
- **Type-1: native or bare-metal hypervisors**
  - Τρέχουν απευθείας πάνω στο υλικό. Κατάλληλα για διαθέτες (servers). Συνήθως προσαρμοσμένα LINUX kernels.
    - Παραδείγματα: Xen (open source), Oracle VM Server, Hyper-V (Microsoft), ESX/ESXi (VMWARE).
    - Γνωστά ως IaaS στην ορολογία του Cloud Computing
- **Type-2 or hosted hypervisors**
  - Τρέχουν ως πρόγραμμα (user-space) πάνω από το Kernel ενός άλλου Λ.Σ. (MacOSX, Windows, Linux)
    - Παραδείγματα: VMware Workstation, VMware Player, VirtualBox, Parallels Desktop for Mac and QEMU



# Virtualization Support at the Hardware Level



- Hardware support to hypervisors comes nowadays standard in popular x86-64 (Intel & AMD) processors.
  - **AMD-v/Intel VT-x:** allows a guest virtual machine to run at privilege levels in the processor that enable proper operation.
  - **AMD-Vi/Intel VT-d:** is a processor feature that enables virtual machines (directed I/O).



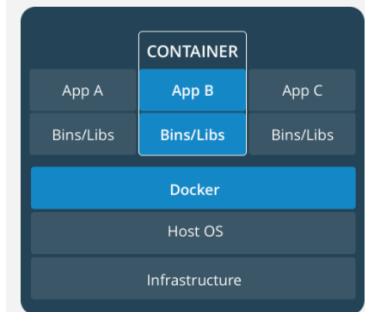
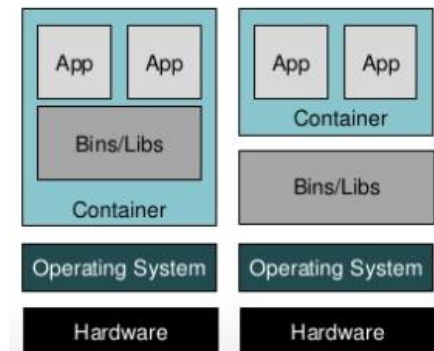
**Also supported in AMD Ryzen ( Secure Virtual Machine (SVM) in BIOS.)**



# Virtual Private Server

- **Virtual Private Server (VPS):** αξιοποιεί Εικονικοποίηση για να μοιράσει τον διαθέτη σε **containers**, το κάθε ένα εκ των οποίων λειτουργεί ως ξεχωριστός διαθέτης

- Κάθε VPS επιτρέπει root access, process management, stop/reboot VPS, κτλ.
- Παράδειγμα: **OpenVZ**, [Solaris Containers](#) & **LXC (Linux Containers)**
  - Στο μάθημα ενδέχεται να πάρετε 1 OpenVZ VPS με root δικαιώματα .
  - VPS δημιουργήθηκαν αρχικά για λόγους ασφάλειας (chroot jails)
- Σήμερα είναι της μόδας software VPS τα οποία χρησιμοποιούνται για την διανομή λογισμικού (π.χ. Docker PaaS)
  - Ξεφεύγουμε από εικονικοποίηση Λ.Σ. και εισερχόμαστε στον χώρο της Εικονικοποίησης υψηλότερων επιπέδων (εκτός ΕΠΛ421)



# IaaS, PaaS, SaaS



## Separation of Responsibilities

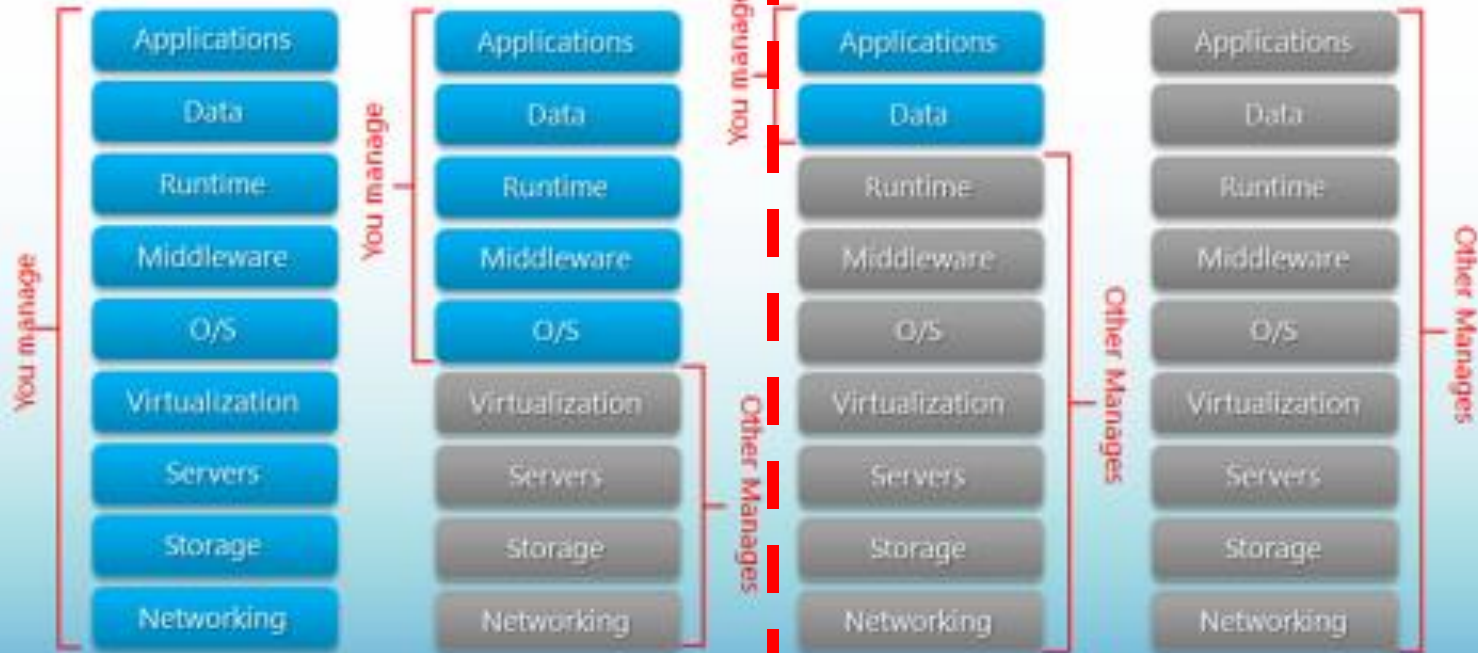
**EPL421**

On-Premises

e.g., Amazon  
EC2  
Infrastructure  
(as a Service)

e.g.,  
SQLserver  
Azure  
Platform  
(as a Service)

e.g., GMAIL  
for business  
Software  
(as a Service)

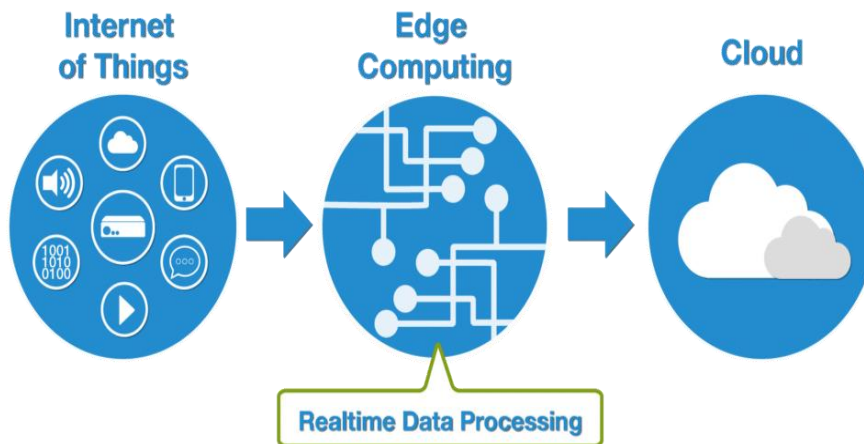




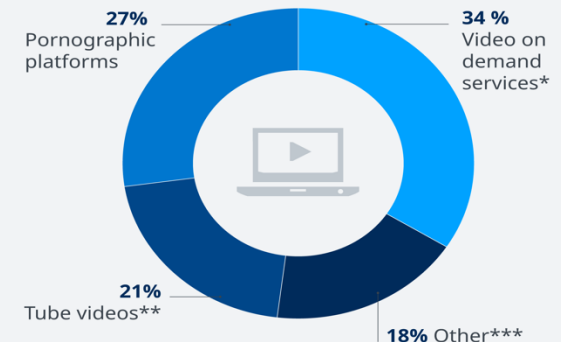
# Edge Computing

Push datacenters closer to users:

- Minimize user latency
- minimize energy costs (less net traffic)



Global online video traffic



\*Netflix, Amazon, etc. \*\*Youtube, vimeo, etc.

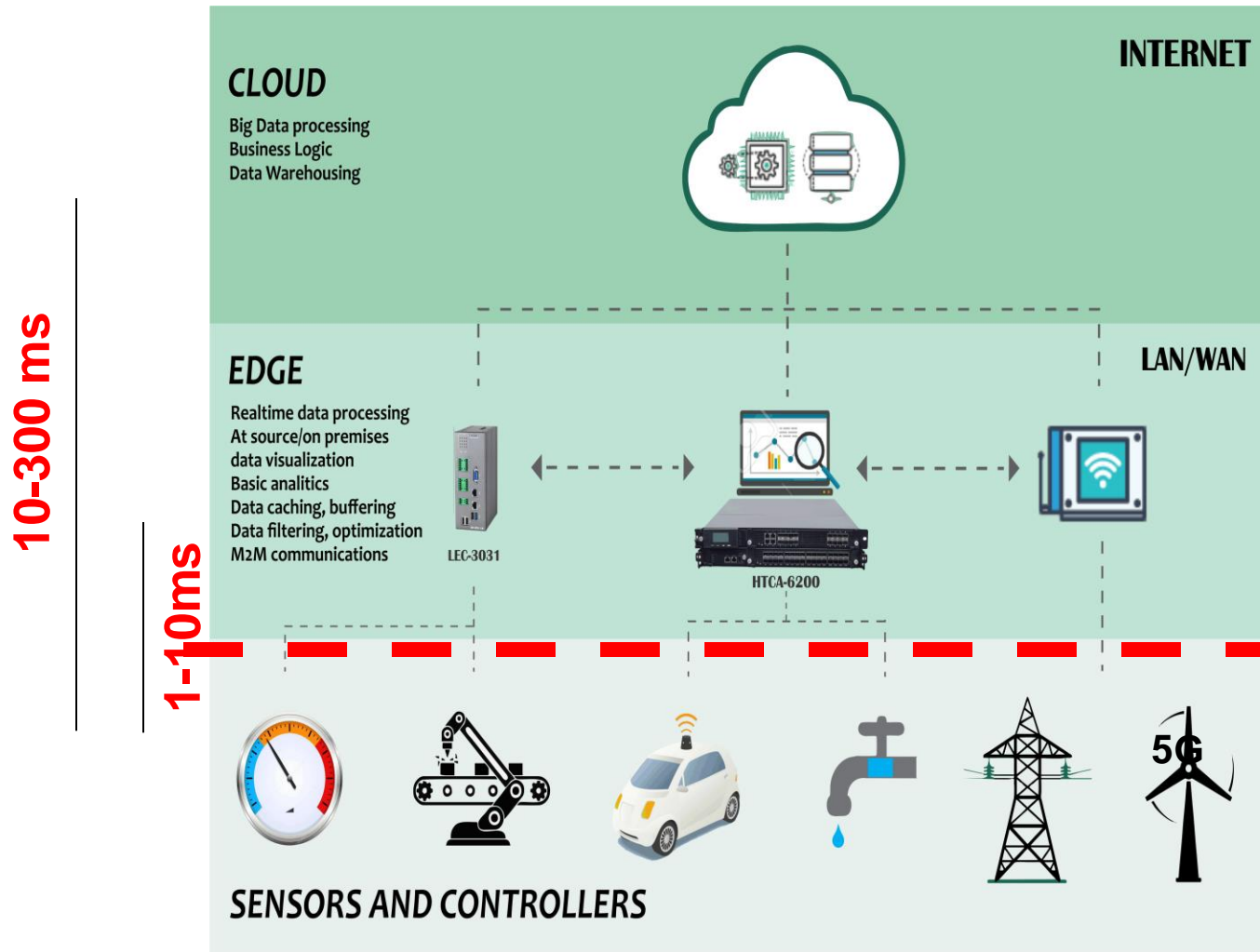
\*\*\*Social media and other platforms

Source: Lean ICT - Towards Digital Sobriety / The Shift Project 2019

©DW



# Edge Computing



NY Stock Exchange Example

Industry 4.0 Example (robotic decision on the Edge)

**Objective Latency: 1ms!**



# Edge Computing

## # Google Datacenters are more widespread and closer

```
$ping google.com  
PING google.com (172.217.171.238): 56 data bytes  
64 bytes from 172.217.171.238: icmp_seq=0 ttl=112 time=36.124 ms
```

## # Amazon (US) Datacenters are farther away

```
$ ping amazon.com  
PING amazon.com (205.251.242.103): 56 data bytes  
64 bytes from 205.251.242.103: icmp_seq=0 ttl=226 time=137.962 ms
```

## # Amazon (Germany) Datacenters are closer, still not within 1 ms

```
$ ping amazon.de  
PING amazon.de (52.95.120.34): 56 data bytes  
64 bytes from 52.95.120.34: icmp_seq=0 ttl=216 time=78.836 ms
```





## # Let's find the latency of something local - We reached the EDGE!

```
$ ping www.ucy.ac.cy  
PING ucyweb.ucy.ac.cy (194.42.1.70): 56 data bytes  
64 bytes from 194.42.1.70: icmp_seq=0 ttl=58 time=1.212 ms
```



# Edge Computing

- Objective to bring

		3G	4G	5G
	Deployment	2004-05	2006-10	2020
	Bandwidth	2mbps	200mbps	>1gbps
	Latency	100-500 milliseconds	20-30 milliseconds	<10 milliseconds
	Average Speed	144 kbps	25 mbps	200-400 mbps

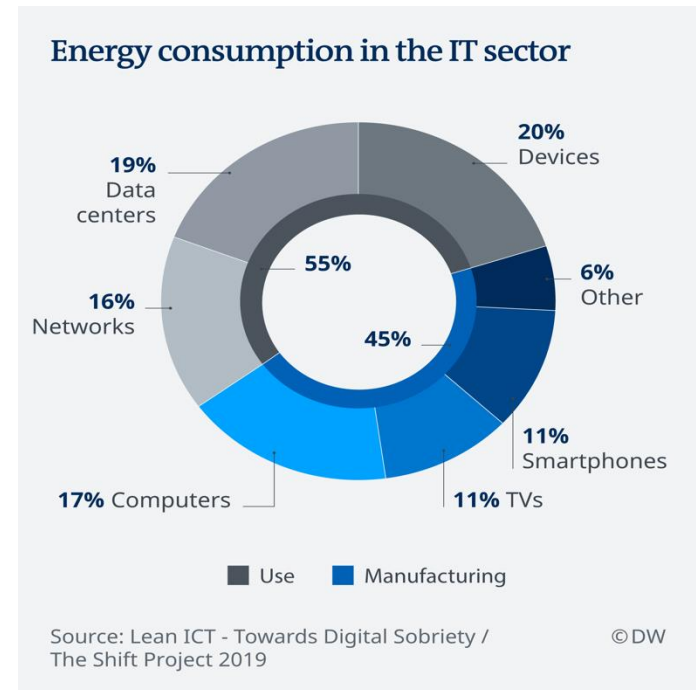


# Green Computing

- **Green computing** is the environmentally responsible and eco-friendly use of **computers** and their resources.

- The study of designing, manufacturing/engineering, using and disposing of computing devices in a way that reduces their environmental impact.

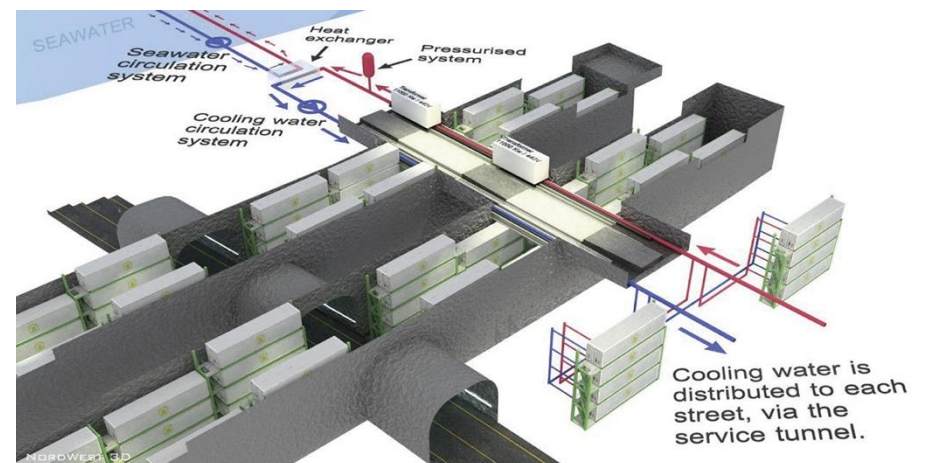
<https://whitelabelitsolutions.com/meaning-green-computing/>





# Norway's Biggest Data Center

- Norway generates 90% of its energy from hydrodynamic.
- Lefdal Mine Data Center: Where the cloud goes underground. Why?
- Heating from machines used to warm up neighbourhoods



<https://itbizadvisor.com/2017/05/lefdal-mine-data-center-where-the-cloud-goes-underground/>

# Finding Hardware Specs



```
<ada>$ cat /proc/cpuinfo
```

```
processor: 0                                # vmware -v
vendor_id: GenuineIntel                    VMware ESX Server 3i 3.5.0 build-
cpu family: 6                               158869
model: 13
model name: QEMU Virtual CPU version (cpu64-rhel6)
stepping: 3
microcode: 1
cpu MHz: 2400.084
cache size: 4096 KB
physical id: 0
siblings: 1
core id: 0
cpu cores: 1
apicid: 0
initial apicid: 0
fpu: yes
fpu_exception: yes
cpuid level: 4
wp: yes
```

```
$cat /proc/version
```

```
Linux version 5.4.0-96-generic (buildd@lgw01-
amd64-051) (gcc version 9.3.0 (Ubuntu 9.3.0-
17ubuntu1~20.04)) #109-Ubuntu SMP Wed Jan 12
16:49:16 UTC 2022
```



# Περιεχόμενο Μαθήματος

- **Εβδ. 1-3: Εντολές UNIX και Εργαλεία Ωφελιμότητας (System Utilities)**  
Εισαγωγή, Ιστορική Αναδρομή, Απλές και προχωρημένες εντολές, Κανονικές εκφράσεις, Ανακατεύθυνση (redirect) και διοχέτευση (pipe), Διαχείριση Αρχείων, Διεργασίες, Εντολές περιβάλλοντος, επικοινωνίας, εισόδου/εξόδου, ελέγχου διεργασιών, Εργαλεία Ωφελιμότητας: sed, awk κτλ.)
- **Εβδ. 4-5: Προγραμματισμός στο Κέλυφος (bash)**  
Εισαγωγή, Μεταβλητές, Συνθήκες Ελέγχου (if, select), Επαναληπτικοί Βρόγχοι (for, while, until), Έλεγχος Ροής (break, continue), Συναρτήσεις, Πίνακες, λογικοί και αριθμητικοί τελεστές κ.α.



# Περιεχόμενο Μαθήματος

- **Εβδ. 6: Συστήματα Αρχείων.**

Εισαγωγή στα Αρχεία και Συστήματα Αρχείων στο Unix (Τύποι Αρχείων, Partitions, i-nodes, blocks), Μέθοδοι Επεξεργασίας Αρχείων: Standard I/O vs. Χαμηλού Επιπέδου I/O, Μέτα-πληροφορίες Αρχείων, Δομή και Πρόσβαση, Διαχείριση Αρχείων με κλήσεις συστήματος, Διαχείριση Καταλόγων, Προσπέλαση Καταλόγων με συναρτήσεις βιβλιοθήκης

- **Εβδ. 7-8: Διεργασίες: Περιβάλλον, Έλεγχος και Σήματα**

Διεργασίες: Δομή & Αναπαράσταση, Ταυτότητες Διεργασιών, Δημιουργία Διεργασιών, Μεταβλητός Περιβάλλοντος, Ορφανές Διεργασίες, Διεργασίες και Αρχεία, Αναμονή Διεργασιών, Zombie Διεργασίες, Σήματα και Διεργασίες

**MIDTERM**





# Περιεχόμενο Μαθήματος

- **Εβδ. 9-10: Δια-διεργασιακή Επικοινωνία (IPC)**

Επικοινωνία μεταξύ Διεργασιών, Μελέτη τρόπων επικοινωνίας: i) Σωλήνες (Pipes), ii) FIFO (Named Pipes), iii) Ουρές Μηνυμάτων (Message Queues), iv) Κοινόχρηστη Μνήμη (Shared Memory) με Σηματοφόρους (Semaphores) και v) Προγραμματισμός Υποδοχών (Socket Programming).

- **Εβδ. 11: Πολυνηματικός Προγραμματισμός (Multithreading)**

*Πλεονεκτήματα Νημάτων, Είδη Νημάτων, Πολυνηματικά Μοντέλα, Η Βιβλιοθήκη <rthread.h> και παραδείγματα, Θέματα διαχείρισης ταυτοχρονίας, διαχείριση αδιεξόδων.*

- **Εβδ. 12: Παρουσιάσεις Φοιτητών**

**Systems Programming in Windows:** Sockets, Process Management, Threads and Concurrency, IPC

**Scripting Languages:** PHP, Ruby, Python, Perl, Smalltalk

- **Εβδ. 13: Προχωρημένα Θέματα**

*Map-Reduce, DevOps, κτλ.*