

유니뷰 JDB 서버의 수행 제어 명령의 설계 및 구현

○
옥 재 호*, 정 민 수**, 윤 기 송***

(주) 에듀뱅크*, 경남대학교 컴퓨터 공학과**,

한국전자통신연구원 컴퓨터&소프트웨어기술연구소 인터넷서비스연구부***

Design and Implementation of Execution Control Command for Uniview JDB Server

Jae-Ho Ock*, Min-Soo Jung**, Ki-Song Yoon***

*Edubank Corp., **Dept. of Computer Engineering, Kyungnam University,

***Internet Service Dep., Computer & software Technology Lab., ETRI

요 약

유니뷰(uniview) 시스템은 이기종 분산환경에서 동작하는 프로그램을 디버깅하는 도구로서 시스템 종속적인 부분은 디버깅 서버로 분리시킴으로써 사용자 인터페이스의 이식성과 개발의 효율성을 증대시킨 클라이언트/서버 시스템이다. 현재 유니뷰 시스템은 GNU 디버거인 gdb를 기반으로 동작하며 C 및 C++언어를 지원한다. 본 논문에서는 자바 어플리케이션 디버깅을 지원하기 위한 유니뷰 jdb 서버와 서버 시스템 개발 과정 중 프로그램 수행 제어 명령에 대한 설계 및 구현에 대해 설명한다.

1. 서 론

최근 중소형급 컴퓨터가 다양해지면서 클라이언트 서버 시스템과 같은 이기종 컴퓨터에서 동작하는 분산 시스템이 많이 등장하고 있으며, 분산 시스템 환경에서의 많은 연구가 진행되고 있다. 하지만 분산 처리 시스템의 개발은 통신에 소요되는 시간이 예측할 수 없다는 점과 여러 프로그램들이 동시에 동작한다는 점 때문에 순차적 프로그램의 디버깅 보다 훨씬 복잡하다.

이러한 배경에서 한국 전자 통신 연구소에서는 94년부터 시작된 과학기술처 특정 연구과제로서 미국 실리콘밸리의 탠덤사와 국제 공동 과제로 유니뷰 시스템을 개발 하였다[1]. 유니뷰 시스템은 클라이언트 서버 모델의 분산처리 디버거로써 유니뷰 시스템 클라이언트는 이기종 호스트의 각 디버거들에 대한 단일화된 디버깅 개념과 인터페이스를 제공하며, 유니뷰 시스템 서버는 동일 호스트에 존재하는 한 개 이상의 디버거 프로세스와 연동함으로써 디버깅 서비스를 제공하며 개방성과 확장성을 가지므로 분산 시스템을 효과적으로 디버깅할 수 있는 도구를 제공한다. 하지만 유니뷰 시스템은

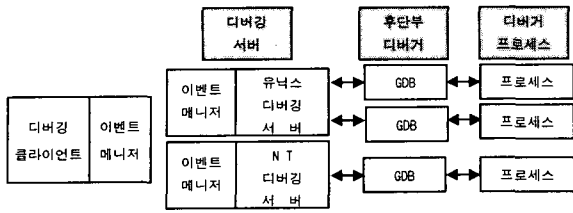
후반부 디버거로써 GNU 디버거를 사용하고 있으므로 현재 널리 사용되고 있는 자바언어를 지원하지 못한다.

따라서 본 연구는 자바 디버거로 알려진 jdb를 이용하여 자바 언어로 구현된 어플리케이션에 대한 디버깅이 가능한 확장된 유니뷰 시스템 개발을 목표로 한다.

본 논문의 구성은 다음과 같다. 2장에서는 유니뷰 시스템에 대해 설명한다. 3장에서는 유니뷰 JDB 서버의 프로그램 수행 제어 명령의 설계 및 구현에 관해 기술하고 마지막 4장에서는 결론 및 이후 연구 계획에 관해서 기술한다.

2. 유니뷰 시스템

유니뷰는 높은 수준의 그래픽 사용자 인터페이스를 제공하는 심볼릭 분산 디버거로서, 현재 UNIX와 NT 플랫폼을 포함한 이기종 환경 하에서 분산 시스템을 디버깅할 수 있다. 그림 1은 유니뷰의 시스템 전체 구조를 보여준다.



<그림 1> 유니뷰 시스템의 전체 구조

유니뷰는 디버깅 클라이언트와 디버깅 서버의 두 개의 하위시스템으로 구현되었다. 하나의 디버깅 서버는 디버깅 호스트에 각각 하나씩 존재하면서 호스트의 후단부 디버거를 제어하는 역할을 수행한다. 현재는 UNIX와 NT 플랫폼에서의 후단부 디버거로 GNU 디버거인 gdb를 사용하고 있다. 디버깅되는 프로세스에 대해 각각 하나의 gdb 프로세스가 존재하고 각 디버깅 서버는 파이프를 통하여 gdb와 연결되어, 하나 이상의 gdb 프로세스를 제어할 수 있다. 디버깅 클라이언트는 디버깅 관련 정보들을 모아 이를 통일된 그래픽 사용자 인터페이스를 이용하여 사용자에게 보여준다. 클라이언트와 서버간의 통신은 이벤트 매니저를 통하여 이루어진다. 이벤트 매니저는 서버와 클라이언트 상호간의 통신 메커니즘을 제공하는 라이브러리로서 ONC/RPC를 기반으로 하여 설계되었다[3,4]. 클라이언트는 이벤트 매니저를 이용하여 서비스를 요청하는 이벤트를 서버에게 보내고, 서버는 해당 이벤트를 처리하고 이벤트 매니저를 통하여 클라이언트로 회신 이벤트를 보내게 된다.

이와 같은 클라이언트/서버 구조로 인하여 유니뷰는 디버깅되는 프로그램에 대해 단일한 사용자 인터페이스를 제공할 수 있으며, 중앙 클라이언트에서 모든 디버깅 프로그램에 대한 제어와 인터페이스를 담당함으로써 분산 디버깅을 수행할 수 있다. 또한, 클라이언트와 서버간의 인터페이스를 명확히 정의함으로써 후단부의 호스트 플랫폼이나 디버거의 종류에 관계없이 통신할 수 있는 개방성을 가지며, 이기종 환경 하에 있는 여러 개의 분산된 프로세서들을 디버깅할 수 있다.

유니뷰의 순차적 디버깅 기능은 후단부 디버거로 사용한 gdb가 가지는 모든 특성과 제한점을 공유한다. 즉 유니뷰는 단일 스레드, 단일 프로세스 프로그램에 관하여 gdb의 모든 심볼릭 디버깅의 특성 및 일부 제약 조건 하에서 제공되는 gdb의 다중 스레드 및 멀티태스킹 프로그램의 심볼릭 디버깅의 기능을 제공한다.

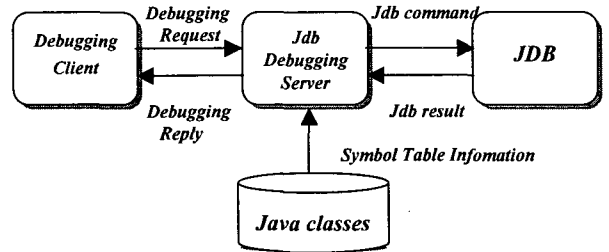
3. 유니뷰 JDB 서버

3.1 유니뷰 JDB 서버의 개요

jdb 서버는 기존의 gdb를 이용한 유니뷰 시스템에

Sun사의 jdb 디버거를 이용한 서버를 추가 함으로써 C/C++뿐만 아니라 java 언어에 대한 디버깅이 가능하도록 한다. gdb를 이용한 유니뷰 시스템은 실행시 클라이언트가 요구하는 다양한 디버깅 정보를 제공하지만 jdb 디버거는 gdb 디버거에 비해 제한된 디버깅 정보만을 제공함으로써 클라이언트가 요구하는 디버깅 정보를 전부 제공해 줄 수가 없다. 특히 유니뷰 시스템이 프로그램 디버깅시 꼭 필요한 프로그램 심볼 테이블 정보를 제공하지 않는다[6].

하지만 자바 프로그램은 컴파일시 생성되는 클래스파일 내에 심볼 테이블에 대한 정보를 포함하고 있어 자바 클래스 파일에 대한 분석을 통해 심볼 테이블에 대한 정보를 얻을 수 있다. 그림 2는 클라이언트, 서버 및 jdb, 자바 클래스파일 사이의 연관성을 나타내는 그림이다.



<그림 2> 클라이언트, 서버 및 Jdb 사이의 연관도

그림에서 jdb 디버거 서버는 jdb와 파이프를 통해 디버깅 실행 명령어와 실행 결과를 주고 받으며, 해당 자바 클래스 파일을 읽어 심볼 테이블에 대한 정보를 저장하여 클라이언트의 요구에 맞게 디버깅 정보를 보내준다.

3.2 수행 제어 명령의 설계 및 구현

jdb 서버에 의해 제공되는 프로그램 수행 제어 명령어는 resume, resume to, step의 세가지이다.

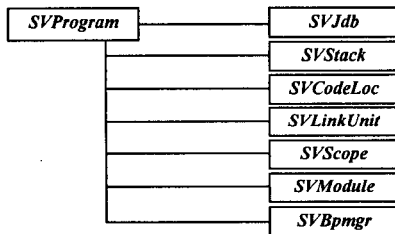
Resume은 프로그램의 수행을 재개하여 다음 정지할 원인이 발생할 때까지 계속 수행하며, jdb 명령어 'cont'에 대응된다. resume to는 프로그램의 수행을 재개하되 지정된 위치 이전에 정지할 원인이 발생하면 정지한다. 정지할 원인은 브레이크포인트를 만나거나 프로그램이 hold 명령어를 받아 멈추거나 또는 다른 원인에 의해 프로그램이 수행을 정지하는 경우이다. jdb에 대응하는 명령어는 없으며, 임의로 원하는 위치에 브레이크포인트를 걸고 resume 한다.

Step은 프로그램의 수행을 단계적으로 제어하는 것으로 step in, step over, step out의 세가지 모드가 있다. Step over란 현재의 위치에서 다음 문장을 수행하라는 것으로 다음 문장이 함수 호출인 경우 그 함수의 수행이 끝나고 다음으로 넘어가서 정지한다. Step in은 다음 문장이 함수 호출인 경우 그 함수의 시작 위치로 이동하여

정지하는 명령으로 다음 문장이 함수 호출이 아닌 경우나 함수의 시작 위치로 이동하는 것이 어려운 경우는 step over와 같은 효과를 낸다. Step out은 현재의 스택 프레임에서(함수 내부인 경우)함수를 끝까지 수행하고 호출한 위치로 돌아가서 멈추라는 명령이다. jdb에서 각각 step over는 next, step in은 step, step out은 step out 명령에 대응된다. 이들 step 명령을 수행하는 중에도 정지할 원인이 발생하면 프로그램은 그 위치에서 수행을 정지하게 된다.

그 밖에 프로그램 수행과 관련하여 hold 명령이 있다. hold 명령은 프로그램 실행 중에 프로그램에 인터럽트 신호를 보내 현재의 위치에서 서게 하는 명령으로 프로그램이 정지하면 프로그램의 상태는 hold로 바뀌고 프로그램의 현재 위치가 소스 화면에 나타나고 수행 명령 메뉴들이 활성화된다. hold는 jdb의 suspend에 대응된다. 따라서 프로그램 수행이 무한 루프인 경우에 프로그램을 정지시킬 수 있다.

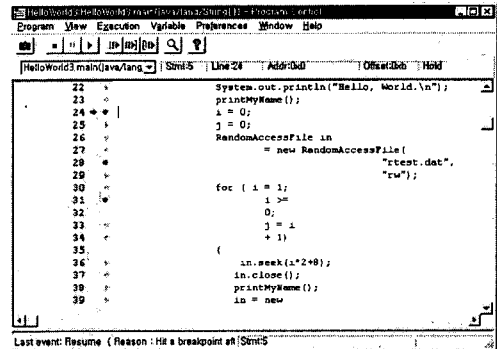
그림 3은 수행 제어 명령에 관련된 jdb 디버거 서버의 프로그램 클래스 다이어그램이다.



<그림 3> 프로그램 클래스 다이어그램

- SVProgram 클래스 : 디버깅되는 프로그램에 대한 모든 정보를 가진 클래스로서 프로그램의 정적 정보를 구축하는 일과 수행 제어를 담당하는 역할을 한다.
- SVLinkUnit 클래스 : 디버깅 중인 자바 클래스파일과 관련된 사용자 클래스에 대한 정보를 저장한다.
- SVModule 클래스 : 링키지 유닛의 하부구조로 각 클래스파일 대응하는 멤버함수들에 대한 정보를 가진다.
- SVScope 클래스 : 클래스파일내 멤버 함수에 대한 소스파일 정보와 멤버 함수 내 문장에 대한 정보(SVStmtTable)를 가진다.
- SVStack 클래스 : 프로그램이 정지할 때마다 현재의 스택을 계산하고 유지하는 역할을 한다.
- SVCCodeLoc 클래스 : 프로그램 안의 코드 부분 위치를 나타내는 데 사용된다. 브레이크 포인트 위치, 프로그램의 현재 위치 등을 나타낸다.
- SVJdb 클래스 : 후단부 jdb와의 연결 및 수행 제어 명령을 수행하며, 실행 결과를 반환한다.
- SVBpmgr 클래스 : 브레이크 포인트에 대한 설정 및 해제, 브레이크 포인트에 대한 정보를 저장한다.

그림 4는 구현된 jdb 서버의 실행 제어 명령중 resume 명령에 대한 실행 결과이다.



<그림 4> 프로그램에 대한 resume 실행 결과

실행 후 프로그램은 다음 브레이크 포인트에 커서가 위치하며 현재 상태가 hold 상태로 바뀐다. 그리고 현재 상태에 대한 정보가 화면 상단에 나타나게 되며 수행 명령 메뉴가 활성화 된다.

4. 결론 및 이후 연구계획

유니뷰 시스템은 이기종 분산환경에서 동작하는 프로그램을 디버깅하는 도구로서 C/C++과 유니뷰 JDB 서버의 개발과 함께 자바 언어에 대한 디버깅을 지원한다. 현재 유닉스 개발 환경인 솔라리스와 NT 환경의 포팅이 완료된 상태이며, 자바 프로그램의 특성에 맞추어 클라이언트의 기능 확장을 준비 중에 있다[2,5]. 앞으로 개발 과정 중에 발견된 jdb의 여러 가지 버그들에 대한 해결이 연구 과제로 남아 있다.

참 고 문 헌

- [1] 성명제 외 4인, "분산 디버거 유니뷰 시스템의 개발", 정보과학 98년 논문지(C), 8월호 1998.
- [2] 이공선 외 4인, "분산 진단/교정 서버모듈의 NT 이식," 정보처리학회 97년 추계학술발표논문집, 제 4권 2호, pp.1475-1478, 경기대학교, 1997.
- [3] 조영욱 외 4인, "분산처리 디버거 유니뷰 서버의 구조," 정보과학회 97년 추계학술발표 논문집, 제 24권 2호(I), pp.327-330, 이화여자대학교, 1997.
- [4] J. Bloomer, *Power Programming with RPC*, O'Reilly & Associates, O'Reilly & Associates, Inc.
- [5] W. Stevens, *Advanced Programming in the UNIX Environment*, Addison Wesley, p.744, Addison-Wesley
- [6] <http://java.sun.com/>, Sun Microsystems, Java Home Page