

분산 진단/교정 시스템을 위한 데이터 디스플레이에 관한 설계 및 구현

정 연정, 조 영욱, 이 단형

컴퓨터 & 소프트웨어 기술 연구소 네트워크컴퓨팅연구부 네트워크소프트웨어연구실

A Design and Implementation of Data Display for Distributed Diagnosis/Correction System

Yeonjeong Jeong, Youngwook Cho, Danhyung Lee

Network Software Lab., Network Computing Dep., Computer & Software Technology Lab., ETRI

요 약

분산 처리 시스템은 여러 대의 컴퓨터에 분산된 프로그램들이 상호 통신을 통해 협력하면서 작업을 수행하는 시스템이다. 본 논문은 분산 처리 시스템의 소프트웨어 개발을 효율적으로 지원하는 유니뷰(Uniview)를 개발하는 데 있어서 디스플레이 데이터의 설계 및 운용 방안을 제시한다. 디스플레이 데이터는 디버깅 작업과 관련된 정보를 효과적으로 페치, 저장할 수 있는 구조가 되어야 한다. 제안된 디스플레이 데이터 구조와 운용 방법은 디버깅 작업의 흐름에 따라 데이터를 생성, 유지하며 필요한 데이터만을 검사하여 효율적으로 운용한다.

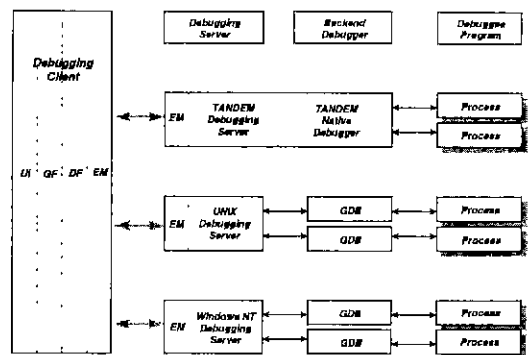
1. 서론

분산 처리 시스템은 여러 대의 컴퓨터에 분산된 프로그램들이 상호 통신을 통해 협력하면서 작업을 수행하는 시스템이다[1,3]. 분산된 컴퓨터 상에서 상호 통신하면서 동작하는 프로그램들로 이루어진 시스템은 단일 프로그램에 비해 개발 비용이 높고 소프트웨어 질을 보장하기 어렵다. 이러한 분산 환경의 소프트웨어 생산성을 높이기 위한 도구의 개발 및 상품화가 활발히 이루어지고 있다[4].

본 논문은 분산 처리 시스템의 소프트웨어 개발을 효율적으로 지원하는 유니뷰(Uniview)를 개발하는 데 있어서 디스플레이 데이터의 설계 그리고 운용 방안을 제시한다. 디스플레이 데이터는 디버깅 작업과 관련된 정보를 효과적으로 페치, 저장할 수 있는 구조가 되어야 한다. 디버깅 작업의 흐름에 따라 필요한 데이터를 생성, 유지하며 불필요한 데이터를 삭제하여 효율적으로 운용할 수 있어야 한다.

본 논문은 제 2 장에서 분산 진단/교정 시스템의 구조를 살펴보고, 제 3 장에서 디스플레이에 필요한 기능을 살펴본다. 제 4 장에서는 디스플레이 데이터의 자료 구조와 관리 구조를 살펴본다. 제 5 장에서는 디스플레이의 실행을 살펴보고 제 6 장에서 결론을 맺는다.

분산처리 진단/교정 시스템은 이기종 분산환경에서 동작하는 프로그램을 디버깅하는 도구로서 시스템 종속적인 부분은 디버깅 서버로 분리시킨 클라이언트/서버 구조로 되어있다. 디버깅 서버는 각 호스트상에서 동작하는 프로그램을 제어하고 디버깅 정보를 유지 관리하는 역할을 한다. 클라이언트는 사용자 인터페이스를 제공하고 디버깅 서버들과 통신하면서 디버깅 정보를 총괄 처리하는 역할을 한다[4].



2. 분산 진단/교정 시스템 구조

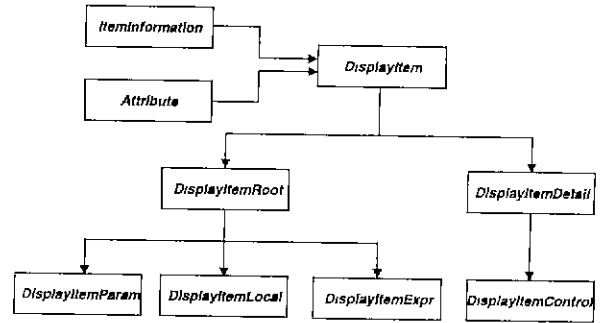
<그림 1> 분산 진단/교정 시스템 구조

분산처리 진단/교정 시스템은 이식이 힘든 코드를 후단부 디버거를 이용하여 독립적인 서버를 구축함으로써 플랫폼에 종속적인 부분을 최대한 분리시켰다.

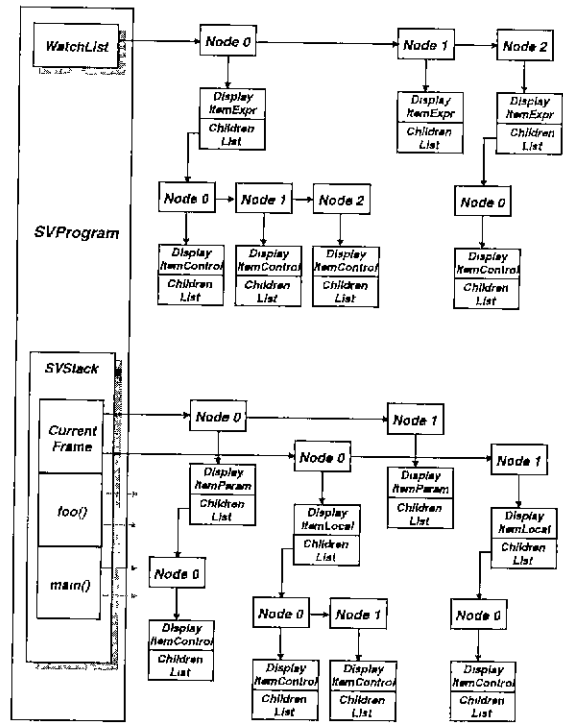
3. 디스플레이 가능

디버깅 작업을 수행하기 위해 변수에 필요한 정보를 생성하고 생성된 데이터를 함수의 실행 동안 관리 및 유지하고 삭제하는 기능이 필요하다. 본 절에서는 이러한 기능을 수행하기 위한 모듈을 조사하고 정의한다.

- 파라미터, 로컬 변수 페치: 파라미터, 로컬 변수들을 생성하고 필요한 정보를 구성한다
- 와치 변수 페치: 와치하고자 하는 변수를 클라이언트로부터 받아 스코프, 변수 값등의 정보를 구성한다.
- 초이스 리스트 페치: 변수 값을 나타내기 위한 형식을 구성하고 그 형식을 클라이언트에 등록한다. 변수 값을 나타내기 위한 형식으로 십진법, 16진법, 8진법, 바이너리 등이 있다.
- 변수의 자식 페치: 파라미터, 로컬 변수, 와치 변수 등이 구조체, 클래스인 경우 그 자식 변수들을 생성하고 필요한 정보를 구성한다. 변수가 배열인 경우 차수가 낮은 배열들 생성하고 필요한 값이나 정보를 구성한다.
- 와치 변수 복사: 와치 변수들을 나타내고 있는 윈도우를 복사할 경우 새로운 와치 변수들을 구성한다.
- 변수 삭제: 함수의 수행이 끝났을 경우 필요 없는 파라미터, 로컬 변수나 필요 없는 와치 변수들을 클라이언트로부터 전달 받아 삭제한다
- 변수 형식 변환: 변수의 형식을 변환한다. 예를 들어 십진수에서 16진수로 변수 값의 형식을 변환한다.
- 변수 값 수정: 디버깅 작업 중 수정하고자 하는 변수의 값을 변환시킨다.
- 배열 바운드 설정: 배열을 나타내는 범위를 지정한다. 예를 들어 sample[10] 있다면 0에서 5까지의 인덱스를 기지는 배열만을 나타내고자 할 때 배열의 바운드를 지정하게 된다.



<그림 2> 디스플레이 데이터 자료 구조



<그림 3> 디스플레이 데이터 관리 구조

4.1 디스플레이 데이터 클래스 구조

디스플레이 데이터는 변수에 대한 정보와 디버깅에 필요한 정보를 가진다. 본 논문은 디스플레이 데이터를 역할에 따라 파라미터, 로컬, 와치 변수로 나누고 아래 그림에서

DisplayItemParam, DisplayItemLocal, DisplayItemExpr 으로 정의하고 이들의 공통된 정보를 DisplayItemRoot 로 정의한다. 어떤 변수의 자식들은 DisplayItemControl 에 정보를 가지게 되며 이들은 부모에 의해 포인트된다. 모든 변수는 자식을 가질 수 있으며 그 자식 또한 자식을 가질 수 있다.

4.2 디스플레이 데이터 관리 구조

본 논문은 함수의 파라미터 변수와 로컬 변수의 디스플레이 데이터를 관리하기 위하여 그 함수에 해당하는 스택의 프레임에 각각의 포인터 링크드 리스트를 만들어 관리한다. 와치 변수의 디스플레이 데이터는 프로그램에 포인터 링크드 리스트를 만들어 관리한다.

위 <그림 3> 에서 프로그램은 디버깅하는 프로그램을 관리하기 위한 클래스이다. 스택은 함수 호출에 따른 각 함수의 정보를 저장하며 새로운 함수가 호출되면

증가하게 된다. 스택의 각 프레임은 함수의 파라미터 변수와 로컬 변수를 관리하기 위하여 각각에 대한 포인터 링크드 리스트를 가진다

프로그램의 실행시 파라미터 변수와 로컬 변수의 변화는 현재 실행되고 있는 함수에서 발생하게 된다. 그 함수는 스택의 탑에 존재하게 되고 스택의 탑에 있는 파라미터 변수와 로컬 변수를 검사하면 된다. 이와 같이 스택에서 파라미터 변수와 로컬 변수를 관리함으로 모든 파라미터 변수와 로컬 변수를 검사하지 않고 필요한 변수만을 검사하여 실행 효율을 높일 수 있다.

와치 변수는 프로그램의 실행 동안 검사하고자 하는 변수이다. 와치 변수는 위치에 따라서 다른 스코프를 가지지기 때문에 이들을 하나의 그룹으로 나누고 검사한다. 프로그램 실행 단계마다 모든 와치 변수는 검사된다. 위 <그림 3> 에서 프로그램에 바로 연결되어 있는 것이 와치 변수를 관리하기 위한 포인터 링크드 리스트이다.

파라미터, 로컬 및 와치 변수에 자식이 있는 경우 즉 구조체나 클래스의 구성 요소가 존재하는 경우 부모는 자식의 디스플레이 데이터를 관리하기 위한 포인터 링크드 리스트를 가진다.

5. 디스플레이 실행

프로그램이 시작하면 디버깅 클라이언트로부터 main() 함수의 파라미터, 로컬 변수의 생성 요청이 발생한다. 서버는 후단부 디버거를 이용하여 main() 함수의 파라미터, 로컬 변수를 알아내고 그들에 대한 정보를 생성, 분류하여 디스플레이 데이터를 구성한다. 구성된 디스플레이 데이터는 그 함수를 관리하기 위한 스택의 프레임에 존재하는 포인터 링크드 리스트에 등록된다.

프로그램 수행 중 함수에 대한 step in 이 발생하면 스택이 증가되고 위에서 설명한 main() 함수와 같은 방법으로 그 함수에 대한 디스플레이 데이터가 생성되고 관리된다.

사용자가 어떤 변수를 와치할 경우 서버는 후단부 디버거를 이용하여 와치 변수에 대한 정보를 구성하고 프로그램에 존재하는 포인터 링크드 리스트에 디스플레이 데이터를 등록하여 관리한다

파라미터, 로컬 및 와치 변수의 자식을 보고자 할 경우 서버는 후단부 디버거를 이용하여 그 변수의 자식들을 알아내고 그들에게 필요한 정보를 구성하여 디스플레이 데이터를 생성한다. 생성된 자식의 디스플레이 데이터는 부모 디스플레이 데이터에 의해 포인트되어 관리된다.

와치 변수의 갱신은 프로그램의 포인터 링크드 리스트 따라 각 변수의 스코프에 맞게 후단부 디버거를 수행시켜 그 변수의 새로운 값과 디스플레이 데이터에 존재하는 값과 비교한다. 파라미터, 로컬의 경우 스택의 탑에 존재하는 변수들이 실행되고 있는 함수의 변수이므로 모든 파라미터, 로컬 변수를 검사하지 않고 이들만을 검사하여 효과적인 검사를 수행한다.

