

『腾讯后台开发』实习生技能要求

原创：张小方 高性能服务器开发 2018-05-24

如题，应届生除了要良好地掌握算法和数据结构以外，以下一些技能点列表希望对大家有帮助，有兴趣的朋友可以参考这个针对性地补缺补差。文章列出的技能点有的要求熟悉，有的了解即可，注意技能点前面的修饰词。如果没有明确给出“熟悉”“了解”等字眼，要求均为熟悉。

一、操作系统方面

■ 多线程相关与线程之间同步技术

熟练使用（但不局限于）以下linux API

linux下的线程创建、等待、获取线程id

```
1  int pthread_create(pthread_t *thread, const pthread_attr_t *attr, void *(*start_rout
2  int pthread_join(pthread_t thread, void **retval);
3  pthread_t pthread_self(void);
```

常见线程之间的同步技术（何时该用那种技术）

互斥体

```
1  int pthread_mutex_init(pthread_mutex_t *mutex, const pthread_mutexattr_t *mutexattr)
2  int pthread_mutex_destroy(pthread_mutex_t *mutex);
3  int pthread_mutex_lock(pthread_mutex_t *mutex);
4  int pthread_mutex_trylock(pthread_mutex_t *mutex);
5  int pthread_mutex_unlock(pthread_mutex_t *mutex);
```

信号量

```
1  int sem_init(sem_t *sem, int pshared, unsigned int value);
2  int sem_destroy(sem_t *sem);
3  int sem_wait(sem_t *sem);
4  int sem_post(sem_t *sem);
5  int sem_getvalue(sem_t *sem, int *valp);
```

条件变量

```

1  int pthread_cond_init(pthread_cond_t *restrict cond, const pthread_condattr_t *restr
2  int pthread_cond_destroy(pthread_cond_t *cond);
3  int pthread_cond_signal(pthread_cond_t *cond);
4  int pthread_cond_broadcast(pthread_cond_t *cond);
5  int pthread_cond_wait(pthread_cond_t *restrict cond, pthread_mutex_t *restrict mutex
6  int pthread_cond_timedwait(pthread_cond_t *restrict cond, pthread_mutex_t *restrict

```

读写/自旋锁

```

1  int pthread_rwlock_init(pthread_rwlock_t *restrict rwlock, const pthread_rwlockattr
2  int pthread_rwlock_destroy(pthread_rwlock_t *rwlock);
3  int pthread_rwlock_rdlock(pthread_rwlock_t *rwlock);
4  int pthread_rwlock_tryrdlock(pthread_rwlock_t *rwlock);
5  int pthread_rwlock_wrlock(pthread_rwlock_t *rwlock);
6  int pthread_rwlock_trywrlock(pthread_rwlock_t *rwlock);
7  int pthread_rwlock_unlock(pthread_rwlock_t *rwlock);
8  //这两个函数在Linux和Mac的man文档里都没有，新版的pthread.h里面也没有，旧版的能找到
9  int pthread_rwlock_timedrdlock_np(pthread_rwlock_t *rwlock, const struct timespec *
10 int pthread_rwlock_timedwrlock_np(pthread_rwlock_t *rwlock, const struct timespec *
11 int pthread_spin_init (__pthread_spinlock_t *__lock, int __pshared);
12 int pthread_spin_destroy (__pthread_spinlock_t *__lock);
13 int pthread_spin_trylock (__pthread_spinlock_t *__lock);
14 int pthread_spin_unlock (__pthread_spinlock_t *__lock);
15 int pthread_spin_lock (__pthread_spinlock_t *__lock);

```

- 熟悉守护进程的创建、原理
- 了解计划作业**crontab**
- 熟悉进程、线程状态查看命令 (**top**、**strace**、**pstack**)
- 熟悉内存状态查看命令**memstat**、**free**
- 熟悉IO状态查看命令**iostat**、**df**、**du**
- 了解linux文件的权限、用户、时间 (ctime、mtime、atime) 、inode等文件基本属性，熟练使用**chmod**、**chown**、**chgrp**等基本命令。
- 熟悉文件传输命令**scp**、**rz**、**sz**命令、
- 熟悉文件定位命令**find**、**whereis**命令。
- 熟悉软链接，熟悉**ln**命令。

- 熟悉lsof命令。

二、网络

- 熟悉tcp状态机（三次握手、四次挥手）。
- 熟悉tcpdump命令。
- 熟悉网络状态和防火墙状态查看命令：netstat、ifconfig、iptables
- 熟悉 socket API，包括但不限于（**connect**、**accept**、**bind**、**listen**、**send/sendto**、**recv/recvfrom**、**select**、**gethostbyname**）

```
1  int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
2  int accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
3  int bind(int socket, const struct sockaddr *address, socklen_t address_len);
4  int listen(int sockfd, int backlog);
5  ssize_t send(int sockfd, const void *buf, size_t len, int flags);
6  ssize_t sendto(int sockfd, const void *buf, size_t len, int flags, const struct soc
7  ssize_t recv(int sockfd, void *buf, size_t len, int flags);
8  ssize_t recvfrom(int sockfd, void *buf, size_t len, int flags, struct sockaddr *src
9  int select(int nfds, fd_set *readfds, fd_set *writefds,
10             fd_set *exceptfds, struct timeval *timeout);
11 void FD_CLR(int fd, fd_set *set);
12 int  FD_ISSET(int fd, fd_set *set);
13 void FD_SET(int fd, fd_set *set);
14 void FD_ZERO(fd_set *set);
15 struct hostent *gethostbyname(const char *name);
```

- 熟悉epoll，熟悉水平触发与边缘触发。

```
1  int epoll_create(int size);
2  int epoll_ctl(int epfd, int op, int fd, struct epoll_event *event);
3  int epoll_wait(int epfd, struct epoll_event *events, int maxevents, int timeout);
```

- 熟悉阻塞socket和非阻塞socket在connect、send、recv等行为上的区别，如何将socket设置为非阻塞的。

三、脚本工具

- 了解shell基本语法、变量操作、函数、循环/条件判断等程序结构。
- 熟练使用文本编辑工具vi/vim。
- 了解使用文本处理命令grep、sed、cut。
- 了解awk命令。

四、数据库

- 熟悉数据表结构设计（三范式、字段属性）。
- 了解查询优化（索引的概念与创建、sql优化）。
- 熟悉常见的mysql API函数：

```
1  mysql_real_connect
2  mysql_select_db
3  mysql_query
4  mysql_store_result
5  mysql_free_result
6  mysql_num_rows
7  mysql_close
8  mysql_errno
```

五、编程语言

C/C++方面

- 熟悉内存分布（堆、栈、静态/全局/局部变量、虚指针...）
- 熟悉Makefile。
- 熟悉gdb调试（断点、查看内存、执行跟踪、了解CPU主要寄存器作用...）。
- 熟悉性能分析工具(gprof)。

- 熟悉C-Runtime常用函数（如字符串格式化函数printf、scanf，字符串比较连接函数、内存分配函数、文件与目录操作函数等）。
- 熟悉stl库。
- 熟悉OO思想、常见设计模式（如单例模式、工厂设计模式、装饰者模式、Builder模式、生产者消费者模式、策略模式等）。
- 熟悉RAII、pimpl惯用法。
- 有一定的代码质量和重构能力。

文章版权所有，转载请保留文章末尾版权信息和公众号信息。

欢迎关注公众号『easyserverdev』。如果有任何技术或者职业方面的问题需要我提供帮助，可通过这个公众号与我取得联系，此公众号不仅分享高性能服务器开发经验和故事，同时也免费为广大技术朋友提供技术答疑和职业解惑，您有任何问题都可以在微信公众号直接留言，我会尽快回复您。

