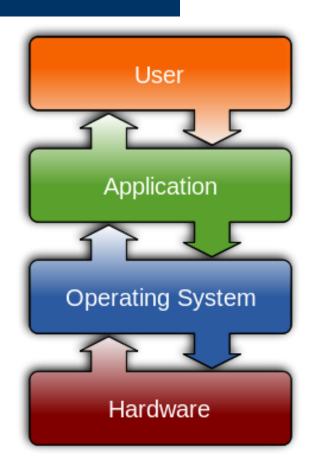
## OS lab summary

- >OS lab review
- >Is computer mysterious?
- >Your Feedback
- Future

## OS lab review

### What is an OS?

- look from outside
  - a bridge connecting application and hardware
- look inside
  - resource manager
    - process, device, memory, file...
  - service provider
    - system call

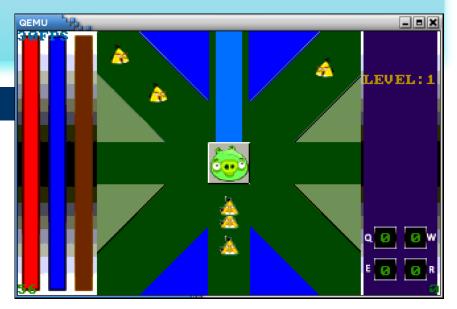


## Guide Question

#### What is an OS? (cont.)

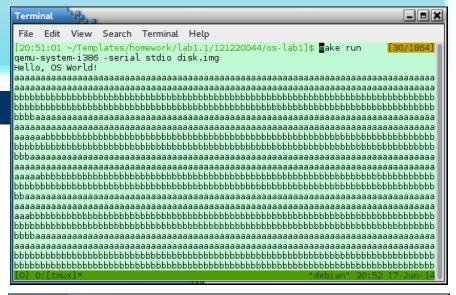
- When a program runs over OS, how do they connect with each other?
- After you issue command to run a "hello world" program, what happen to the OS exactly? - Guide Question 2
- Try to answer two guide questions
  - now
  - at the end of this course

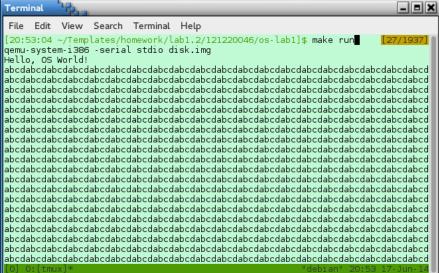
- booting
- interrupt
- I/O



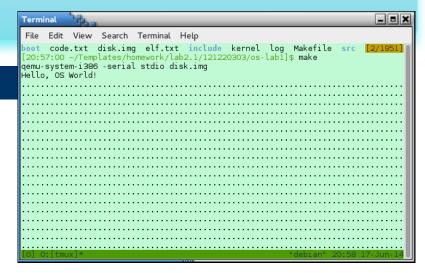


- thread creation
- context switch
- thread schedule
- thread sleep, wakeup





- locking
- semaphore
- message passing
- device driver





- RAMDISK
- process management
- memory management
- system calls

```
Terminal
File Edit View Search Terminal Help
[21:05:18 ~/Templates/homework/lab3.2/121220131/os-lab1]$ make run
qemu-system-i386 -serial stdio disk.img
before init_proc
listing all registed devices:
timer #0, #0
ttyl #1, #0
tty2 #1, #1
tty3 #1, #2
tty4 #1, #3
ram #6, #0
pm #8, #0
mm #9, #0
Hello, OS World!
we are in usr process
```

```
√ tty1
                                                                   2014/06/17 21:07:55
Proc: 13 Hello World!0
                                                                               File Edit View Search Terminal Help
      cr3 37C000
      Capture: 2
      "2" readed!
      pdir 37D000
      dest pid 13
      father pid 12
      cr3 37D000
      offset: 100000
      page allocated 1002
      src mem 1000000, dest mem 1002000
      page allocated 1003
      src_mem 1001000, dest_mem 1003000
      dest cr3 C039C004 src cr3 C039C000
      pde : 73C007
      pte: 23E007
      pa 23E000 dest C023E000
      i[1]: 23EB2C
      go wait.
      here to exec.
      pdir 37D000
      pages left: 28670
      page allocated 1002
       page allocated 1003
```

## Lab4 (is waving to you)

- file operations
- directory-base file system

```
[1]uzh@Nanos /# cd txt
21yzh@Nanos /txt# ls -l
        416
        192
     824581
               big
     867499
               code.txt
       2023
               f1
        919
               fΖ
               f3
     136304
               f 4
      48649
      67533
               malloc free.txt
      68150
               malloc_free111111.txt
       8000
               num1.txt
       9396
               num2.txt
[3]yzh@Nanos /txt# cat f1 | grep void
void print_reverse(void (*)(char), char *, int);
void print_str(void (*)(char), char *);
vfprintf(void (*printer)(char), const char *ctl, void **args) {
void print_reverse(void (*printer)(char), char *str, int end){
void print_str(void (*printer)(char), char *str){
[4]yzh@Nanos /txt#
```

## Any more?

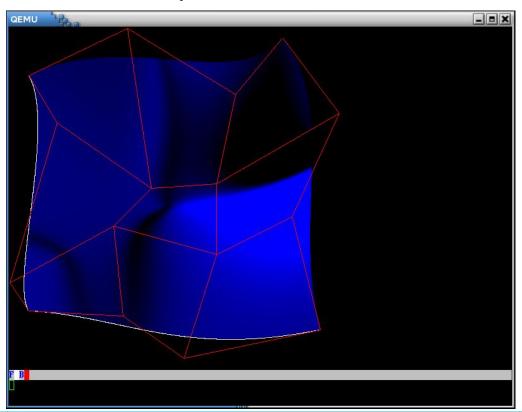
- That's \*ALL\* about operating system.
- The rest is to practice, practice, practice.
- OS lab gives you a chance to look close to
  - an operating system with deep insight
  - the "mysterious" hardware
- It is a link between concept and practice.
  - more impressive than memorizing the textbook

### Hardware

- Hardware is NOT mysterious.
  - register store data
  - address mode & protection mechanism
  - interrupt & exception change the execution flow, must be handled first
  - device receive request command word, then do something

# More

• draw a Bezier surface



## More

- You can add anything you want to your OS.
  - sound card
  - SVGA
  - signal
  - network
  - compiler
  - -
- http://wiki.osdev.org/Expanded\_Main\_Page

• Is computer mysterious?

- from Computer Architecture, Princeton University

**Application** 

Application Requirements

Technology Constraints

- from Computer Architecture, Princeton University

#### **Application**



- from Computer Architecture, Princeton University

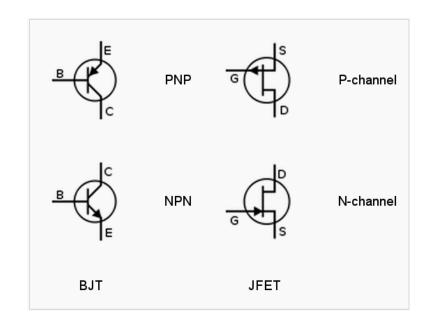
#### **Application**



Devices

- from Computer Architecture, Princeton University

#### Application



Circuits

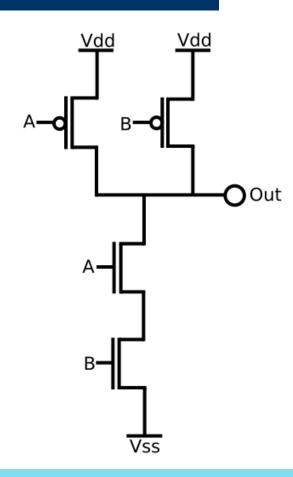
Devices

Physics

- from Computer Architecture, Princeton University

#### **Application**

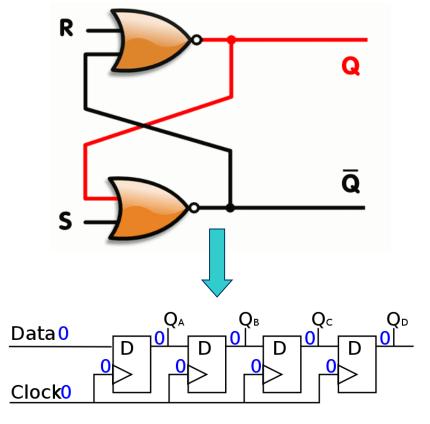
Gates
Circuits
Devices
Physics



- from Computer Architecture, Princeton University

#### **Application**





- from Computer Architecture, Princeton University

#### **Application**

Micro-architecture

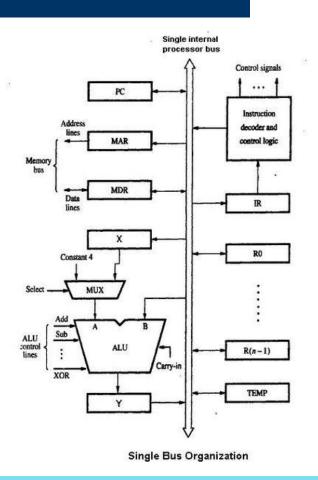
Register-Transfer Level

Gates

Circuits

Devices

Physics



- from Computer Architecture, Princeton University

#### **Application**

#### Instruction Set Architecture

Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices

Physics

#### MIPS32 Add Immediate Instruction

001000	00001	00010	0000000101011110
OP Code	Addr 1	Addr 2	Immediate value

Equivalent mnemonic: addi \$r1, \$r2, 350

- from Computer Architecture, Princeton University

#### **Application**

Operating System/Virtual Machines

Instruction Set Architecture

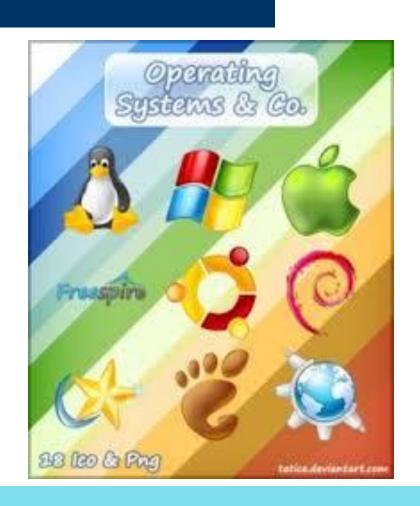
Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices



- from Computer Architecture, Princeton University

#### **Application**

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices



- from Computer Architecture, Princeton University

**Application** 

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

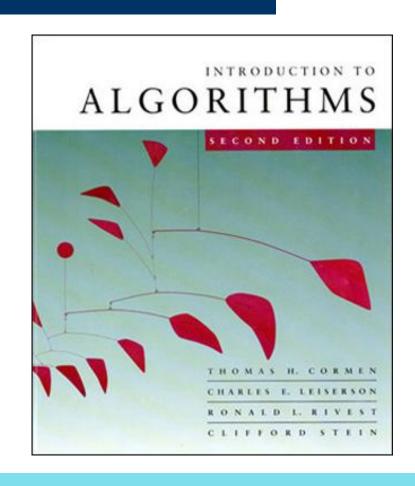
Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices



- from Computer Architecture, Princeton University

**Application** 

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices



## Have you ever wondered...

Why can computer do so many things?



Why can computer do so many things? **Application** 

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Micro-architecture

Register-Transfer Level

we can already do computation here

Gates

Circuits

Devices

- From the aspect of theory
  - deterministic finite automaton gives CPU the power to do computation automatically

**Application** Algorithm Programming Language Operating System/Virtual Machines Instruction Set Architecture Micro-architecture Register-Transfer Level Gates Circuits Devices **Physics** 

- From the aspect of theory
  - ISA makes CPU controllable

- From the aspect of theory
  - pushdown automaton makes it possible to translate source language to ISA

Application			
Algorithm			
Programming Language			
Operating System/Virtual Machines			
Instruction Set Architecture			
Micro-architecture			
Register-Transfer Level			
Gates			
Circuits			
Devices			
Physics			

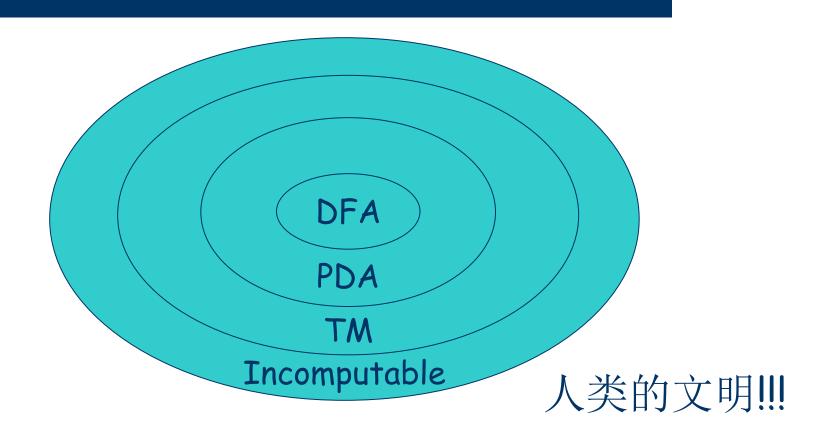
- From the aspect of theory
  - Turing machine gives computer the power to store data, and greatly improves its ability of computation

**Application** Algorithm Programming Language Operating System/Virtual Machines Instruction Set Architecture Micro-architecture Register-Transfer Level Gates Circuits Devices **Physics** 

- From the aspect of theory
  - we solve problems by algorithms, assuming that Turing machine to be the computation model

**Application** Algorithm Programming Language Operating System/Virtual Machines Instruction Set Architecture Micro-architecture Register-Transfer Level Gates Circuits Devices **Physics** 

## The ability of computation



# Your feedback

### Submission

- 26 submissions
  - 22 effective submissions
  - 4 non-effective submssions
  - bonus point =  $((22/N)^2 + (4/N) * 0.01) * 20$

#### **Opinions**

- 因最终没有完成实验感到遗憾: 8
- 难: 6
- 不清楚bug在哪里: 5
- 理论!= 实现: 3
- 动手收获多: 12(5)
- 提供阶段性参考代码: 4
- 理论课和实验课不同步: 3

#### Happiness

- 到驱动那里可以看到自己的一个实验结果,一个 小图标转啊转的时候别提多开心了
- 开始把框架代码跑出来看见网址在屏幕上跑的时候,自己盯着屏幕高兴了好几分钟

- be proud of yourself
  - you are growing, even you fail
  - 闻所未闻 → 一知半解 → 融会贯通

#### Gains

- oslab是门神课
  - C programming language
  - debugging
  - deeper insight into the low-level of computer system
  - searching material

#### Gains

http://weibo.com/1735047885/B3Qtc9is6

毕竟,最终走上体系结构方向的同学并不多;换 句话说,大部分同学学习操作系统课程,其目的 应当是从中\*\*受用\*\*,而非以一个研究者的视角、 全面理解体系结构

- What can you do after the OS course?
  - have deep insight in progam execution
    - small virus, gdb, OJ...

#### Theory v.s. Implementation

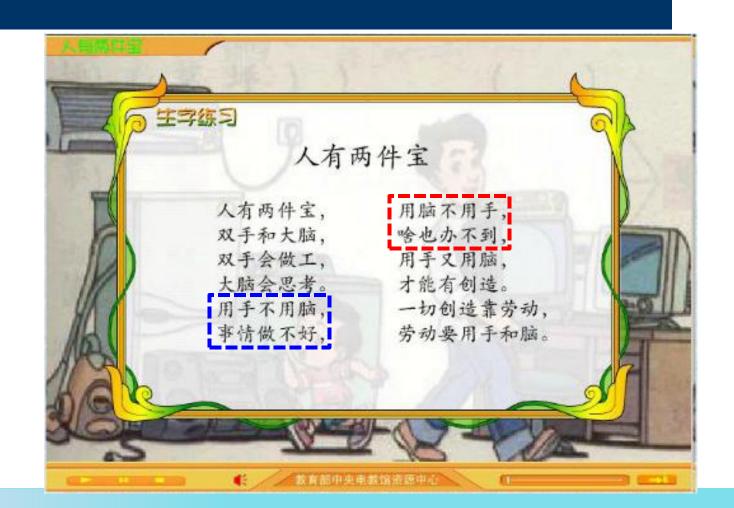
- 实验课的困难主要来自于自己对理论的不熟悉
- 个人感觉这个实验和理论课仍然有些脱节.....因为理论课上讲的东西其实并不和具体实现时遇到的问题相对接,所以这门实验课几乎等于是一门独立于理论课之外的存在
- still no good solution...
  - top-down v.s. bottom-up
  - but implementation requires some related concept learned in the theoratical course

#### 计算思维 v.s. 计算机思维

我觉得像我们这种科研型大学,对于本科生的培养可能主要都是在科研方向上的培养。比如说教一堆数学啊,算法证明啊什么的,对于工程能力教的就比较少。

- 计算思维 abstraction
  - How to model a problem?
- 计算机思维 implementation
  - How to implement a debugger?

#### 计算思维 v.s. 计算机思维



#### 计算思维 v.s. 计算机思维

计算思维

计算机思维的

**Application** 

Algorithm

Programming Language

Operating System/Virtual Machines

Instruction Set Architecture

Micro-architecture

Register-Transfer Level

Gates

Circuits

Devices

Physics

#### System complexity

- Dijkstra算法 → 三角不等式
- locking → ???
  - lock state before CR == lock state after CR
    - CR for critical region
  - 这导致我们必须针对实际的情况...
- That's exactly why system becomes a field.
  - Components are interactive with each others.
  - hard to describe with equations

#### Correctness

- 写了一段时间之后渐渐发现oj跟操作系统实验完全就不是一回事,oj过了,就一切都没了
- 提供标准代码
- pay more attentions to the correctness of your program
  - Testing is one-side-error.
  - 出来混, 迟早要还的

#### Searching materials

- 我希望实验课内容发布的时候可以给我们提供一些参考资料(要具体的,比如某个文档的某几页,某一项技术等等,不要说翻手册,我们毕竟是低年级,没有那么多的时间来积累翻手册的经验)
- That's why you should receive training in this course.
  - 什么时候有时间来积累经验?

#### English materials

- 我恨所有英文教材和资料,我是中国人
- •【百度学术搜索正式上线!】刚过去的"论文季"是否还让你心有余悸?百度学术搜索收录国内外学术站点超过70万家,如知网、万方、维普,外文学术站点acm、IEEE、springer等,共计收录中外文学术资源总量逾2亿,中文超1.5亿,外文超1亿,让学术搜索"一键直达"!
  - 目前百度学术与谷歌学术的差距,大于中美学术的差距

#### Course arrangement

- 理论课和实验课应该分开在两个学期
- 实验课延长为两个学期
- 实验占的分数太少



#### What do you want to be?

- theory
  - Why there are hard problems?
- system
  - How to build a effcient, reliable computer?
- application
  - What can be done by computer?
- "码农"
  - How to realize requirements?
    - usually more closer to the user application

#### Treat them fairly

- 什么是计算?
- 什么是计算机?
- 计算机能做什么?

- They direct 3 fields theory, system, application - respectively.
- Why are people still chasing the answers?

#### To make a better world



#### But why you are here?

Why not "Lan Xiang"?

Try to make a better world!

#### From "有像我一样不会写代码的cser么?"

from bbs.nju.edu.cn

[本篇全文] [回复本文] [本篇作者: lunatic] [本篇人气: 30]

发信人: lunatic (ICS NJU), 信区: D\_Computer标题: Re: 有像我一样不会写代码的cser么?

发信站: 南京大学小百合站 (Fri May 10 23:31:51 2013)

一句话大家心知肚明,要是从进校园开始大作业都独立完成,完成不了就挂科,抄袭就处分,自然也就逼出来了。一步落后步步落后,来上操作系统的人连声明和定义、编译和链接都没明白只记得"以前好像讲过",让你去看IA32手册写操作系统有可能性么?

我们都是活生生的人,从小就被不由自主地教导用最小的付出获得最大的得到,经常会忘记我们究竟要的是什么。我承认我完美主义,但我想每个人心中都有那一份求知的渴望和对真理的向往,"大学"的灵魂也就在于超越世俗、超越时代的纯真和理想一一我们不是要讨好企业的毕业生,而是要寻找改变世界的力量。

#### 宋公语录

学汽车制造专业不是要学开车,是要学发动机怎么制造

 We should have a deep insight into the world of computer science.

## For those who will not choose computer science as your career

- Computer will still be a powerful tool for your work.
- Deeper insight into computer science can still help you make good use of such a powerful tool.

## For those who are not willing to overcome difficulty

但这种程序设计思维不是天生的,对于每个人一开始都是困难的,需要有一个学习、也就是自己独立克服困难的过程。有的同学愿意多花时间克服这个难度,但有些同学面对有难度的东西第一反应就是退缩,想着怎么借助他人的劳动,得过且过。可悲的是,我们目前的本科教育为这种得过且过提供了可能性,因此才有回帖中在目前的体制下仍然过的很滋润而不自知的人出来现身说法(那个很会编程却卖萌的PhD graduate / assistant professor除外)。

教育除了知识的记忆之外,更本质的是能力的训练,即所谓的training。而但凡training 就必须要克服一定的难度,否则你就是在做重复劳动,能力也不会有改变。如果遇到难度 就选择退缩,或者让别人来替你克服本该由你自己克服的难度,等于是自动放弃了获得 training的机会,而这其实是大学专业教育最宝贵的部分。

# For those who are not willing to overcome difficulty

第三个问题:这里尤其要面对那几个出来现身说法的id。也许目前在自己专业基本功方面,这么得过且过的确没有给你们个人造成什么严重的后果,没有影响到你个人眼下的"survive",但这只能证明我们南大的教育系统是存在bug的。这种"只要不影响我现在survive,就不要紧"的想法其实非常的利己和短视:你在专业上的技不如人,迟早有一天会找上来,会影响到你个人职业生涯的长远的发展;更严重的是,这些以得过且过的态度来对待自己专业的学生,他们的survive其实是以透支南大教育的信誉为代价的一一如果我们一定比例的毕业生都是这种情况,那么过不了多久,不但那些混到毕业的学生也没那么容易survive了,而且那些真正自己刻苦努力的学生,他们的前途也会受到影响。

## 每一个你不满意的现在,都有一个你没有努力的曾经-郭斯特







### 每一个你不满意的现在,都有一个你没有努力的曾经-郭斯特







## For those who want to explore the world of computer science

- Every discipline has its "ultimate question".
  - OS how to build efficient, reliable OS?
  - CG how to simulate the real world better?
  - ML what can be learned by machine?
  - TCS why there are hard problems?
  - CT what can be computed?
- Most of them are still unsolved.
  - OS today still have bugs.
  - P = NP?

-

## For those who want to explore the world of computer science

• The world of computer science is fantastic.

OS are only here

**Application** Algorithm Programming Language Operating System/Virtual Machines Instruction Set Architecture Micro-architecture Register-Transfer Level Gates Circuits Devices **Physics** 

#### 站得高,看得远

• 为什么高中生不如牛顿伟大?

• 站在时代的前沿

• 我们学校做得不够好

#### You are at the golden age

- Try to learn as much as possible!
  - golden age: 18~25
- Nothing is impossible!
  - Internet
  - free software
  - MOOC

 Thanks to jyy, thanks to GNU/Linux to let you know ----

# You have the power to change the world!