

OS course

Class 1

This course

Agenda for today

- Talk about this course
- Talk about the UNIX OS

English

- As a new regulation from CHE there shall be 2 courses given in English in your B.Sc degree. And it just so happens OS was chosen.
- The English requirement will cover
 - TEST
Yes the test itself is in English.
Furthermore, You must answers to questions in the English language
 - All presentations
 - All homework
 - Everything said in class
 - Syllabus
- The English requirement will not cover recitations
- I speak Hebrew (It is even my mother tongue). You can write to me in English or Hebrew (Or Swedish, Or Bulgarian) in personal communication (such as email, whatsapp) but we will only use English during class. **This is not negotiable.**
- I lived in Plymouth, Devon and in Raleigh, NC and I am fluent in English as well. I also thought OS course in English about 5 times in the past!
- When specific item will require translation to Hebrew (for job interview purposes etc.) I will say it in Hebrew but move back to English afterward.

Me

- Dr. Nezer J. Zaidenberg
- New faculty member
- Thought OS course (or advanced OS courses) more times than I can count (in NYU, TAU, JYU, MTA, HIT (CS + EE versions), Ruppin (CS+EE versions), Shenkar) – at least 30 times.
- This is also my field of expertise and I consult in the industry in the field (To Elta, Rafael, Mafaat, Motorola, IBM, Cisco, Matrix and others)

How to reach me

- Best is probably whatapp 054-5531415 (you can also try calling)
- Email : Scipio@scipio.org
- I can meet students before or after class (Please schedule in advance) or using zoom (Please schedule in advance)
- Tête-à-tête meetings and personal communication can be in Hebrew

Invalid

- I don't look this way but I am an invalid.
- I have balance problems
- Therefore, I sit not stand during lectures.
- This is not out of disrespect toward any of you.



Scribe

- We will hold scribe –
 - Option 1 - Every week I will appoint 3 to summarize what was said in class
 - Option 2 - I will appoint 3 students to summarize all lectures. These students will receive 10 points bonus to the final grade.
- The scribe is a written summary – corrected by me – of all lectures and recitation and can serve you to study for the exam.
- The scribe is 100% correct as far as you are concerned. Even if I somehow missed something in the scribe and you gave that answer in the exam – I will accept the scribe answer as correct.

Recordings

- Two ground rules
 - I stand by everything said in class. I don't object to being recorded.
 - This is a frontal class. Unless there will be an exception all classes are frontal. Not zoom. This is not a hybrid class. (IMHO Hybrid classes don't work)
- Based on Rule 1
 - I will try to set up zoom recording during class (and you can also record using your mobile phones or tape recorder) and upload it ASAP after class.
 - This is BEST EFFORT only. No guarantees.
- Based on Rule 2
 - You may not join Zoom, I will not transfer this class to a hybrid class.
 - If something was not recorded for any reason, it will not serve as a reason to cancel class material etc.

For all practical purposes the
scribe is the ground truth
(So even if I made a mistake in
class and corrected it in the
scribe the recording cannot save
you)

So how do we study to the exam

- Don't use old notebooks (I found many mistake in the course and corrected them!)
- Scribe is best
- There will be guides for every topic
- Stevens' APUE (Stevens' and Rago APUEv2 or 3 preferred) and Robert Love's Linux Kernel development (3e) are the best books for this course both are by A-W.

Prior knowledge

- I assume you all know how to
 - How to use in UNIX shell (ls(1), cd(1), cp(1) etc.)
 - Write a program in C, CPP and compile it under Linux
 - Debug a program under Linux
 - Write simple programs in C or CPP that read write and manipulate files (using FILE *, file descriptors or fstream)
 - Know OOP and OOAD basics in CPP including what is constructor, destructor, polymorphism, inheritance, vTable and how to write simple OOP code in C++.
 - How to write simple networking code in C/C++ (socket(2), bind(2), listen(2), accept(2), connect(2), send(2), recv(2), sendto(2), recvfrom(2), close(2))
- If you don't know any of these **PLEASE TELL ME NOW**

During the classes I will use
vi(1), gdb(1) (As god intended)

This is not a requirement and
any environment will do

Homework

- There will be up to 10 homework exercises.
- Most of them will have bonus work.
- You have to solve 7 for up to 40% of the grade
- You can solve more for bonuses
- I will try to give homework exercise every week
- Firm believer in hard work

My alter Ego →



Grade structure

- Up to 40% homework (that means 40% unless there is a strike or force majeure and the number of exercises is significantly cut)
- 60% Exam
 - Out of which 40% is quiz style questions and 3 are open (20% each)
 - All open questions must be answered in English
 - You have to pass the exam to pass this course
- You can acquire at least 20% as bonuses for answering homework probably more.

Agenda - Course

- We will start with learning processes
- Then we will move to network systems and understand the concept of file descriptors
- We will also learn IPC, memory mapped files, IO multiplexing
- We will learn several design patterns from GoF and POA and how they are implemented in UNIX
- We will learn user threads and synchronization as well as sync patterns (Reactor, Thread poll, Active Object, Pipeline, Guard, Singleton etc.)
- We will learn system programming using modern C++ APIs
- We will learn Linux kernel development (creating kernel modules, sync in kernel using spinlocks and wait queues, reading files from kernel etc.)
- We will learn some more interesting kernel topics (Scheduling (CFS), Block layer, UFS) – time permitting

Organization

- I prefer to start at 15:00 and teach 90 minutes straight (finish on 16:30)
- Do you want a break or to start on 15:10?

Let's get on with the nitty gritty!
A little over simplified history
lesson

1970

MULTICS

Bell labs

UNIX

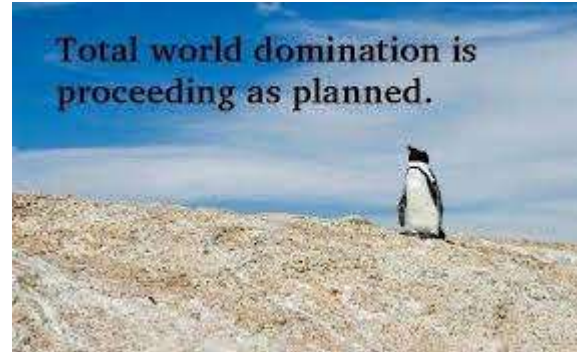
System V vs BSD. Trails

Standard, Multiple Vendors

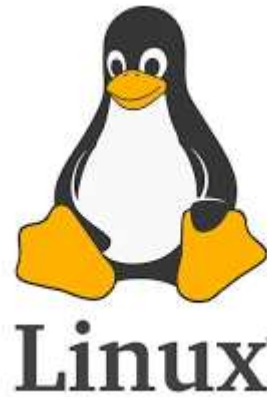


DEC PDP 11

1991



LINUX



Apple - Detour



Not based on Linux

Follow the UNIX standard

Based on BSD Licenses UNIX distribution (Darwin) from Carnegie Mellon university

Many closed APIs

Most of the first half of the course will be identical and work on Mac out of the box

Second half of the course will not work on MAC without VM

I will mention the few exceptions

Printer story

1980s

Free as in beer Free as in speech

FSF and GNU

Applications

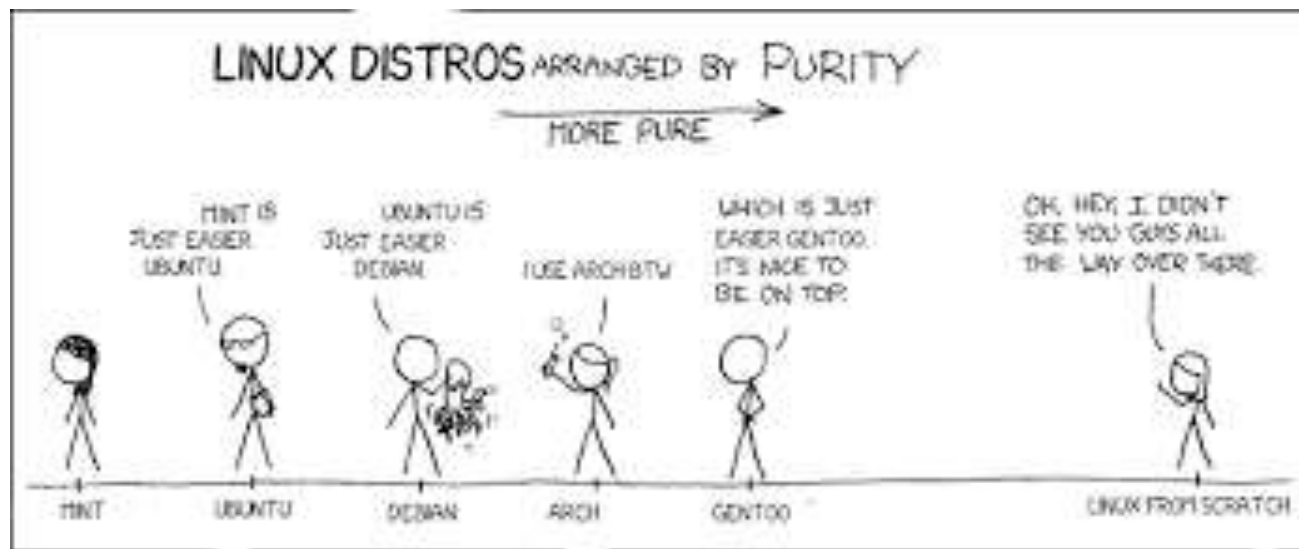
Other groups (Apache,
Mozilla, KDE, Gnome,
etc.)

Linux from Scratch

Parkinson Install

Linux distributions

Redhat
Debian
Kali
Cloudera



SuSE
Mandrake
TurboLinux
Centos

Means humanity to others

Relatively User friendly
and widely used

Ubuntu

6 months release cycle

LTS and STS releases

Server
Desktop
Core (Embedded, IoT)

Free

VirtualBox

What is Emulator

VM vs Emulator

Virtual Machines

Hardware Virtualization

OS/Process <-> Hypervisor OS

VMWare VMPlayer

Class VM

Ubuntu 22.04 LTS

I will provide Ubuntu 22.04 Desktop LTS (With compilers and development tools preinstalled) for both VMWare and VirtualBox

Please make sure one of them works for you.

If you have ARM based Mac computer read manuals on how to run these – For the 2nd half of the course. But you can use Xcode for the first half