

# C/UNIX程序设计

## Course Project

汪玉

办公室: 罗姆楼 4-303

[yu-wang@tsinghua.edu.cn](mailto:yu-wang@tsinghua.edu.cn)

# 课程回顾

---

- UNIX历史与简介
- C语言复习，开发工具简介
- 文件系统
- 进程与内存
- 进程间通信
- 网络与套接字

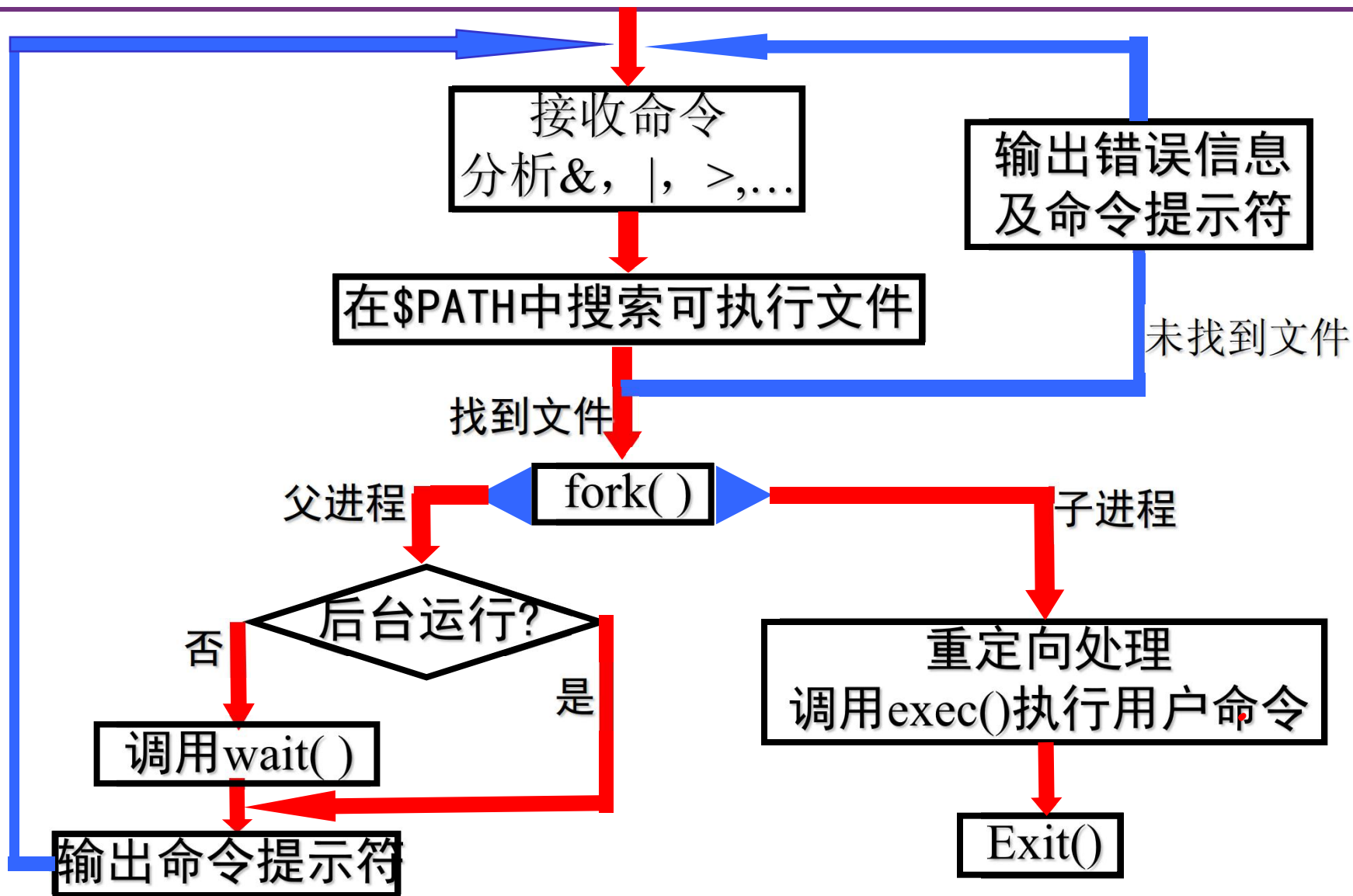
# 课程项目

- 基本要求
  - ▶ 能实现一个系统的实际应用
  - ▶ 具有一定的难度、综合性和工作量
  - ▶ 参考选题或自主选题
- 练习目的
  - ▶ 感受一下实际项目开发
  - ▶ UNIX编程思想实践

# 选题1: yaush

- Yet Another Unix Shell
- 实现一个命令行解释器，也就是常说的Shell
- 功能要求
  - ▶ 用户输入命令与参数，能正常执行命令
  - ▶ 输入、输出重定向到文件
  - ▶ 管道
  - ▶ 后台执行程序
  - ▶ 作业控制(jobs, bg, fg)
  - ▶ 历史命令(history)
  - ▶ 文件名tab补全，各种快捷键
  - ▶ 环境变量、简单脚本

# 回忆



# 基本流程

---

- 获得用户输入
- 命令解析
- 命令执行

# 获得用户输入

- `while(c=getchar())`
- GNU readline
  - ▶ 行编辑库
  - ▶ 各种快捷键支持、Tab补全等
- libedit
  - ▶ 类似GNU readline

# 命令解析

- 词法分析(lexer)与语法分析(parser)
- 词法分析
  - ▶ 把用户输入的字符串转换为单词(token)列表
  - ▶ 可以使用有限状态机实现
  - ▶ 也可以使用lex等专门的词法分析工具
- 语法分析
  - ▶ 根据token列表构建抽象语法树(Abstract Syntax Tree, AST)
  - ▶ 为进一步执行命令做准备
  - ▶ 语法检查等

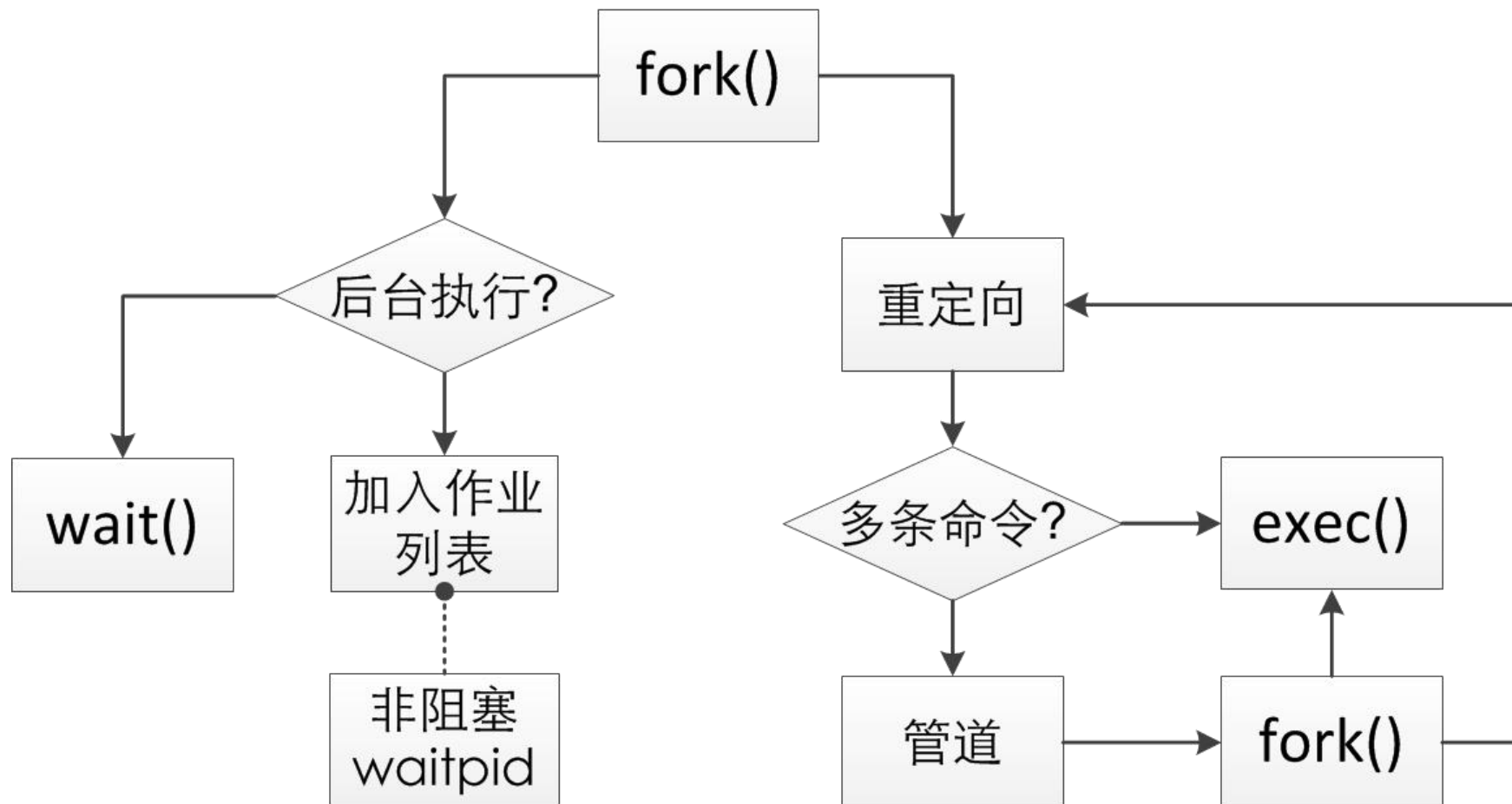


# 例

```
ls -l | wc > out.txt
```

- 词法分析
  - ▶ ‘ls’, ‘-l’, ‘|’, ‘wc’, ‘>’, ‘out.txt’
- 语法分析
  - ▶ {cmd: [‘ls’, ‘-l’],  
stdout: pipe } ->  
{cmd: [‘wc’],  
stdin: pipe,  
stdout: ‘out.txt’}

# 命令执行



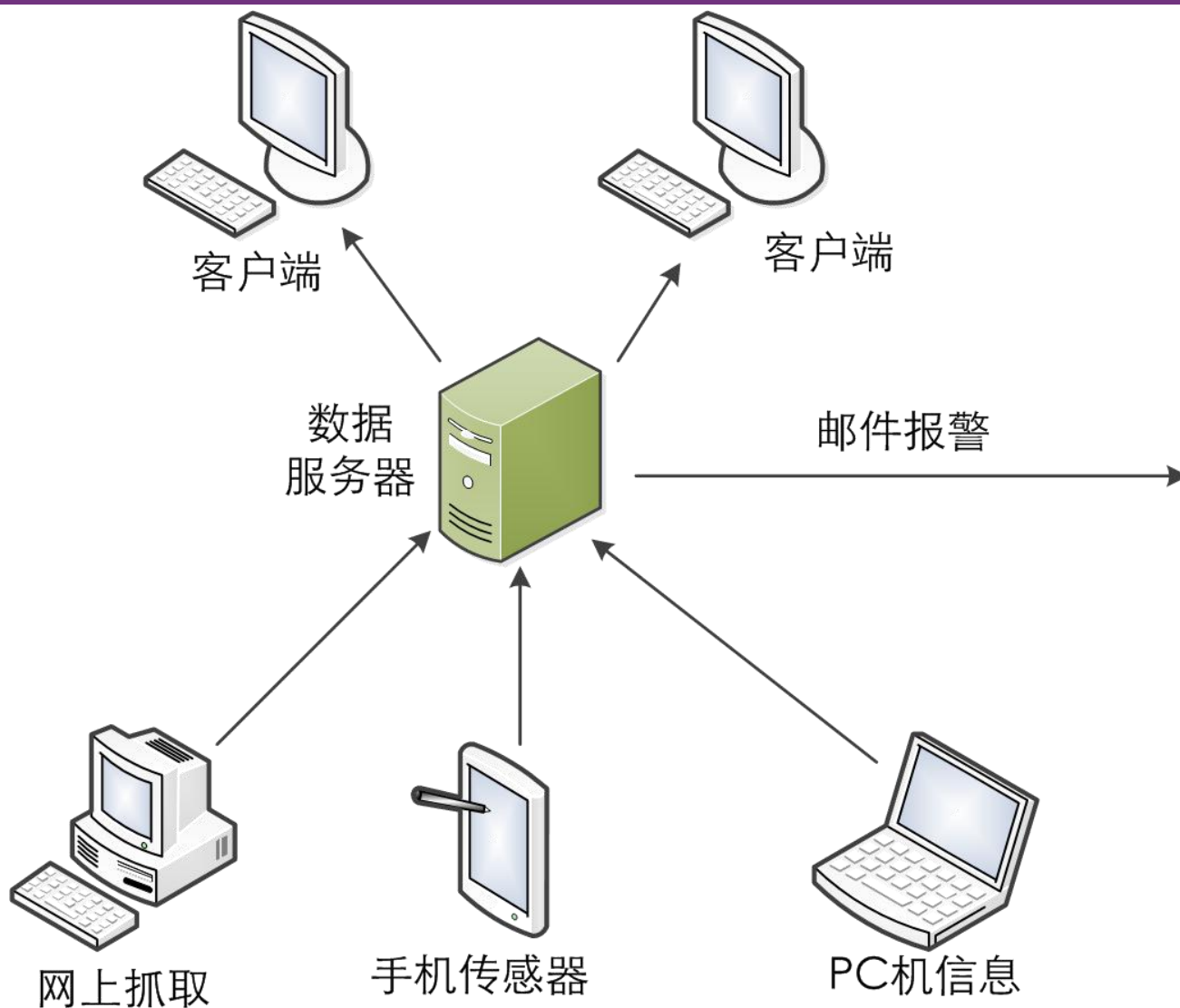
# 要求

- 不得使用shell编写！
- 建议使用C
- 可以自定义语法规则，不一定使用经典shell语法
- 代码结构清晰，注释详细
- 1人独立完成

# 选题2: 传感监控系统

- 使用任意设备获得某种传感数据
- 通过网络发送给中心机
- 使用Web或客户端展示数据
- 数据分析、邮件报警

# 整体结构



# 数据获取

- 真·传感器板
  - ▶ 温度、加速度
- 手机
  - ▶ 加速度
  - ▶ 地理位置
- PC机
  - ▶ CPU温度
  - ▶ 内存占用
  - ▶ 网络延迟、带宽、丢包率
- 网络抓取
  - ▶ 天气信息
  - ▶ PM2.5

# 数据汇总

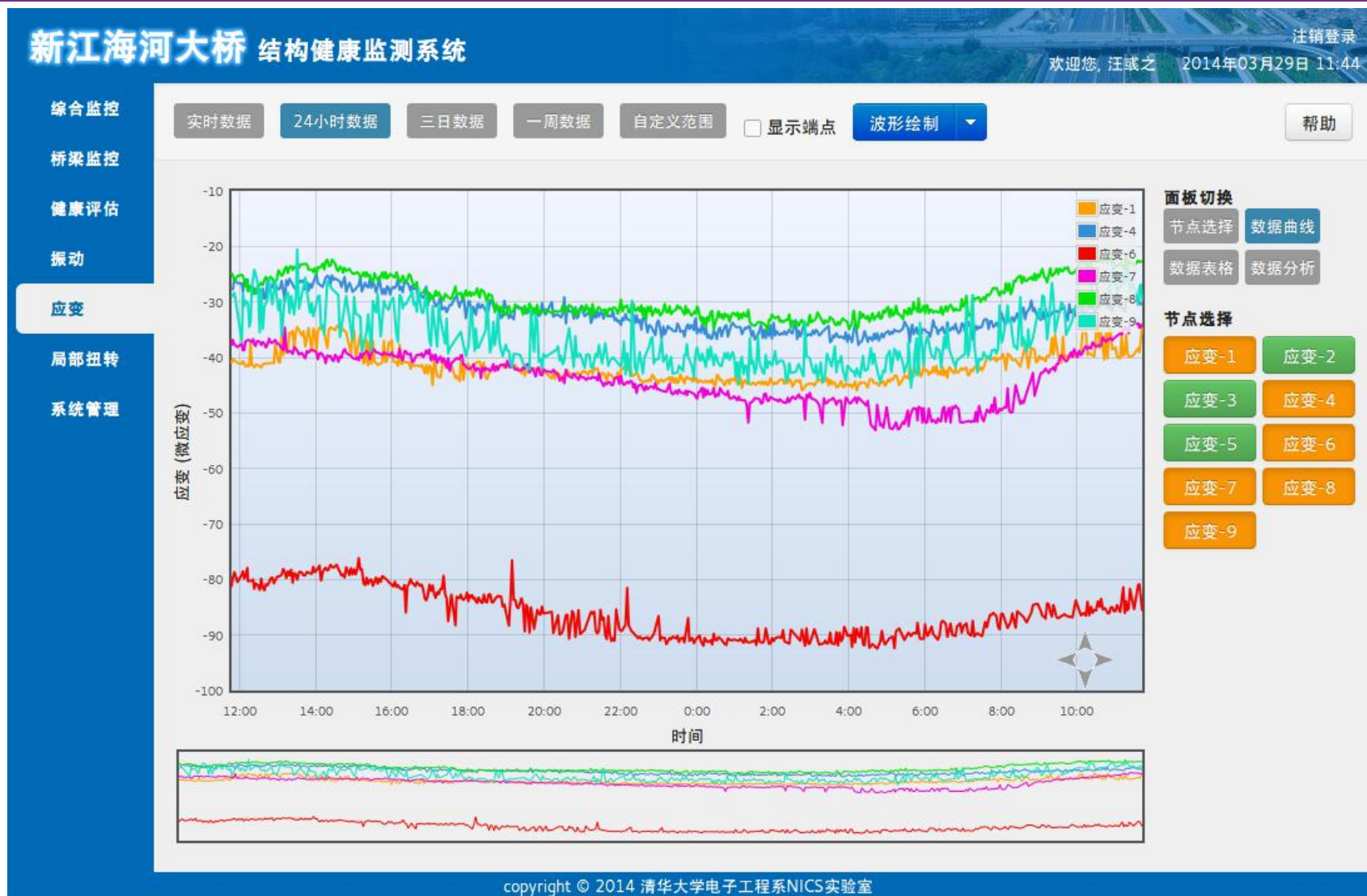
- socket服务器
- 自定义协议，或使用zeromq等消息库
- 支持多传感器、多客户端
- 时间序列
- 传感数据存放在数据库中

# 客户端

- 设计过的UI
- 自定义协议，从服务器获取数据
- 绘图、表展示数据
- 可以基于WEB
  - ▶ 服务端使用某种Web框架, RESTful API
  - ▶ HTML5, SVG, JSON, D3.js, AngularJS ...
- 可以基于传统客户端
  - ▶ QT、GTK
  - ▶ NodeWebkit



# 客户端



# 基本要求

- 至少3个进程（数据采集、汇总分发、客户端）
- 尽量使用多台机器/手机/开发板
- 不一定使用C语言
- 建议尝试一些fashion的框架、协议、编程语言等

# 选题3: 卷积神经网络运行框架

用 tensorflow/pytorch/mxnet 等框架进行训练

- 要求实现

- ▶ 从训练好的(或格式转换过的)模型中加载参数
- ▶ 读取、展示图片(可以使用OpenCV)
- ▶ 实现基本运算单元并运行网络
  - | 可使用blas等数值计算库
  - | 实现全连接层、卷积层、pooling层
  - | 核心原理: 实现矩阵计算  $y = \sigma(Wx+b)$
- ▶ 实现写数字(MNIST数据集)的识别, 完成demo

# 选题x： 自主命题

- 要有一定的工作量
  - ▶ 多个模块设计
  - ▶ 代码量1500行以上（不包括html/css）
- 要有一定的实用意义
- 最好能与自己的研究内容有交集

# 作业提交

- 提交内容
  - ▶ 代码、说明文档（设计报告）
  - ▶ 小组合作的，提交一份代码，每人独立写作报告，并说明自己完成了哪部分工作
- 提交方式
  - ▶ 使用github
- 基本安排
  - ▶ 一周内完成选题，在网络学堂提交 github 仓库
  - ▶ 1月10日左右最终展示，期末前(待定)交报告
- Tips
  - ▶ 助教会通过github提交日志确认各位的工作量
  - ▶ 不建议赶deadline，助教在github上看着你们