线程池

常用的线程池类型

1. cacheThreadExcutor
2. fixThreadExecutor

线程池的核心参数

1. corePoolSize
2. maximumPoolSize
3. threadFactory
4. statu {RUNNG, SHUTDOWN, STOP, TIDYING, TERMINATED}
5. blockingQueue队列，为何是blockingQueue
6. RejectedExecutionHandler (AbortPolicy, 默认抛出拒绝异常) {DiscardPolicy,DiscardOldestPolicy}
7. keepAliveTime (与getTask 超时时 对work的回收有关，cacheThreadPool 用到)
8. allowCoreThreadTimeOut

线程池的核心数据结构

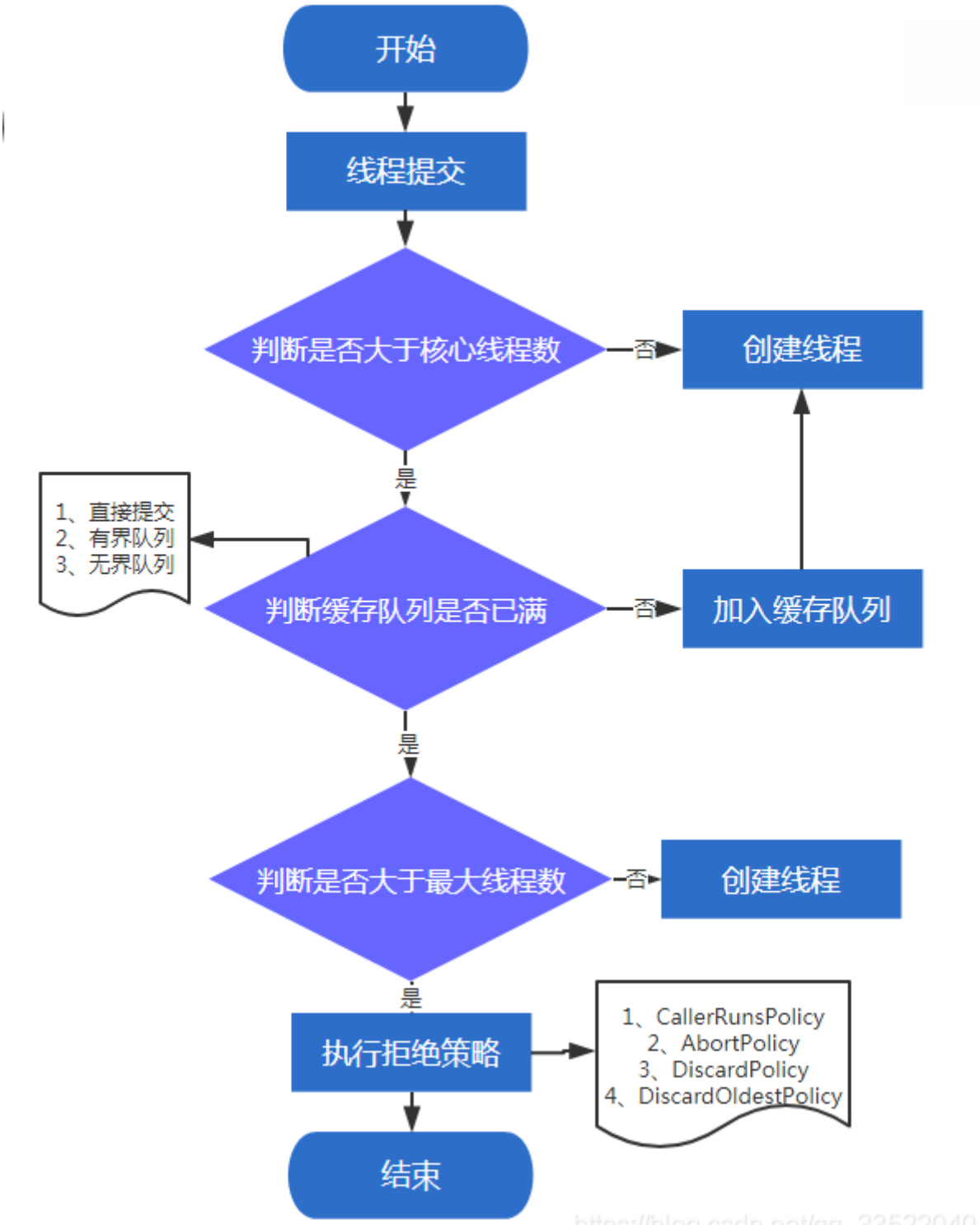
5. private final AtomicInteger ctl 线程状态标志器
6. ReentrantLock mainLock 是因为线程池也有状态，涉及到状态更新时，会先加锁
7. HashSet
8. Condition termination = mainLock.newCondition();

Worker extends AbstractQueuedSynchronizer implements Runnable

```
AbstractQueuedSynchronizer 的存在是因为 要操作线程的状态，因此需要操作前进行cas,
```

线程池的核心流程

```
submit, gettask, 回收线程 等等
```



线程

线程的核心数据

1. 状态threadStatus
2. 名字
3. runnable target
4. groupstackSize
5. groupstackSize
6.

Threadlocal

线程池threadlocal

1. 线程池 sumit -> newTaskFor , 因此需要对这一步进行 线程变量的 暂存(需要实现自定义线程池，并且重写 taskfor方法)
2. 线程初始化 thread.init() , 涉及到 threadlocal 数据的转移, 因此需要 实现新的 thread 对象，实现新的 thradlocal 数据的转移