

# 赛题 7 #

# 基于 Posix 接口的协程框架

直播导师：李清清

## 基于 Posix 接口的协程框架

### What is glibc?

The GNU C Library project provides *the* core libraries for the GNU system and GNU/Linux systems, as well as many other systems that use Linux as the kernel.

These libraries provide critical APIs including ISO C11, POSIX.1-2008, BSD, OS-specific APIs and more.

These APIs include such foundational facilities as open, read, write, malloc, printf, getaddrinfo, dlopen, pthread\_create, crypt, login, exit and more.

<https://www.gnu.org/software/libc/>

对于大部分应用程序，实际并不关心实际的运行实体是什么。基于 linux kernel 的线程调度机制相较于协程占用资源多，上下文切换时间长。当前行业内的解决方案主要是语言级别的协程调度方案，典型的如 go routine。本课题的目标是基于 glibc 的 posix 接口编程，提供用户无感知的协程调度框架，解决提升存量软件性能。

### 产出标准：

根据实际情况，优先完成目标一，二

目标一：pthread 相关库的协程化，包含线程管理以及基于线程的条件变量，锁等接口。

目标二：基于目标一，实现网络编程相关接口的协程化。

目标三：全量的 posix 接口协程化配合。

# 协程机理

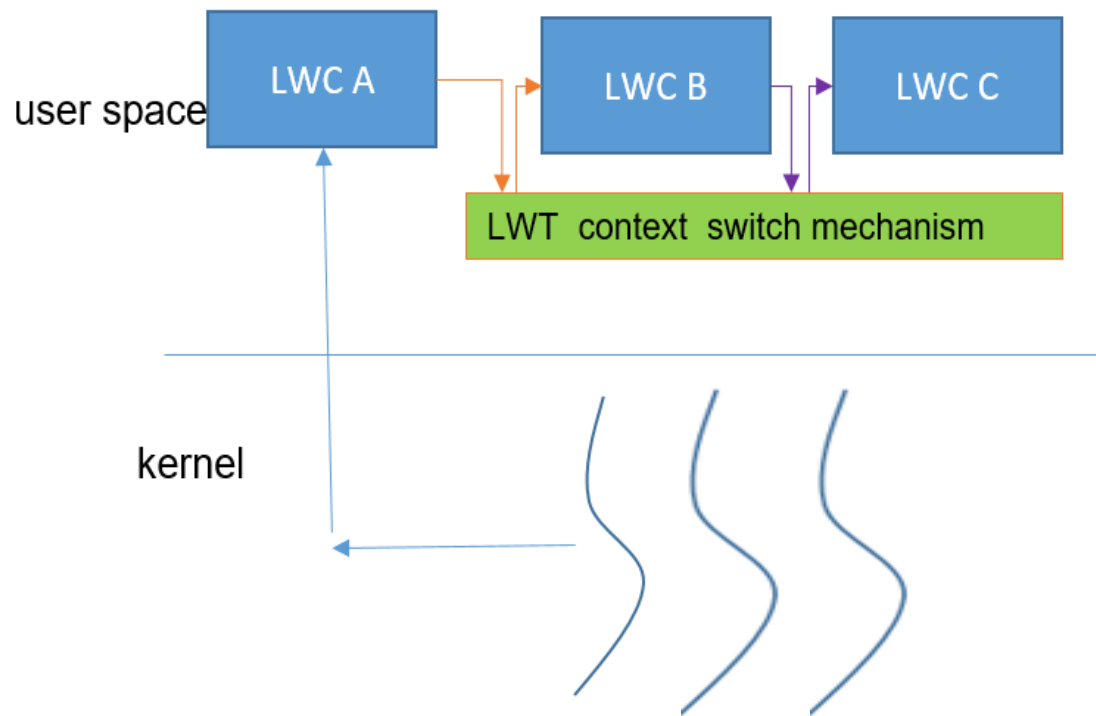
协程机制	原理	案例	备注
API级别	采用系统ucontext, setjump/longjump的接口实现线程切换	利用getcontext/setcontext接口切换协程： <a href="https://github.com/cloudwu/coroutine">https://github.com/cloudwu/coroutine</a>	参考C语言的案例
库级别	实现基本的用户态调度+协程私有栈切换+接管部分系统调用接口	1、腾讯Libco:信号触发调度（线程启动了一个epoll，通过epoll监听事件到来），调度协程。接管socket相关接口。 <a href="https://github.com/Tencent/libco">https://github.com/Tencent/libco</a> 2、libtask（go协程作者实现的简易C协程）	参考C语言的案例
语言级别	语言级别实现用户态协程调度	GO（多线程调度）/C++/.....	C++为无栈协程，需编译器配合。

- 协程可以解决的问题：
- 1、阻塞编程，优化性能。
  - 2、IO瓶颈的操作，优化性能。
  - 3、简化异步编程模型。

## 解题思路

基于 pthread 库实现协程的机理，当用户创建线程时，实际创建协程，并实现协程调度。

- 1、context switch切换机制。
- 2、pthread库hook，用户无感知，主要包含：
  - 创建、删除线程，实现用户态的调度。
  - 线程绑核，线程优先级设置。
  - 条件变量（信号相关），互斥锁。
  - 需要接管掉一部分底层的阻塞接口，配合调度接口。
  - 定时器问题。
- 3、通信相关接口，如socket相关（IO瓶颈优化）。



ps: go协程（GMP模型）是一个比较易用的实现，  
libco是专门针对网络编程所做的性能优化，两者可参考。