## 1,应用技术（知识广度）

## 2,基础技术（知识深度）

### 2.1 JMM内存模型

### 2.2 深拷贝浅拷贝

### 2.3 Serializable序列化

### 2.4 fail-fast

### 2.5 线程

### 2.6 线程池使用、原理、源码

### 2.7 进程之间的通信

（1）管道（pipe） TDMQ和IBMQ的消息转换

（2）信号（signal）

（3）消息队列（message queue） cmq消息发送

（4）共享内存（shared memory） redis

（5）信号量（semaphore）

（6）套接字（socket）

### 2.8 线程之间的通信

### 2.9 CAS

### 2.10 Atomic原子操作类

### 2.11 唤醒线程的方式

（1）使用wait，notify来实现

（2）使用了LockSupport进行线程的阻塞和唤醒

### 2.12 类加载机制

### 2.13 对象的创建过程

### 2.14 对象的内存布局

### 2.15 synchronized

### 2.16 AQS

### 2.17 Semaphore、CountDownLatch、CyclicBarrier

### 2.18 读写锁

### 2.19 阻塞队列、同步队列、条件队列、延迟队列

### 2.20 Fork/Join框架

### 2.21 Disruptor

### 2.22 Callable&Future&FutureTask

### 2.23 CompletionService

### 2.24 CompletableFuture

### 2.25 并发的设计模式