

```

        createReplicationBacklog();
    }

    // [5]
    ...
}

```

【1】如果处理的是 PSYNC 命令，则调用 `masterTryPartialResynchronization` 函数尝试进行部分同步操作，部分同步成功则退出函数，否则函数继续执行。

【2】执行到这里，代表无法进行部分同步操作，主从节点需要进行全量同步。更新 `client.replstate`，进入 `SLAVE_STATE_WAIT_BGSAVE_START` 状态。

【3】将该客户端添加到 `server.slaves`。

【4】如果主节点的复制积压区未创建，则创建复制积压区。这时需要初始化 `server.replid`，置空 `server.replid2`。

【5】执行到这里，主节点需要生成一个 RDB 文件，并发送到从节点。这里针对以下 3 种情况进行处理。

(1) 主节点正在生成 RDB 数据，并保存到磁盘中。如果当前从节点列表中存在其他处于 `SLAVE_STATE_WAIT_BGSAVE_END` 状态的从节点客户端，则说明当前生成的 RDB 文件也可以被当前从节点客户端使用，这时将当前从节点客户端 `client.replstate` 设置为 `SLAVE_STATE_WAIT_BGSAVE_END` 状态，直接使用当前生成的 RDB 文件。

(2) 主节点正在生成 RDB 数据，并直接发送到 Socket（无磁盘同步）。这时需要等待该 RDB 操作完成后，再生成一个新的 RDB 文件。

Redis 提供了主节点无磁盘同步功能（类似于从节点无磁盘加载），开启该功能后，主节点生成的 RDB 数据并不会保存到磁盘中，而是直接发送到 Socket 中，适用于磁盘性能较低而网络带宽较充足的场景。

(3) 当前程序中不存在子进程，调用 `startBgsaveForReplication` 函数生成 RDB 文件。

13.4.2 全量同步

`startBgsaveForReplication` 函数负责生成 RDB 文件并发送给从节点：

```

int startBgsaveForReplication(int mincapa) {
    ...
}

```