

# Welcome

元培学院 孙英博

# 内容

- 自我介绍
- ICS与计算机系统的相关研究
- 小班课的内容、评标和助教的自我管理
- 回课、Lab的注意事项
- Linux系统的基本使用；实验的环境配置
- 今天大班课的内容（如果有时间）

# 自我介绍

- （锻炼提问能力：重复上一个人的名字，结合ta的介绍，向ta提一个问题；得到回答后继续自我介绍）
- 姓名 and 年级 and 专业
- 兴趣爱好 and/or 家乡城市 and/or 一道家乡特色小吃 and/or 你经常出现在（校园的）哪个地方 and/or 任何和你有关的有趣的事、经历
- 时间1-2min

# 系统研究talk

# 小班课内容

- 回课 & 补充 & 讨论
  - 在助教回课/补充的时候，可以随时打断！
- 题目练习（希望减少这部分的比例；也想听听大家的意见）
- （私货）： Lab Discussion，先放在这儿，一会儿具体说是什么
- 复习
- 可能的guest lecture
- 安排表格

# 小班课分数构成

- 回课50%
- 出勤20%
  - 无故缺勤每次扣5%
  - 有原因的话提前和助教联系!
- 作业和讨论30%
  - 可能有作业，但选做（不给大家额外工作量）
  - 练习大题时会抽人上黑板做，自告奋勇有bonus分数（加在这部分）
- 总体原则：
  - 减小方差

# 助教自我管理



- 不鼓励任何形式的内卷
  - 例如把精力花在美丽的ppt模板、用英文做ppt或者回课，而非帮助同学们理解知识点
- 时间管理：（一定）不迟到（尽力）不拖堂
- 根据每位同学的反馈，灵活地调整课程内容、讲授方式
  - 匿名反馈：<https://www.wjx.cn/vm/O3IrsSH.aspx>
  - 实名反馈：微信或者邮件
- 答疑：
  - 在微信群（或公共平台）的问题，当天晚10点前答复
  - 私聊的问题没有这个保证（信息太多可能会忘，求提醒）
  - 期中、期末前有机会的话集中答疑

# 回课的注意事项

- 基本要求
  - 时间：建议15min左右，别超过20min
  - 回课/Lab Discussion的同学（最好）要在当周周二晚上8:00（这个时间我有空，周二之前可约）和我讨论。讨论预计20min，本身不计成绩
- 建议
  - 可以参考：往年ppt、教材、大班ppt，但不要直接用别人的ppt
  - 特别鼓励：尝试和书上不同的顺序、内容组织方式，体现自己的理解；
  - 鼓励：互动、**随堂练习**(CSAPP课后习题/往年期中期末题/CMU题)
  - 鼓励：对课本知识拓展，但要保证内容完整；不要拓展的太多
- 评分标准
  - 是否清晰、涵盖了主要内容
  - 有没有自己的理解（欢迎分享，即使理解不准确也**不会**影响成绩
  - 互动质量高、拓展有趣是加分项



# Lab的注意事项

- 环境：一会儿说
- 原则：
  - 红线：学术诚信
    - **严禁抄袭**，autolab有查重——**NO** Github/CSDN, **NO** 同学/学长学姐代码
    - 严禁上网查找思路（但查不出来）
    - 什么是OK的？
      - 抄课本代码
      - 和同学、老师讨论思路（和上网找的区别？有没有和参与课程的其他人**讨论**）
      - 搜索和回答具体问题？
        - 比如怎么用gdb？怎么用装环境？
        - 不是Ctrl + c, Ctrl + v 就能解决问题的那种
    - 帮助你的同学debug，或者接受你同学的帮助

# Lab的注意事项

- 网站 [autolab.pku.edu.cn](http://autolab.pku.edu.cn)
- 课程机 好像还没发
- 本地环境配置：参考信科推送；有问题可以向同学、助教寻求帮助
- 一些建议
  - 仔细阅读文档
  - 和同学或者助教讨论思路
  - 早点开始
  - 利用好grace day
  - 有特殊情况及时找助教

以前信科的课是这样的

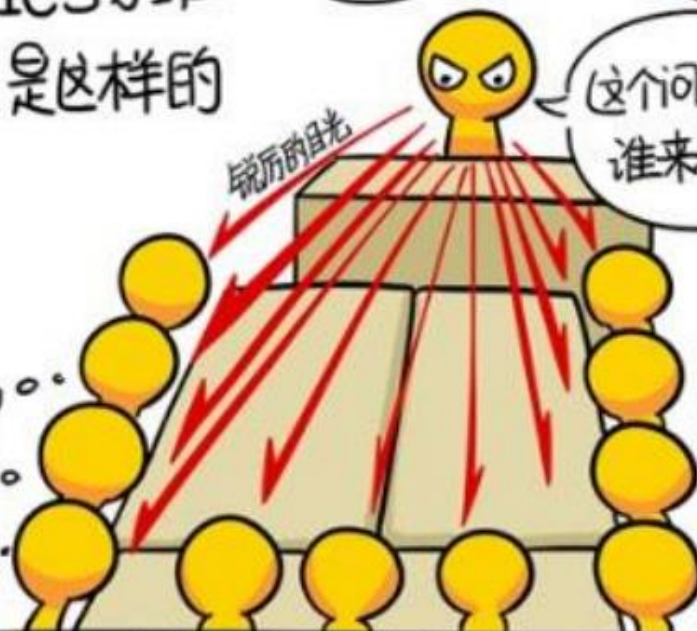
z  
z  
z



一两百号人呢,不可能刚好点到我

好困啊=|||啊!老师在  
看我,还在看我!老师怎么  
一直在看我!再看我,再看我就  
.....妈妈再也不用担心我学习了.

ICS小班  
是这样的



这个问题  
谁来答一下

由于人少,老师一眼望尽全班所有同学

以前作业的头文件是这样的

```
#include <iostream>
```

```
#include <string>
```

```
#include <iomanip>
```

```
#include <cstdio>
```

这是“魔兽世界”大作业的头文件

上了ICS后头文件是这样的

```
#include <assert.h> #include <stdio.h>
```

```
#include <stdlib.h> #include <unistd.h>
```

```
#include <string.h> #include <ctype.h>
```

```
#include <signal.h> #include <sys/types.h>
```

```
#include <fcntl.h> #include <sys/wait.h>
```

```
#include <errno.h>
```

已经基本上看不懂这些头文件了。。



以前周末相遇是这样的 ★



上了ICS后变成了这样





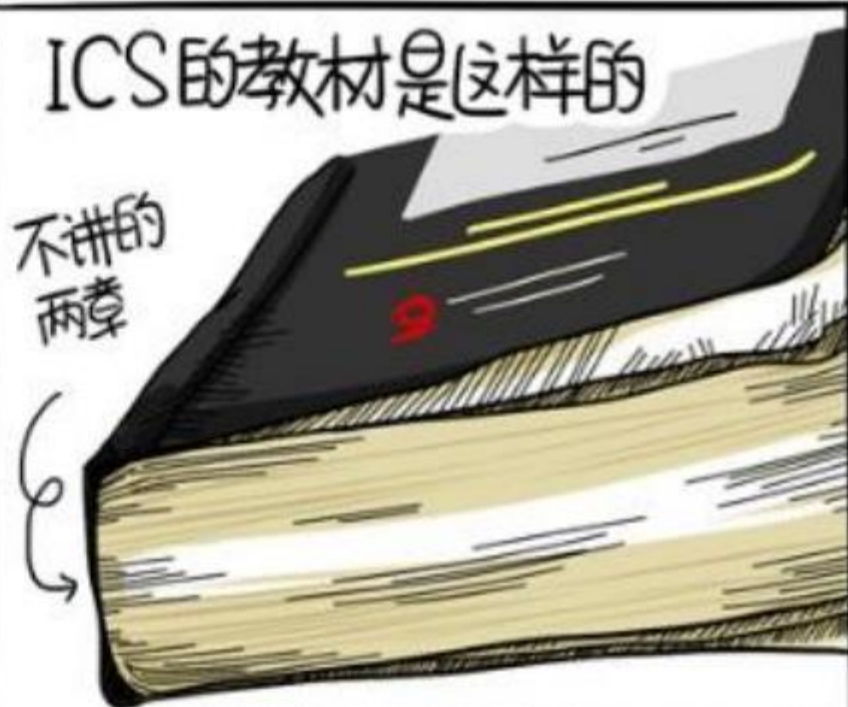
以前教材是这样的

这不是我开学时  
买的《xxxx》吗  
我期末都考完  
了居然一  
页都没看



ICS的教材是这样的

不讲的两章



居然能有教材被我翻阅烂%>.<%



# Linux系统的基本使用

- 演示时间
- 课后（可选的）实践
  - Vim（可选）
  - 权限 [Linux 文件基本属性 | 菜鸟教程 \(runoob.com\)](http://runoob.com)
  - 探索以下命令: more, less, head, tail, tee, find, chown/chmod, which, whoami (用man page即可)
  - 以下命令的以下命令行参数/格式

命令	参数	命令	参数
apt	list,search	gcc	-o, -O1
mkdir	-p	ssh	-i, -p
ls	-l, -a	ssh	.ssh/config文件格式



# 如何连接到课程机

- 好像机器还没准备好
- 利用ssh命令，如ssh -p <port> <username>@<domain/ip>
- 阅读材料
  - [What is SSH \(Secure Shell\)? | SSH Academy](#) – ssh的基本介绍
  - [What is ssh-keygen & How to Use It to Generate a New SSH Key?](#) – 如何省去输密码的过程

# 大班课程回顾

- 本章在计算机系统的位置
- 内存模型（地址空间有限）
- 字长（指针长度）
- 端序（大端、小端）
- 字符串和整数的表示

# 大班课程回顾

- 无符号/有符号整数的类型、存储方式
- 值得注意的常量
  - MAX, MIN
- 类内的运算
  - 单元运算
  - 二元运算
  - 优先级 [C Operator Precedence - cppreference.com](http://cppreference.com)
    - 这个表很多很复杂，我们只看感兴趣的
  - 运算律
  - 不等式的性质
- [C Playground](#) 做实验（或者本地跑当然更好）

# 大班课程回顾

- 类间转换
- 指针，利用指针进行（邪恶的）类型转换

# 练习题1

- 在x86-64机器上运行下列代码，输出是？

```
int main() {  
    unsigned int A = 0x11112222;  
    unsigned int B = 0x33336666;  
    void *x = (void *)&A;  
    void *y = 2 + (void *)&B;  
    unsigned short P = *(unsigned short *)x;  
    unsigned short Q = *(unsigned short *)y;  
    printf("0x%04x", P + Q);  
    return 0;  
}
```

# 练习题2

在x86-64机器上运行下列代码，输出是？（'\0'=0x30）

```
int main() {
    char A[12] = "11224455";
    char B[12] = "11445577";
    void *x = (void *)&A;
    void *y = 2 + (void *)&B;
    unsigned short P = *(unsigned short *)x;
    unsigned short Q = *(unsigned short *)y;
    printf("0x%04x", Q - P);
    return 0;
}
```

# 练习题3

在x86-64机器上，对任意的整型x和y值，ux和uy分别为其转化成无符号数的值，则下面等价的是（不成立的给出反例）：

<code>x &gt; y</code>	<code>ux &gt; uy</code>
<code>(x &gt; 0)    (x &lt; ux)</code>	<code>1</code>
<code>x ^ y ^ x ^ y ^ x</code>	<code>x</code>
<code>((x &gt;&gt; 1) &lt;&lt; 1) &lt;= x</code>	<code>1</code>
<code>((x / 2) * 2) &lt;= x</code>	<code>1</code>
<code>x ^ y ^ (~x) - y</code>	<code>y ^ x ^ (~y) - x</code>
<code>(x == 1) &amp;&amp; (ux - 2 &lt; 2)</code>	<code>(x == 1) &amp;&amp; (!!ux - 2) &lt; 2</code>

# 练习题4

2. 在采用小端法存储机器上运行下面的代码，输出的结果将会是？

(int, unsigned 为 32 位长, short 为 16 位长, 0~9 的 ASCII 码分别是 0x30~0x39)

```
char *s = "2018";  
int *p1 = (int *)s;  
short s1 = (*p1)>>12;  
unsigned u1 = (unsigned) s1;  
printf("0x%x\n", u1);
```

A) 0x00002303   B) 0x00032303   C) 0xffff8313   D) 0x00008313