1. 上周计划完成情况

上周初步完成了eBPF程序编码。在src/dynlib\_monitor.bpf.c中，我定义了event结构体，创建了BPF映射，包括存句柄到路径的handle\_to\_path和发事件到用户空间的events perf buffer。关键是实现了针对dlopen、dlsym和dlclose这三个动态链接函数的uprobe和uretprobe探针。探针里我捕获了进程信息（PID、COMM）、库路径、函数句柄、符号名等数据，并填入了event结构体。初步测试显示，eBPF程序能编译并通过libbpf加载，为用户空间程序的数据处理打好了基础。

2. 本周计划

本周我将开始写用户空间程序，也就是src/dynlib\_monitor.cpp。这个程序负责加载eBPF字节码、管理BPF映射以及从perf buffer接收并处理eBPF事件。具体来说：

用libbpf加载eBPF程序：调用libbpf API加载并附加之前写的eBPF程序。

设置perf buffer回调：实现handle\_event回调函数，接收内核态发来的事件数据。

事件数据解析与格式化：在handle\_event里解析event结构体，根据事件类型（加载、卸载、符号解析）格式化成可读的字符串。这会涉及时间戳转换和dlopen标志解析。

命令行参数处理：实现命令行参数解析，把目标进程名传给eBPF程序里的BPF map。

进程监控与错误处理：确保用户空间程序能正确启动eBPF监控，并处理可能出现的错误，比如eBPF程序加载失败或perf buffer错误。