|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **组别** | **系统架构（采用的编程语言、是否多线程、是BS/CS/Android哪种架构）** | **课程信息管理** | | **课外信息管理** | | **课程导航** | | **模拟系统架构、模拟时钟推进思想** | **日志功能** | **选做功能一（功能简介、核心算法描述）** | **选做功能二（功能简介、核心算法描述）** |
| **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** | **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** | **完成的基本功能描述** | **核心算法描述、算法优缺点及性能** |
| **16** | **1纯 C++ 语言编写**  **2使用cmake构建项目**  **3本地单机运行**  **4支持多用户身份**  **5支持多用户登录使用** | ***学生：***  **1提交课程作业**  **2提交课程资料**  **3下载课程资料**  **4作业名称查询**  **5作业成绩排序**  **6资料名称查询**  **7资料权重排序**  **8课程名称查询**  **9课表课程查询**  **10课程时间查询**  ***老师：***  **1发布作业**  **2查看作业**  ***管理员：***  **1发布课程**  **2修改课程** | ***算法1：***  **快速排序**  **封装实现可复用的快排，复杂度O(nln n)**  ***算法2：***  **模糊查找**  **实现时间复杂度为O(klnx)，空间复杂度O(x)**  ***算法3：***  **哈夫曼编码实现压缩和解压缩。**  ***算法4:***  **Aho–Corasick algorithm(AC自动机)实现近似线性复杂度的作业查重算法。** | **学生：**  **1活动时间查询**  **2活动名称查询**  **3进入活动菜单（进入后可以对个人活动增删改)**  **4活动闹钟提醒**  **5活动类别查询，并且按照时间进行排序**  **管理员：**  **发布和修改班级活动** | ***算法1:***  **二叉搜索树：**  **活动时间统一用二叉搜索树进行查找，效率高，时间复杂度O（ln n)**  ***算法2：***  **多线程：闹钟的管理线程时刻比较当前模拟时间与活动时间，进行时间监测和提醒。**  ***算法3：***  **冲突检测：通过对区间边界和中点坐标的数量关系侦测冲突。** | ***导航方式:***  **课程名称导航**  **上课时间导航**  **上课地点导航**  **多交通方式**  **支持跨校区**  ***导航策略:***  **最短路径策略**  **最短时间策略**  **交通工具策略**  **自动检测跨校区规划及导航**  **地图随机生成道路拥挤度，便于导航策略的选择。** | ***算法1:***  **Floyd算法,预处理所有导航需要地图的最短路，包括有拥挤度以及自行车道的全源最短路，优点是不用每次都求最短路，查询复杂度O(N)，缺点是地图建筑修改后需要重新预处理一次生成地图数据。**  ***算法2(选做算法):***  **基于Floyd算法以及记忆化搜索实现经过固定点的最短路,复杂度O(K!+N^3),K为固定点个数，N为图中点总数。** | **时钟推进功能:**  **1系统10s等于真实的一小时。**  **2暂停:**  **暂停状态下闹钟进程失效，恢复后用当前时间减去暂停总时长，系统保留暂停前的状态。**  **3快进:**  **记录快进开始时刻，之后将时间差分成快进前部分和快进后部分，分别乘以不同的倍率，再与登录时刻求和作为当前模拟时刻。** | **记录了操作的用户，事件（具体到事件的类别），时间和用户类别，全部记录入log日志文件中。** | **实现经过固定点的最短路规划，基于Floyd算法以及记忆化搜索.**  **此问题worst case为 NP 完全问题，算法使用 Floyd算法预处理一次全源最短路，之后枚举经过点的顺序排列更新答案，记录路径最后输出最短路的规划方案。**  **算法思路借鉴 stackoverflow** [**此作者回答**](https://stackoverflow.com/questions/222413/find-the-shortest-path-in-a-graph-which-visits-certain-nodes)**。** |  |