Sistemas Operativos Avanzados

vpuente@unican.es

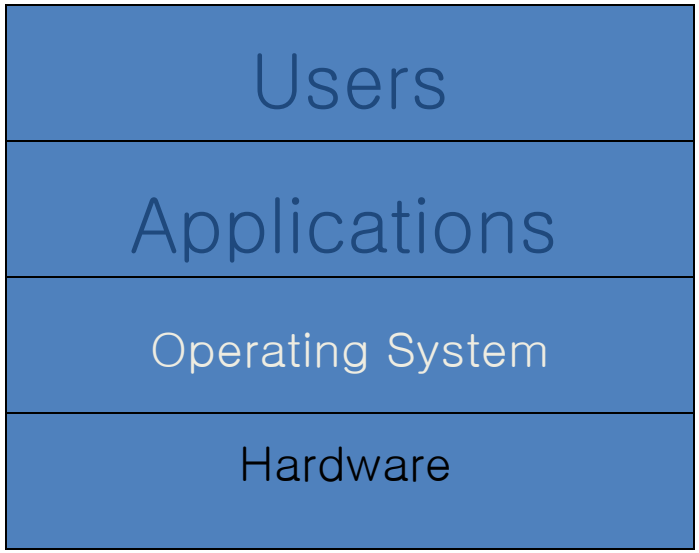
www.ce.unican.es

What is an Operating System?

Operating System (OS):

Software that converts hardware into a useful form for applications

Not easy to define precisely…



What DOES OS Provide?

- Role #1: Abstraction - Provide standard library for resources
- What is a resource?
  - Anything valuable (e.g., CPU, memory, disk, I/O device)
- What abstraction does modern OS typically provide for each resource?
  - CPU:
    - process and/or thread
  - Memory:
    - address space
  - Disk:
    - files
- Advantages of OS providing abstraction?
  - Allow applications to reuse common facilities
  - Make different devices look the same
  - Provide higher-level or more useful functionality
- Challenges
  - What are the correct abstractions?
  - How much of hardware should be exposed?

What DOES OS Provide?

- Role #2: Resource management – Share resources well

- Advantages of OS providing resource management?
  - Protect applications from one another
  - Provide efficient access to resources (cost, time, energy)
  - Provide fair access to resources

- Challenges
  - What are the correct mechanisms?
  - What are the correct policies?

OS Organization

- How to cover all the topics relevant to operating systems?

Three PIECES

- Virtualization:
  - Make each application believe it has each resource to itself
- Concurrency:
  - Events are occurring simultaneously and may interact with one another
- Persistence: Access information permanently
  - Lifetime of information is longer than lifetime of any one process
  - Machine may be rebooted, machine may lose power or crash unexpectedly

Advanced Topics (beyond our reach)

- Current systems
  - Multiprocessors
  - Networked and distributed systems
  - Virtual machines
  - Containers
  - …
- Many of the pushed by the explosive demand (a.k.a. Massive complexity under constrained cost)
- This is the support of the world: it will keep changing …
- Some of them covered in SVS (M1679)

Why study Operating Systems?

- Build, modify, or administer an operating system
- Understand system performance
  - Behavior of OS impacts entire machine
  - Tune workload performance
  - Apply knowledge across many layers
    - Computer architecture, programming languages, data structures and algorithms, and performance modeling
- Fun and challenging to understand large, complex systems
- **Is the glue that "holds" all the ideas in place**

Approach

- We will follow "Operating System: Three Easy Pieces" (OSTEP) style
  - From the **bottom** concepts to state-of-the-art approaches
  - Eminently **practical** style: all supported by "simulators" and simple coding examples
  - Assumes some **basic** knowledge in architecture, C, assembler and system administration
  - More than just a textbook…

- Structure
  - The three parts are split in small *pieces* (~40 in the book)
  - Each chapter is built over the previous one (can't miss the beat)
  - Each chapter has attached a "Homework" to reinforce the : from using python simulators to write small pieces of code (C)
  - 5 + 1 Labs, developed on top of **xv6**

Lecture/Lab structure

- We mix dynamically both
  - The real thing is that there is no separation between "theory" and "lab"

- Sessions of:
  - 1st hour: Introduction to the topic
  - 2nd hour: Introduce/develop of Labs
  - Personal work (out the lab): 6 hours (labs and homework)
  - 10 hours/week
  - Strict schedule

- Although the original course/book is designed for 15-week semester (150h work), we will need to drop some details or advanced topics (and half of the labs)

Material

- Available in http://www.ce.unican.es/

- All written material will be in "English"
  - Lecture notes, Homework/Lab guides, etc....

- *Git* as communication "device": all material will be delivered via http://gitlab.com (lab work) and http://github.com (lecture notes)
  - An e-mail inviting to join the course project will be sent to unican account
  - Slides, labs, other reference material is there
  - It uses "git" to have a "time-track"
    - Lecture notes updates
    - Additional material

- Use git to allow you and me "track" your personal work

- http://piazza.com is a great tool to resolve issues and collaborate (support **anonymous** questions!)

Book (ostep.org)

This book **is and will always be free** in PDF form, as seen below. For those of you wishing to **BUY** a copy, please consider the following:

- Lulu Hardcover (v1.00): this may be the best printed form of the book (it really looks pretty good), but it is also the most expensive way to obtain *the black book* of operating systems (a.k.a. *the comet book* or *the asteroid book* according to students). Now just: **$38.00**
- Lulu Softcover (v1.00): this way is pretty great too, if you like to read printed material but want to save a few bucks. Now just: **$22.00**
- Amazon Softcover (v1.00): Same book as softcover above, but printed through Amazon CreateSpace. Now just: **$25.90** (but works with Prime shipping)
- Downloadable PDF (v1.00): this is a nice convenience and adds things like a hyperlinked table of contents, index of terms, lists of hints, tips, systems advice, and a few other things not seen in the free version, all in one massive DRM-free PDF. Once purchased, you will always be able to get the latest version. Just: **$10.00**
- Kindle: Really, just the PDF and does not include all the bells and whistles common in e-pub books.

New Partnership: We have a new partnership with Educative; they offer a way to take an OS course (based on OSTEP) through their platform at a low cost. However, don't worry: the book on this website is and will always remain free.

Lulu Discount Codes: These always exist in some form, look around for one?

Warning: Some resellers on Amazon buy old versions of the books and claim to sell them as "new" on Amazon (click here for an example); buy from them at your own risk. In general, buy either directly from Lulu.com or Amazon.com (not a reseller). For Amazon, go to this page and look for Seller Information to be Amazon.com.

Can't bear to go out in public without OSTEP? How about an Operating Systems: Three Easy Pieces T-shirt or laptop sticker or bathmat or blanket or mug or check out the whole store?

Donate: By popular demand, another way to support this site and its contents: **donate!** Click to donate $1 - $10 - $20 - $50 - or click here to donate any amount you want! Your donation helps keep this book going. Think about it: if everyone who came to this website donated just one dollar, we'd have at least three dollars. Thanks!

Another way to help the book out: cite it! Here is the BiBTeX entry (seen below); you can also link to the site of the best free operating systems book on the market.

> **Operating Systems: Three Easy Pieces**
> Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau
> Arpaci-Dusseau Books
> August, 2018 (Version 1.00)

And now, the free online form of the book, in chapter-by-chapter form (now with chapter numbers!):

Intro	Virtualization		Concurrency	Persistence	Security
Preface	3 *Dialogue*	12 *Dialogue*	25 *Dialogue*	35 *Dialogue*	52 *Dialogue*
TOC	4 Processes	13 Address Spaces code	26 Concurrency and Threads code	36 I/O Devices	53 *Intro Security*
1 *Dialogue*	5 Process API code	14 Memory API	27 Thread API code	37 Hard Disk Drives	54 *Authentication*
2 Introduction code	6 Direct Execution	15 Address Translation	28 Locks code	38 Redundant Disk Arrays (RAID)	55 *Access Control*
	7 CPU Scheduling	16 Segmentation	29 Locked Data Structures	39 Files and Directories	56 *Cryptography*
	8 Multi-level Feedback	17 Free Space Management	30 Condition Variables code	40 File System Implementation	57 *Distributed*
	9 Lottery Scheduling code	18 Introduction to Paging	31 Semaphores code	41 Fast File System (FFS)	
	10 Multi-CPU Scheduling	19 Translation Lookaside Buffers	32 Concurrency Bugs	42 FSCK and Journaling	**Appendices**
	11 *Summary*	20 Advanced Page Tables	33 Event-based Concurrency	43 Log-structured File System (LFS)	*Dialogue*
		21 Swapping: Mechanisms	34 *Summary*	44 Flash-based SSDs	Virtual Machines
		22 Swapping: Policies		45 Data Integrity and Protection	*Dialogue*
		23 Complete VM Systems		46 *Summary*	Monitors
		24 *Summary*		47 *Dialogue*	*Dialogue*
				48 Distributed Systems	Lab Tutorial
				49 Network File System (NFS)	Systems Labs
				50 Andrew File System (AFS)	xv6 Labs
				51 *Summary*	

INSTRUCTORS: If you are using these free chapters, **please just link to them directly** (instead of making a copy locally); we make little improvements frequently and thus would like to provide the latest to whomever is using it. Also: we have made our own class-preparation notes available to those of you teaching from this book; please drop us a line at **remzi@cs.wisc.edu** if you are interested.

Homework

- Some chapter (most) include homework
  - Homework can be used to solidify your knowledge of the material in each of the chapters
  - Most homework are based on running little **simulators,** which mimic some aspect of an operating system: For example, a disk scheduling simulator could be useful in understanding how different disk scheduling algorithms work:
    - Most of them provides the **solution**
  - Some home-works are just short programming exercises, allowing you to explore how real systems work and complement Lab work.

- Homework are done in **personal-time**

Labs: C and xv6

- Refresh C knowledge
- Use a “toy” kernel to dig into implementation details
  - It is a clean and beautiful little kernel, and thus a perfect object for our study and usage.
  - It was developed by OS Eng. In MIT as a port of K&R original Unix R6/PDP11 (6.828 and 6.S081)
  - Use al real kernel (such as linux) will be certainly overkill

```
cigal xv6-wisc (master)*$ cloc *
     145 text files.
     143 unique files.
      15 files ignored.

http://cloc.sourceforge.net v 1.64  T=1.34 s (99.9 files/s, 8217.7 lines/s)
-------------------------------------------------------------------------------
Language                     files          blank        comment           code
-------------------------------------------------------------------------------
C                               45            946            621           5855
Assembly                         9             58            124           1748
C/C++ Header                    20            177            138            955
D                               57              0              0            154
make                             1             40             47             90
Perl                             2             11             22             33
-------------------------------------------------------------------------------
SUM:                           134           1232            952           8835
-------------------------------------------------------------------------------
```

Prerequisites

- All OS and architecture previous subjects(ugh!)

Evaluation

- Full details in:
  - http://web.unican.es/estudios/Documents/Guias/2021/es/G677.pdf

- 20% Parciales
  - 10% Lab
  - 10% Teoria
- 80% Examen final
  - 40% Lab
  - 40% Teoria

Schedule

Date	Chapter	Lab	Homework
6-sep.	1 Intro	P0 Lab Intro and review C	
8-sep.	4. The Abstraction: The Process/ 5. Interlude: Process API		Process Intro / Process API
13-sep.	6. Mechanism: Limited Direct Execution		Direct Execution
20-sep.	7. Scheduling: Introduction	P0 Due, P1 System Calls	Scheduler
22-sept	8: Scheduling: The Multi-Level Feedback Queue		MLFQ Scheduling
27-sep.	9: Scheduling: Proportional Share/10. Multiprocessor Scheduling		Lottery Scheduling
29-sep.	13. The Abstraction: Address Space / 14. Memory API	P1 Due, P2 Scheduling	VM API
4-oct.	15. Address Translation		Relocation
6-oct.	16. Segmentation/17. Free-Space Management		Segmentation
11-oct.	18 Pagin Intro.	*P2 Due*	Free Space
13-oct.	19. Translation Lookaside Buffers	*P3 Memory*	Paging
18-oct.	20. Paging: Smaller Tables		TLBs
20-oct	21. Swapping: Mechanisms/22. Swaping: Policies		Multi-level Paging/Paging Mechanism
25-oct	26. Concurrency: An Introduction / 27. Interlude: Thread API		Threads (Intro)/Threads (API)
27-oct	28. Locks		Threads (Locks)
3-nov	29. Lock-based Concurrent Data Structures		
8-nov	30. Condition Variables	P3 Due, P4 Threads	Threads (CVs)
????	Mid Term Exam (Processes & Memory)		
10-nov	31. Semaphore		
15-nov	32. Common Concurrency Problems.		Threads (Bugs)
17-nov	36. I/O Devices		
22-nov	37. Hard Disk Drives		
24-nov	39. File and Directories		Disks
29-nov	40. File system Implementation.	P4 Due, P5 File systems	39. File and Directories
1-dic	41. Fast File System / 42. Crash Consistency: FSCK		FS Implement
13-dic	42. Crash Consistency: Journaling		FFS
15-dic	43. Log-structured File Systems	P5 Due	