C/UNIX程序设计

Course Project

汪玉
办公室: 罗姆楼 4-303
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

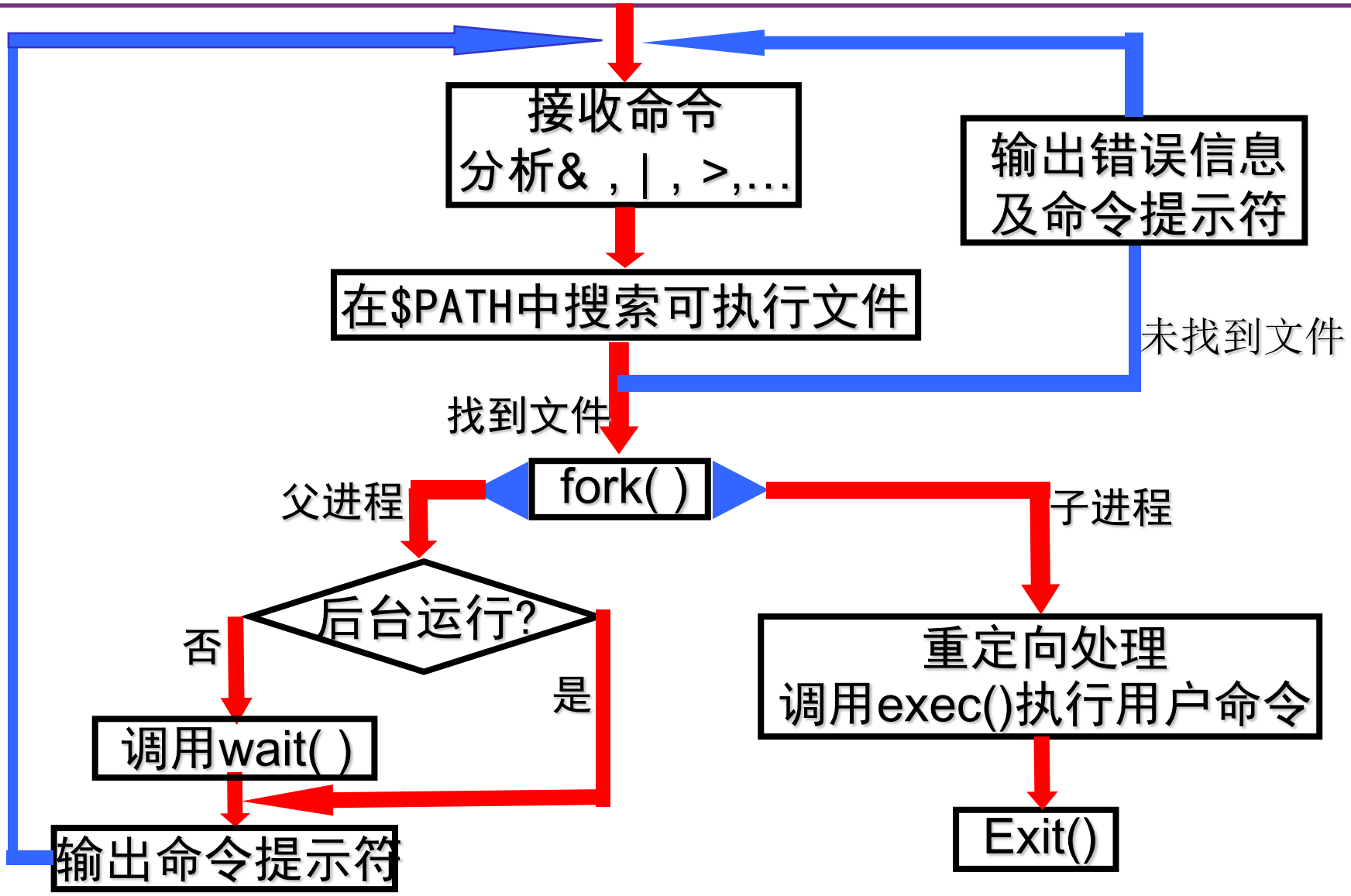
课程项目

- 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


接收命令
分析&，|，>,...
在$PATH中搜索可执行文件
输出错误信息
及命令提示符
未找到文件
找到文件
fork( )
父进程
子进程
后台运行?
否
是
调用wait( )
输出命令提示符
重定向处理
调用exec()执行用户命令
Exit()

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

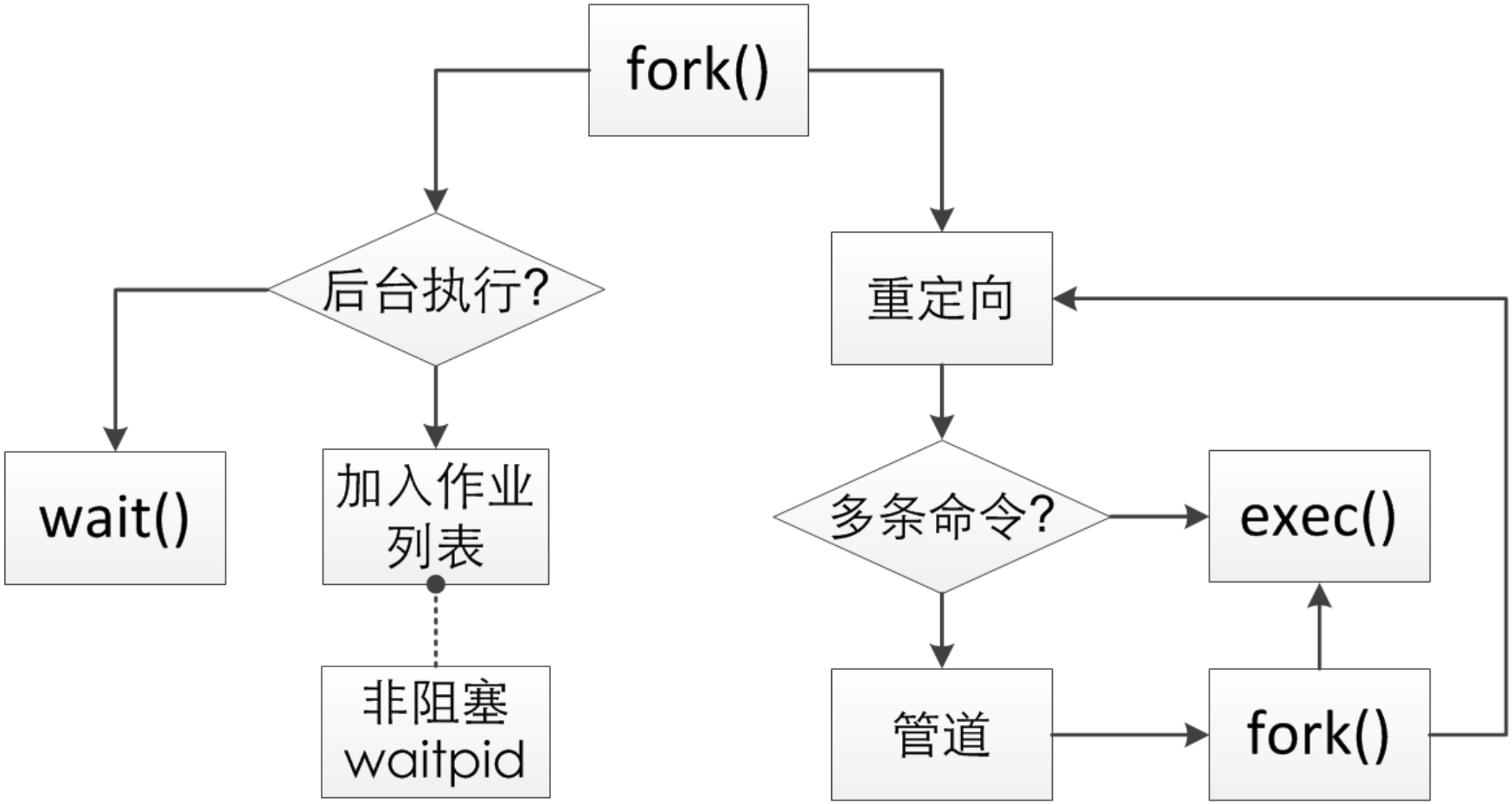
- 词法分析(lexer)与语法分析(parser)
- 词法分析
  - 把用户输入的字符串转换为单词(token)列表
  - 可以使用有限状态机实现
  - 也可以使用lex等专门的词法分析工具
- 语法分析
  - 根据token列表构建抽象语法树(Abstract Syntax Tree, AST)
  - 为进一步执行命令做准备
  - 语法检查等

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



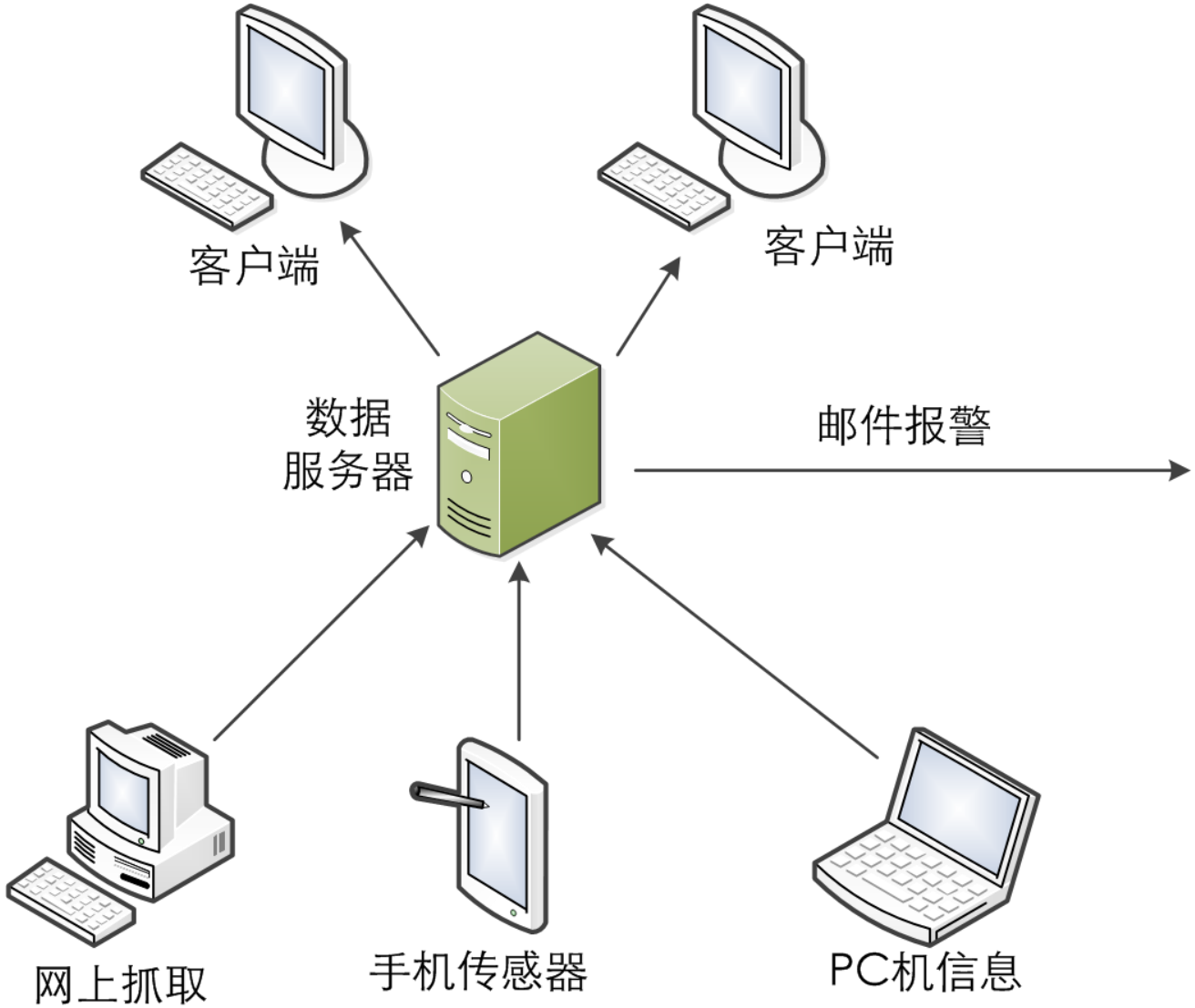
要求

- 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

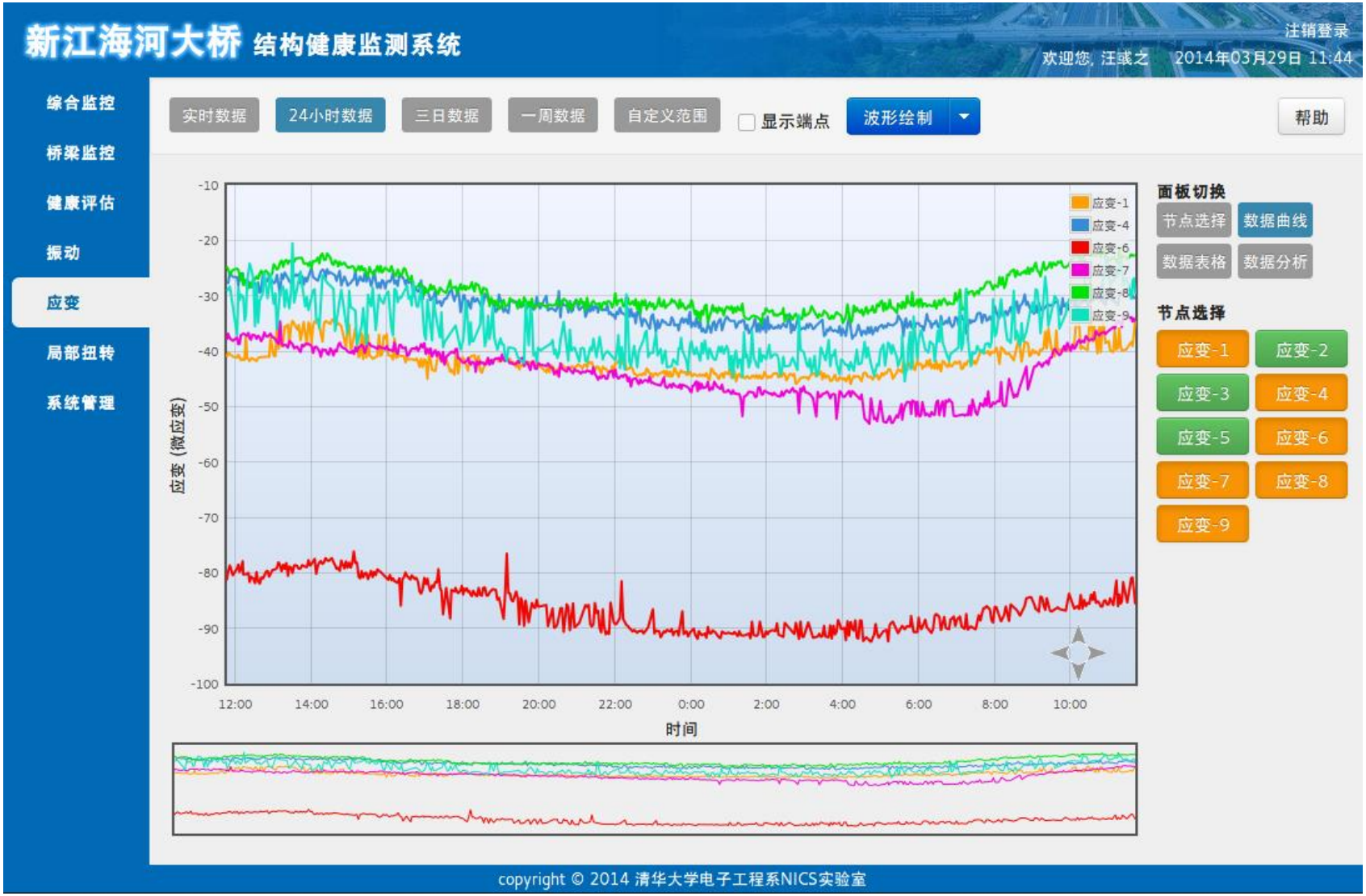
数据汇总

- 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


新江海河大桥 结构健康监测系统
注销登录
欢迎您, 汪彧之 2014年03月29日 11:44
综合监控
桥梁监控
健康评估
振动
应变
局部扭转
系统管理
实时数据
24小时数据
三日数据
一周数据
自定义范围
显示端点
波形绘制
帮助
应变-1
应变-4
应变-6
应变-7
应变-8
应变-9
应变 (微应变)
-10
-20
-30
-40
-50
-60
-70
-80
-90
-100
12:00
14:00
16:00
18:00
20:00
22:00
0:00
2:00
4:00
6:00
8:00
10:00
时间
面板切换
节点选择
数据曲线
数据表格
数据分析
节点选择
应变-1
应变-2
应变-3
应变-4
应变-5
应变-6
应变-7
应变-8
应变-9
copyright © 2014 清华大学电子工程系NICS实验室

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
- 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
- 可以找助教借Arduino

选题4: 数字逻辑电路仿真

- 定义一系列基本逻辑器件
  - 与、或、非、异或、与非、或非……
  - 触发器、时钟
- 功能定义
  - 用户通过声明逻辑器件及其连接关系得到一个电路
  - 运行simulate函数仿真电路
  - 可以定义器件延迟
  - 输出
    - 电路事件
    - 运算结果
    - vcd格式的时序图

基本思路

- 定义wire类

```
class Wire {
public:
    Bit value;
    Component  in, out;
    void setValue(Bit value, bool isPar);
}

void Wire::setValue(Bit value, bool isPar) {
    this->value = value;
    if (!isPar) this->out->trigger(isPar);
}
```

基本思路

- 非门

```
class Inverter public Gate {
public:
    Inverter(Wire in, Wire out) {
        in = in; out = out;
    }
    Wire in, out;
    void trigger(bool isPar);
}

void Inverter::trigger(bool isPar)  {
    this.out.setValue(~this.in.value, isPar);
}
```

基本思路

- 与门

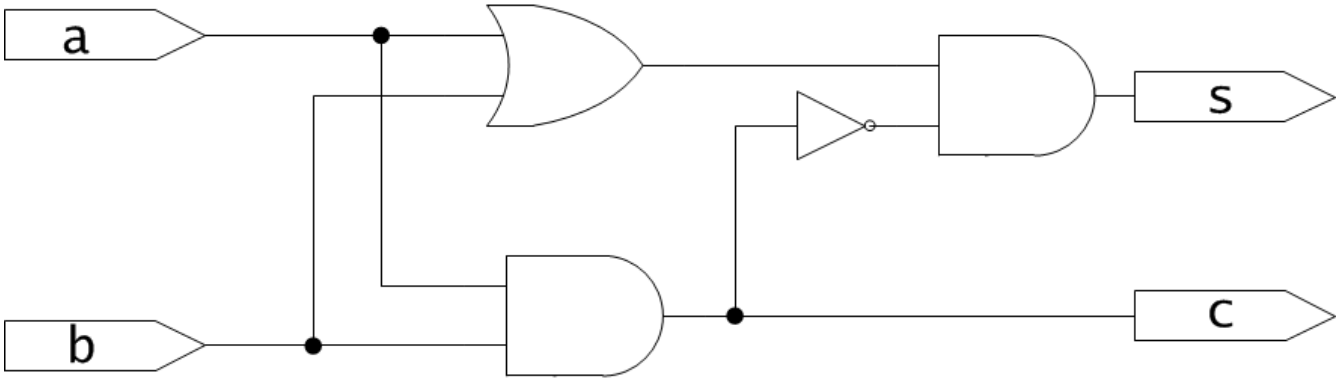
```
class AndGate public Gate {
public:
  AndGate(Wire in1, Wire in2, Wire out) {
      in1 = in1; in2 = in2, out = out;
  }
  Wire in1, in2, out;
  void trigger(bool isPar);
}

void AndGate::trigger(bool isPar)  {
  Bit v1=this.in1.value, v2=this.in2.value;
  this.out.setValue((v1 & v2), isPar);
}
```

基本思路

- 组合逻辑
  - 定义信号线
  - 定义逻辑门
  - 指定输入信号，获得输出信号
- 延迟仿真、时序电路
  - 定义时间片变量cur_time
  - 维护事件队列，每个事件有发生时间和需要执行的操作

例




```
wire a, b, s, c;
wire d, e;
AndGate(a, b, c);
OrGate(a, b, d);
Inverter(c, e);
AndGate(d, e, s);
```

要求

- 至少实现带延迟的组合逻辑
- 建议使用高级语言编写
- 妥善处理串行赋值与并行赋值
- 若要处理时序逻辑，建议2人组队完成

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，助教会告知向哪个仓库推送
- 基本安排
  - 下周一之前写邮件告诉助教自己的选题、github帐号
  - 6月30日最终展示，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