



Cornell University

# Welcome

## Operating Systems

### CS 4410

Spring 2021

Lorenzo Alvisi & Robbert van Renesse

[Alvisi, Bracy, Schneider, Sirer, van Renesse]

- ◉ We strive to make CS4410/CS4411 welcoming, safe, equitable, and respectful environments, consistent with Cornell's commitments
- ◉ We recognize that the society with live in is none of those things, that we have implicit biases, and that we have to work hard every day to counter those biases to create an inclusive environment
- ◉ If you witness a bias incident or have been the victim of one, please file a confidential report with Cornell
- ◉ If you have any suggestions (improvements to the web site, syllabi, slides etc) feel free to email us at [cs4410-prof@cornell.edu](mailto:cs4410-prof@cornell.edu)

## Emotional Help

Engineering Advising	<a href="http://www.engineering.cornell.edu/resources/advising">www.engineering.cornell.edu/resources/advising</a>	Academic advising for engineering students
Arts College Student	<a href="http://as.cornell.edu/advising">as.cornell.edu/advising</a>	Listing of general support services for a variety of
Gannett	<a href="http://www.health.cornell.edu">www.health.cornell.edu</a>	Cornell University Health Service
CAPS	<a href="http://health.cornell.edu/services/mental-health-care">health.cornell.edu/services/mental-health-care</a>	If you experience emotional stress, contact Counseling and Psychological services
Student Disability Services	<a href="http://sds.cornell.edu">sds.cornell.edu</a>	Ensures that all aspects of student life are accessible, equitable, and inclusive of those with disabilities. Send accommodation letters to Ms. Coralia Torres (ct635), by Feb. 23.

Get help. Get documentation. The earlier, the better  
And look out for each other!



- ◉ Who is Who
- ◉ Before you take this class..
- ◉ Communication
  - Lectures, OHs, FAQ, etc.
  - Getting help
- ◉ Homework, exams
- ◉ Grades & Policies

# About RvR

- ◉ Ph.D. C.S., Vrije Universiteit Amsterdam  
Amoeba Distributed Operating System
- ◉ Industry: Research Scientist @ AT&T Bell Labs  
Unix, Plan 9
- ◉ Serial entrepreneur  
Reliable Network Solutions (IP -> Amazon)  
D.A.G. Labs (acquired by FAST, then by Microsoft)  
Exotanium (ongoing)
- ◉ Interests: scalable and fault tolerant distributed systems
- ◉ Non geek: musician (trad. jazz) swing dance, unicycling

# About Lorenzo

- ◉ Ph.D. C.S., Cornell University  
Fault-tolerant distributed computing  
Physics MS from University of Bologna
- ◉ UT Austin: 1996-2015  
Academy of Distinguished Teachers
- ◉ Serial:  
Consumer of dark chocolate  
Motorcycle owner (4 currently)
- ◉ Interests: scalable and fault tolerant distributed systems
- ◉ Non geek: classical music, literature, hiking, riding



# Who are the TAs?

 Michael Chin CS B. Sc. he, him, his	 Minghao Li CS B. Sc. she, her, hers	 Youer Pu CS Ph.D. she, her, hers	 Xinwen Wang CS Ph.D. he, him, his
 Jack Ding CS B. Sc. he, him, his	 Simon Huang CS B. Sc. he, him, his	 Freya Ryd CS B. Sc. she, her, hers	 Nicholas Zahabiun CS M.Eng. he, him, his
 Ziwei Gu CS M.Eng. he, him, his	 Yu-Ju Huang CS Ph.D. he, him, his	 Nikhil Saggi CS M.Eng. he, him, his	 Jiejun Zhang CS M.Eng. she, her, hers
 Mindy Lee CS M.Eng. she, her, hers	 Kartikay Jain CS B. Sc. he, him, his	 Haocheng Shen CS B. Sc. he, him, his	 Yunhao Zhang CS Ph.D. he, him, his



- ◉ Who is who
- ◉ Before you take this class..
- ◉ Communication  
Lectures, OHs, FAQ, etc.  
Getting help
- ◉ Homework, Exams
- ◉ Grades and Policies

# Prerequisites

- ☉ CS 3410, CS 3420 or equivalent
- ☉ Otherwise (or if in doubt):
  - Come and talk to Robbert or Lorenzo, explain your situation and request permission

# Course Content

- ☉ Four Components
  - Lectures
  - Readings
  - Assignments
  - Exams
- ☉ You are expected to keep up with all four

# Draft Syllabus

- ☉ Introduction
- ☉ Architectural Support for OSs
- ☉ Processes and Threads
- ☉ Synchronization
- ☉ Scheduling
- ☉ Memory Management
- ☉ File systems
- ☉ Networking
- ☉ Security

# Textbooks



- ☉ Free online
- ☉ Can buy PDF or printed copy

# Textbooks

Concurrent  
Programming  
with Harmony

Robbert van Renesse  
Cornell University

- Free online
- Linked from CS4410 web site
-  / new typo
-  / new program bug



- Who is who
- Before you take this class..
- Communication
  - Lectures, OHs, FAQ, etc.
  - Getting help
- Homework, Exams
- Grades and Policies

# Communications

- Largely virtual
  - need for more communication!
  - Web page
  - Lectures
  - FAQ
  - Office Hours
  - Study Groups
  - CMS

# Course Web Page

- Schedule, exam & due dates
- Homework release and due dates
- Slides posted before each lecture
- Evolving

# FAQ & CMSX

- FAQ: [sites.coecis.cornell.edu/cs4410faq/](https://sites.coecis.cornell.edu/cs4410faq/)  
Linked from course web page  
You can post questions here and find answers at some later time
- CMSX: <https://cmsx.cs.cornell.edu>  
Assignments  
Grades & Regrades

# Office Hours

- There are 210 20 minute slots each week  
All days of the week (but Sunday), including evenings  
Slots posted on line
- All OHs are over Zoom  
We'll consider special requests for in-person meetings
- You have to register for a slot  
Please specify NetID and include topic

# Lectures

- Tues/Thurs 1:00–2:15 pm live from Bailey Hall and on Zoom.  
Lectures will be recorded, but attendance in person or on line is expected

# Email

- [cs4410-staff@cornell.edu](mailto:cs4410-staff@cornell.edu): time sensitive matters  
Goes to profs and TAs
- [cs4410-prof@cornell.edu](mailto:cs4410-prof@cornell.edu): sensitive matters  
Goes to Robbert and Lorenzo only

Please no email to personal email accounts



- ◉ Who is who
- ◉ Before you take this class..
- ◉ Communication
  - Lectures, OHs, FAQ, etc.
  - Getting help
- ◉ Homework, Exams
- ◉ Grades and Policies

## Homework

- ◉ Assigned approx. once a week
- ◉ Individualized, fillable PDFs
  - (slight) randomization of problem parameters, multiple choice questions, etc.
  - Fully auto-graded (no TAs involved)
  - Regrade requests due within a week
- ◉ Max. 6 slip days total
  - Max. 2 slip days per assignment
  - 10% deduction/day after that

## Study Groups: Connect and Learn

- ◉ Five/six students per group
  - randomized for each homework
- ◉ Each group is assigned a leader (chosen randomly) and a dedicated TA
  - leader organizes at least one meeting for the group
  - <http://lsc.cornell.edu/studying-together/studying-together-online/>
- ◉ You are allowed to discuss the homework with the members of your study group
  - each student must submit own work though...
  - ...and remain fully responsible for their own submission

## Myths about study groups

- ◉ Study groups work better with one's own friends
- ◉ Study groups may be good for weaker students but not for me!
- ◉ Such concerns and more discussed here
  - <https://academia.stackexchange.com/questions/9393/effectiveness-of-grouping-students-to-balance-skills>

# Study Group Code of Conduct

- Each student should feel safe, welcome, respected
- Participate, but don't dominate
- Be patient
- Respect diverse talents and ways of learning
- Fight your implicit biases

**Learning  
The  
Strategies  
Center**

## Find Study Partners!



Studying with peers is a great way to connect with other Cornell students and is a powerful tool for learning.

Cornell's Learning Strategies Center (LSC) helps match you with study partners.

To learn more, visit the LSC's [Studying Together](#) webpage or scan the code →



Scan the QR code to find out more about Study Partners

or visit <http://lsc.cornell.edu/studying-together/>

[lsc.cornell.edu](http://lsc.cornell.edu)



## Find Project Partners!



Taking a CS class? Looking for a friend? Searching for a partner? If so, this CIS Partner Finding Social is for you!

This is a virtual event where you'll have the chance to meet and interact with other students in your CS classes! This is the PERFECT opportunity to find a partner, so come join us on 2/11

from 7:30 ~ 9:00pm if you're taking CS 1000~2000 level classes,  
from 9:00pm ~ 10:30pm if you're taking 3000+ level classes!

RSVP here:

<http://bit.ly/cisSP21>

## CIS Partner Finding Social

2/11/2021 • 7:30PM-10:30PM ET

All majors and  
gender identities  
welcome!



# Academic Integrity and Honor Code

All submitted work must be your own

OK to discuss concepts with any other student

Student in a study group can submit the same code

Different study groups are not allowed to share code

Violations will be prosecuted

# Exams

- ⌚ Randomized, fillable PDFs
  - Same as homework
  - Exam questions are versions of homework questions
  - Include questions about lectures, homework, books
- ⌚ Must be done individually
  - Open-book
  - Internet as “read-only” resource: don’t ask questions
- ⌚ Cumulative



- ⌚ Who is who
- ⌚ Before you take this class..
- ⌚ Communication
  - Lectures, OHs, FAQ, etc.
  - Getting help
- ⌚ Homework, Exams
- ⌚ Grades and Policies

# Exams

- ⌚ Timed
  - 2 hours from download to submission
  - Can be done any time in 48-hour period
- ⌚ 3% penalty each 15 minutes late, up to 1 hour
- ⌚ Fully auto-graded
- ⌚ Regrade requests due within one week

# Semester Grades

- ⌚ 50% Assignments
- ⌚ 50% exams
- ⌚ No curving
  - CS4410 is not a competition
  - Your grade reflects your learning objectives
    - not how well you compared with others
- ⌚ Weighing of individual assignments and exams TBD

# Practicum: CS4411



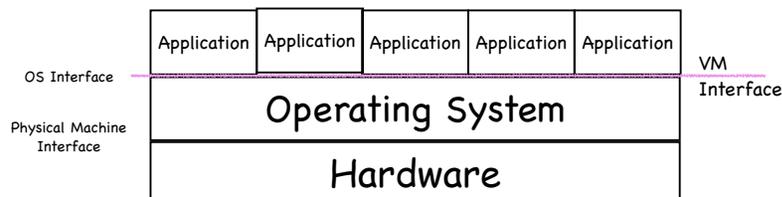
Cornell University

- CS4410 assignments are “small”
- In CS4411, you’re going to have hands-on C development experience with an almost-real operating system: EGOS
  - Write a queue
  - Write a threading package
  - Write a scheduler
  - Write a file system cache
  - Write a file system
- Teams of two programmers

## Introduction CS 4410

### What is an OS?

- An Operating System implements a virtual machine whose interface is more convenient\* that the raw hardware interface



\* easier to use, simpler to code, more reliable, more secure...

## We study a system

# What is a system?

"A complex unit formed of many often diverse parts subject to a common plan or serving a common purpose"

Webster Third New International Dictionary

# What is a system?

interconnections  
components  
"A complex unit formed of many often diverse parts subject to a common plan or serving a common purpose"

a certain behavior at the interface with an environment

Webster Third New International Dictionary

"A set of interconnected components with an expected behavior observed at the interface with its environment"

## Common systems challenges

- ⌚ Emergent properties
- ⌚ Propagation of effects
- ⌚ Incommensurate scaling
- ⌚ Trade-offs

## Emergent Properties

Evident only when components are combined



Millenium Bridge  
London

# Propagation of effects: fighting malaria

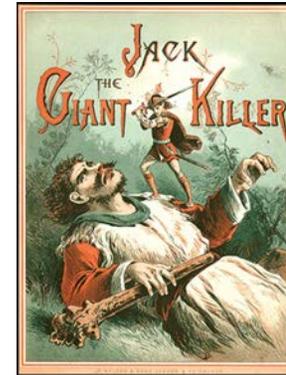


- ☉ WHO sprayed villages in N. Borneo with DDT
- ☉ Wiped out mosquitos, but...
- ☉ Roaches collected DDT in tissue
- ☉ Lizard ate roaches, and became slower
- ☉ Easy target for cats
- ☉ DDT cause cats to die
- ☉ Forest rats moved into villages
- ☉ Rats carried the bacillus of the plague!



# Incommensurate scaling

As the system increases in size or speed, not all components can manage the scale, and things break down



10x higher than Jack!

but also 10x wider and thicker!

About 1000x Jack's weight – but the cross section of the Giant's bones was only 100x Jack's

A human thigh bone breaks at about 10x human weight

The giant would have broken his thighs every time he was taking a step!

On being the right size  
J.B.S. Haldane

# Inevitable Trade-offs

Speed vs power in processors

Bandwidth vs computation in compression

Space vs time almost everywhere

A pawn vs better position in chess

...

# How to Manage Complexity

## ☉ Modularity

Good modularity minimizes connections between components

User 5 (not implemented)

User Programs 4

## ☉ Abstraction

Separate interface from internals; separate specification from implementation

I/O Management 3

Operator Console 2

Memory Management 1

## ☉ Hierarchy/Layering

Constrain interactions so they are easier to understand

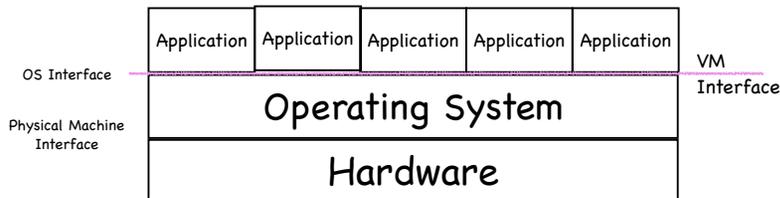
CPU Scheduling and Semaphores 0

Hardware

THE Operating system

# What is an OS?

- ☞ An Operating System implements a virtual machine whose interface is more convenient\* that the raw hardware interface



\* easier to use, simpler to code, more reliable, more secure...



## OS wears many hats

- ☞ Referee
  - Manages shared resources: CPU, memory, disks, networks, displays, cameras...
- ☞ Illusionist
  - Look! Infinite memory! Your own private processor!
- ☞ Glue
  - Offers a set of common services (e.g., UI routines)
  - Separates apps from I/O devices



## OS as Referee

### Resource allocation

Multiple concurrent tasks, how does OS decide who gets how much?

### Isolation

A faulty app should not disrupt other apps or OS

### Communication/Coordination

Apps need to coordinate and share state

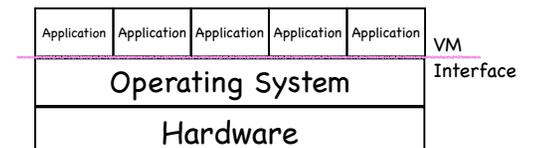


## OS as Illusionist

### Appearance of resources not physically present

#### ☞ Virtualization

Processor, memory, screen space, disk, network





# OS as Illusionist

Appearance of resources not physically present

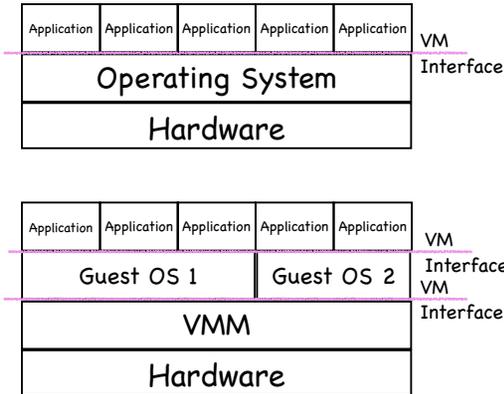
## Virtualization

Processor, memory, screen space, disk, network

The entire computer

Fooling the OS itself!

Eases debugging, portability, isolation



# OS as Glue

## Offers standard services to simplify app design and facilitate sharing

Send/Receive byte streams

Read/Write files

Pass messages

Share memory

UI

## Decouples HW and app development



# OS as Illusionist

Appearance of resources not physically present

## Atomic operations

HW guarantees atomicity at the word level...

What happens during concurrent updates to complex data structures?

What if a computer crashes while writing a file block?

At the hardware level, packets are lost

Reliable communication channels

# What will the course be like?



# Harmony

## Your Automated Concurrency Tutor

```
source.hny X
workspace > source.hny
1 import synch;
2
3 const N = 5;
4
5 def diner(which):
6   let left, right = (which, (which % N) + 1);
7   while choose({ False, True }):
8     P(sema);
9     lock(forks[left]);
10    lock(forks[right]);
11    # dine
12    unlock(forks[left]);
13    unlock(forks[right]);
14    V(sema);
15    # think
16    ;
17  ;
18 ;
19 forks = dict{ Lock() for i in {1..N} };
20 sema = Semaphore(N - 1);
21 for i in {1..N}:
22   spawn diner(i);
23 ;
```

# Issues in OS Design

- ⌚ Structure: how is the OS organized?
- ⌚ Concurrency: how are parallel activities created and controlled?
- ⌚ Sharing: how are resources shared?
- ⌚ Naming: how are resources named by users?
- ⌚ Protection: how are distrusting parties protected from each other?
- ⌚ Security: how to authenticate, authorize, and ensure privacy?
- ⌚ Performance: how to make it fast?

# More Issues in OS Design

- ⌚ Reliability: how do we deal with failures??
- ⌚ Portability: how to write once, run anywhere?
- ⌚ Extensibility: how do we add new features?
- ⌚ Communication: how do we exchange information?
- ⌚ Scale: what happens as demands increase?
- ⌚ Persistence: how do we make information outlast the processes that created it?
- ⌚ Accounting: who pays the bill and how do we control resource usage?