**基于SPARC模拟器的多核调试技术研究与实现**

**报告提纲**

1. **选题的背景及意义**

在SPARC模拟器上需要对应用程序进行调试。由于模拟的是多个BM3803核，因此需要实现针对SPARC架构的多核调试。

在多核调试过程中，为了满足调试态对运行态更接近的模拟，需要实现多核同步调试的功能；而为了控制不同处理器核的运行状态以及信息交互，需要保证调试可以异步执行。

1. **国内外本学科领域的发展现状与趋势**

现阶段，虽然多核架构芯片已成为高端嵌入式产品的首选，但国内外在嵌入式调试技术方面、尤其是基于多处理器所做的研究工作较少，较为成熟的多核硬件调试工具都是基于 JTAG 技术，并且都需要较高的成本。作为单核架构下首选的任务级调试器 GDB，虽然能在嵌入式 Linux 支持下调试多核架构下的程序，但原有单核调试功能远不能满足用户多核调试的需要。

在多核调试过程中，由于系统复杂度和系统的不确定性，多个处理器核之间通信和同步等问题，使得调试难度比单核调试增大很多。虽然高版本的GDB协议支持远程多核调试，但国内外公开的研究中，还几乎没有基于SPARC架构的很好的解决方案。

1. **课题主要研究内容**

* 为了提高模拟器调试功能的可扩展性，运用flex词法分析器高效实现GDB协议的序列化与反序列化，为模拟器提供交互接口。反序列化将协议数据转化为模拟器可以操作的数据结构，并且解析出协议类型；序列化将模拟器完成调试后的结果返回到宿主端。
* 目前多核调试的相关研究，目标机都是软硬件结合的系统，在其上将GDBServer作为独立的应用程序，通过操作系统提供的接口调试其它程序。而在模拟器平台上，一方面没有操作系统的支持，另一方面，为了将来模拟器的功能模块改变后，依然可以对其进行调试，便将调试器作为模拟器框架的一部分，直接访问应用程序的内存空间和处理器核的寄存器来获取调试信息，因此，需要修改模拟器框架，在调度处理器核时加入对调试命令的处理。
* 断点处理机制是调试过程的核心。在具备操作系统的目标机上，GDBServer通过在断点处设置断点指令，并且通过系统调用，完成断点处理；而在模拟器中，需要探索新的方法，记录断点位置，同时实现对条件断点等高级断点的支持，当被调试的核运行到断点处时，正确进行处理。
* 异步模式下的多核调试机制研究。在异步模式下，宿主机端可以在上一条调试信息未返回前，产生新的调试命令，同时还可以针对不同的处理器核进行断点设置。为了支持这些功能，模拟器需要采用相应的机制保证按序正确处理多个调试命令，以及在不影响其它核的情况下，实现对某一个核的调试。

1. **拟采用的研究方法、技术路线**

* 根据GDB调试协议的规范和标准，利用flex词法分析器对数据进行反序列化，得到模拟器可以处理的数据结构；制定模拟器返回的调试结果的标准，将调试信息组装为符合GDB调试协议的数据包返回给宿主端。
* 参考操作系统为调试提供的系统调用(以Linux下的ptrace为例)，在模拟器框架中，实现对被调试程序内存空间以及处理器核寄存器的访问，实现对多种调试命令的支持。
* 在断点处理中，设计不同种类断点对应的数据结构，并且将断点保存在断点队列中。监控被调试核，当处理器核运行到某个断点处时，模拟器需要判断断点是否有效，如果有效，需要暂停处理器核的运行，同时记录下整个系统的快照，根据接下来的调试命令，返回对应的调试信息。
* 高版本的GDB(7.0之后)支持多核异步调试，在异步模式下，处理器可能接收到多个调试命令，将这些命令视为事件，保存在事件队列中，按照顺序处理不同的调试命令，同时，根据命令中的线程ID判断当前调试命令对哪个核起作用。在对一个核调试的过程中，需要保证其它核的正确运行。

1. **已有科研基础与所需的科研条件**

已经完成了双核SPARC模拟器的基本功能，在此基础上实现GDBServer功能。

对于GDB调试协议有一定研究，分析出了调试过程中主要的协议类型。

**6、 研究工作计划与进度安排**

**7、 参考文献**