# 毕业设计过程记录册（2025年4月13日更新）

## 一、上周计划完成情况（2025.4.6-2025.4.12）

### 1. 路径解析增强

* 完成了get\_full\_library\_path()函数优化，现已支持完整的LD\_LIBRARY\_PATH环境变量搜索路径
* 实现了路径解析缓存机制，通过static std::map<std::string, std::string> path\_cache大幅减少重复解析
* 添加了对软链接的自动分析和真实路径解析，使用realpath()函数确保获取物理路径

### 2. 事件处理改进

* 实现了基于<PID, 路径, 时间戳>三元组的事件去重机制，使用哈希表过滤重复事件
* 优化了特殊标记处理逻辑，重构了handle\_event()函数使代码更清晰
* 增加了事件过滤器配置，支持根据进程名、库名等条件进行精确过滤
* 为dynamic\_linking\_event结构添加了事件类型枚举，明确区分不同链接阶段

### 3. 依赖分析优化

* 重写了get\_library\_dependencies()函数，使用libelf库直接解析ELF文件，替代了原先依赖ldd命令的方式
* 添加了依赖树缓存机制，避免重复解析相同库文件
* 实现了递归解析，可以获取完整的依赖树结构而非仅一级依赖

### 4. 关键路径优化

* 使用std::string\_view替代部分字符串拷贝操作，提高性能
* 对format\_timestamp()函数进行了无锁实现，减少线程竞争
* 优化了字符串处理路径，减少不必要的内存分配和复制

### 5. 内存管理

* 为loaded\_libraries映射表实现了LRU清理机制，自动清理长时间未访问的条目
* 在关键路径上预分配事件处理所需内存，避免频繁的动态内存分配
* 改进了JSON输出部分，使用流式处理减少内存占用

## 二、本周计划（2025.4.13-2025.4.19）

### 1. 完善符号跟踪功能

* 增强trace\_dlsym和trace\_dlsym\_return函数实现，捕获更多符号解析细节
* 添加符号映射表，关联符号名称与其定义位置
* 实现懒加载符号解析的跟踪能力

### 2. 重定位追踪功能

* 新增对ELF重定位表的解析和跟踪
* 实现对不同类型重定位（GOT/PLT）的区分和记录
* 增加重定位时间统计功能，分析不同重定位类型的性能特征

### 3. 性能优化

* 重构BPF程序结构，减少内核态与用户态的数据传输量
* 实现采样率控制机制，支持高负载情况下的选择性事件捕获
* 优化perf缓冲区管理，减少事件丢失率

### 4. JSON输出增强

* 扩展JSON结构，包含更多链接过程的详细信息
* 添加时间线生成功能，用于可视化动态链接过程
* 实现增量式JSON生成，支持实时数据导出