다중 스레드 방식을 도입한 형태소 해석기

최 유 경°, 안 동 언, 정 성 종, *이 신 원, **두 길 수,
노 영 만, 오 형 진, 김 금 영, 이 동 광
전북대학교 컴퓨터공학과, *정인대학, **서남대학교
전화 : (063) 270-2416 / 핸드폰 : 016-644-4479

A Morphological Analyzer with Multi-Threads Method

Yoo Kyung Choi°, Dong Un An, Seung Jong hung, *Shin Won Lee, **Gil Su Doo,
Young Man Ro, Hyung Jin Oh, Gum Young Kim, Dong Kwang Lee
Dept of Computer Engineering Chonbuk National University,
*Dept of Computer & Information Chongin College,
**Dept of Computer Science & Information Communications Seonam University
E-mail : ykchoi@ielabhp.chonbuk.ac.kr

Abstract

In recent, a morphological analyzer be used for indexing system in information retrieval system. A morphological analyzer as a indexing system must have multiprocessing ability to deal with multiple users and documents.

To meet the needs of these, we propose a morphological analyzer with multi-threads method.

To use multi-threads method, we consider memory allocation problem, threads synchronization problem, code optimization and so on.

In this paper, first, we report several manners for multi-threads. And next, to evaluate our proposed system, we make a comparison test between proposed system and existing system.

I. 서론

그 동안 한국어 정보처리 분야에서는 형태소 해석기에 대한 연구가 많이 진행되어 왔고, 매우 높은 정확률을 보이고 있다. 특히 정보검색 시스템이 널리 보급되면서, 색인기의 중요성도 커지고 있으며 단순한 명사 추출기보다는 형태소 해석기가 본격적으로 도입되었다. [1]

정보검색 시스템의 색인기로서 사용되는 시스템은 크게 형태소 분석기를 이용하는 방법, 형태소 분석기와 품사 태거를 이용하는 방법, 언어분석 도구를 사용하지 않는 방법의 3가지로 나누어 볼 수 있다. [2]

다량의 문서를 처리해야 하는 정보 검색 시스템이나 다수 이용자가 이용하는 정보검색 시스템에서 색인기로 형태소해석기를 사용하기 위해서는 다중 처리 기능을 가지고 있어야 한다.

한 프로그램에서의 독립적인 하나의 작업 단위를 스레드(thread)라고 한다.

가끔 프로그램에서 한 번에 두 가지 이상의 일을 동시에 수행하게 하는 것이 유용할 때가 있다. 전통적인 예를 들자면, 텍스트를 수정하는 동시에 문서에서 단어의 개수를 세는 것이다. 하나의 스레드가 사용자의 입력을 관리하며 문서의 수정을 행할 수 있고, 같은 문서에 접근하는 또 다른 스레드는 단어의 개수를 저장하는 변수를 계속하여 갱신할 수 있다.[3] 실제적인 응용으로 윈도우 OS의 내부적인 프로그램은 다중 스레드로 설정되어 있어, 멀티태스킹을 수행할 수 있다.

본 논문에서는 정보검색 시스템의 색인기로 사용할

형태소 해석기에 다중 스레드 기능을 도입하여 동시에 여러 입력을 처리할 수 있는 모델로 설계하였다.

이를 구현하기 위해서는 일단 형태소 해석기에서 사용하는 전역변수의 사용을 최소화해야 하며, 불필요한 루틴을 제거하여야 한다.

또한 스레드 간의 메모리 공유로 인해 발생할 수 있는 예기치 못한 결과를 방지하기 위해, 스레드 간 순서를 제어해 주는 등의 동기화 메커니즘을 적절히 적용시켜야 한다.

이러한 필요 조건을 적용하여 시스템을 구현하고, 본 논문에서 제안한 다중 스레드를 도입한 형태소 해석기를 평가하기 위해 다중 스레드를 도입하지 않은 형태소 해석기와 비교하여 처리 능력의 향상 및 실제 응용에 있어서의 효율성을 검증해 보고자 하였다.

이후 논문의 구성은 다음과 같다.

제 2 절에서 다중 스레드를 사용하기 위해서 고려해야 할 사항과 다중 스레드 방식을 적용하기 위해 형태소 해석기에 취한 방법을 소개한다. 제 3 절에서는 다중 스레드 방식을 적용하지 않은 기존의 형태소 해석기와 제안하는 다중 스레드 방식을 적용한 형태소 해석기 간의 비교 실험을 실시하고, 그 결과를 바탕으로 효율성을 검증해 보도록 한다. 마지막으로 제 4 절에서 결론 및 추후 연구계획을 제시하는 것으로 마무리 짓는다.

Ⅱ. 다중 스레드 방식을 도입한 형태소 해석기

2.1 다중 스레드 방식 도입을 위한 고려사항

다중 스레드를 도입하기 위해서는 스레드 간의 동작 특성을 고려한 조치가 필요하다.

다중 스레드 시스템에서는 시스템 상의 자원, 즉 CPU, 메모리 등을 서로 다른 스레드 간에 공유하게 된다. 또한 스레드들은 병렬적으로 작업을 처리하기 때문에 동시에 같은 메모리 영역에 접근하는 경우, 자칫하면 예상치 못한 값이 야기될 수도 있다.

생각해야 할 문제점 또 한가지는 시스템 상의 자원이 한정되어 있다는 점이다. 다수의 스레드가 이러한 시스템 상의 한정된 자원을 동시에 사용하며 병렬적으로 작업을 수행하다보면, 작업의 부하가 커지거나 스레드의 개수가 증가됨에 따라 자원의 사용도가 높아짐으로 시스템이 다운되는 등의 치명적인 상황이 발생할 수도 있다. [4]-[10]

이러한 문제점들을 고려하여 기존의 형태소 해석기에 다중 스레드를 도입하기 위해 처리한 일을 크게 몇 가지로 요약하자면, 아래와 같다.

(1) 형태소 해석기에서 사용하는 전역 변수 및 크기가 큰 지역 변수에 대한 처리
(2) 형태소 해석기에서 사용하는 메모리의 최적화
(3) 스레드 간의 동기화 제어

먼저, (1)에 대한 처리로서, 메모리 동적 할당 기법을 이용하였다. 본 연구실에서 보유하고 있는 형태소 해석기의 변수 사용 현황을 살펴보면 전역변수로는 현재 분석 중인 어절에 대한 사전 정보 적재 관련 변수들, 형태소 해석을 실시할 때 빈번히 접근되어 사용되는 스택에 관련된 변수들이 큰 비중을 차지하고 있다. 특히 이러한 스택 관련 변수들은 형태소 해석이 수행되는 동안 계속 참조되고 변경되기 때문에 더욱 주의를 요하고, 형태소 해석기에서 핵심적인 부분을 차지하고 있다고 할 수 있다. 이러한 스택 관련 전역 변수들은 하나의 구조체로 형성하여 메모리 동적 할당을 해 주고, 이 변수들을 사용하고 있는 함수들에게 구조체의 포인터를 넘겨주는 방식으로 변경하였다.

(2)에 대한 처리로서, 일단 형태소 해석기 시스템의 분석 작업을 통하여 불필요하다고 생각되는 코드 및 사용하고 있지 않는 변수들을 모두 제거하는 작업을 선행하였다. 고정된 크기로 설정되어 있어 입력 어절의 크기에 제한을 받게 설정된 부분도 상황에 따라 동적 할당을 하게 수정하여 시스템의 융통성을 높이고자 하였다.

(3)에 대해서 살펴보면, 다수의 스레드들이 병렬적으로 작업을 처리하게 되므로 한 스레드가 값을 변경하고 있는 메모리 영역을 다른 스레드가 동시에 접근하였을 때 예상치 못한 결과가 발생될 수 있다.

형태소 해석기를 분석하여 이러한 가능성이 있는 부분을 찾아 동기화 기법을 적용하여 주어야 한다. 예를 들어, 처리 중인 입력 어절에 대한 사전의 정보를 검색하여 해당 정보를 저장하는 변수의 경우가 이에 해당한다.

동기화 기법에는 세마포어, 뮤텍스, 조건 변수 등의 방법이 있다. 우리는 세마포어 기법을 사용하였다.

2.2 다중 스레드 방식을 도입한 형태소 해석기

Ⅱ-1.에서 고려한 사항들을 반영한 본 논문에서 제안하는 다중 스레드 기능을 지니는 형태소 해석기의 흐름도는 아래 그림 1과 같다.

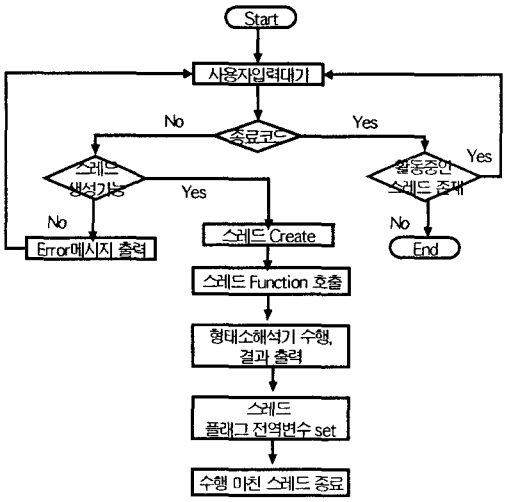

그림 1 다중 스레드 방식을 도입한 형태소 해석기 흐름도

스레드의 생성 개수가 늘어갈수록 한정된 시스템 자원으로 인하여 시스템의 불안정도가 높아지는 상황이 발생하였다. 이러한 상황을 예방하기 위해 스레드를 생성할 때 생성 가능한지의 여부를 체크하여 가능한 경우에만 생성해 나가도록 하였고, 그렇지 않은 경우에는 메시지를 띄워 사용자에게 알리도록 하였다.

또, 형태소 해석기의 실행을 종료하고자 하는 상황이 발생했을 경우, 진행 중인 작업은 마치고 끝을 내는 것이 바람직하다고 생각하였다. 이를 위해 스레드 관련 플래그 변수를 두어 종료 시에는 이 변수를 참조하여 현재 작업을 수행 중인 스레드가 있는지를 살핀 후, 모든 작업이 마무리 된 상태에서 전체 시스템을 종료할 수 있도록 하였다.

Ⅲ. 실험 및 결과 분석

형태소 해석기는 C언어로 이루어져 있으며, 형태소 해석기가 실험 된 환경은 아래와 같다.

- PentiumⅢ-450
- RAM 128M
- Linux 7.0 버전의 OS

실험의 내용은 다음과 같다.

우선 다중 스레드 방식을 도입한 형태소 해석기에서 사용자의 표준 입력에 대한 처리 능력을 살펴보았다.

입력의 크기를 256바이트 이내라고 제한하고, 사용자 표준 입력을 받아 형태소 해석을 수행하는 경우의 결과 예시는 그림 2와 같다.

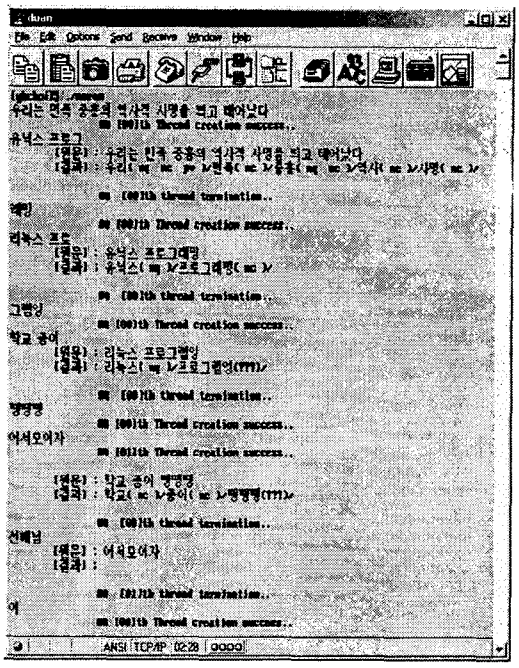

그림 2 다중 스레드 방식에서 표준 입력 처리

그림 2는 스레드의 생성 및 종료 과정을 알아보기 쉽게 하기 위해 임시로 표준 출력으로 진행 과정을 보여주게 한 화면이다.

화면에 출력되는 스레드 아이디는, 스레드들이 생성시와 소멸 시마다 값이 변경되는 스레드 정보 플래그 역할을 하는 전역변수 배열을 검사하여 작업이 끝난 스레드를 확인하고, 이 변수의 정보를 참조하여 새로운 스레드를 생성 시 설정하도록 하였다.

표준 입력의 경우에는 처리량이 작기 때문에 다중 스레드 방식을 도입한 형태소 해석기에서 문제없이 형태소 해석을 수행할 수 있었다. 또한 형태소 해석기 자체의 처리 속도가 빠르고, 기존의 다중 스레드를 도입하지 않은 형태소 해석기와 거의 다름없이 즉각즉각 결과를 반환하였기 때문에 표준 입력 처리에 대해서는 기존의 시스템과 다중 스레드 방식의 시스템을 비교하는 실험 데이터를 제시하지 않았다.

다음으로, 기존의 다중 스레드 방식을 도입하지 않은 형태소 해석기와 논문에서 제안하는 다중 스레드 방식을 도입한 형태소 해석기의 성능을 비교해 보기로 하였다.

이를 위한 실험으로 파일을 입력 단위로 형태소 해석을 실시하여 처리 속도를 비교하여 보기로 하였다. 처리할 입력 파일의 크기는 10KByte 미만에서 파일의

크기를 변화시켜 가면서 실험을 하고, 입력 파일의 수를 늘려 가면서 실험을 해 보았다. 입력된 파일의 개수만큼 스레드들이 동시에 생성되어 각자 자신의 입력 파일을 대상으로 병렬적으로 형태소 해석을 실시하게 된다.

참고로 본 연구실에서 보유하고 있는 형태소 해석기는 약 1MByte의 파일을 형태소 해석하는데 약 7.4초 정도의 시간이 걸린다.

실험을 실시한 결과는 표 1과 같다.

입력 / 시스템	2K/2개	2K/4개	2K/6개	…
기존의 형태소 해석기	0.06[sec]	0.11[sec]	0.15[sec]	…
다중 스레드 형태소 해석기	0.04[sec]	0.09[sec]	0.12[sec]	…
입력 / 시스템	2K/3개	4K/3개	8K/3개	…
기존의 형태소 해석기	0.08[sec]	0.14[sec]	0.26[sec]	…
다중 스레드 형태소 해석기	0.06[sec]	0.11[sec]	0.24[sec]	…

표 1 기존의 시스템과의 비교 실험

실험 결과를 살펴보면, 여러 입력을 처리하는데 있어서 입력의 처리량이 작을 경우에는 다중 스레드 방식을 도입한 형태소 해석기에서의 처리 속도가 기존의 형태소 해석기에서 보다 이득을 볼 수 있었다.

위의 표에서 제시하지는 않았지만, 입력 파일의 크기를 더욱 늘리고, 스레드의 생성 개수를 계속하여 늘려 가며 실험을 했을 경우에는 시스템이 불안해지는 상황도 발생하였다. 이는 메모리와 같은 시스템 상의 한정된 자원을 동시에 여러 스레드들이 이용하여 작업을 처리하고자 하기 때문에 발생하는 듯 하였다.

스레드의 개수를 계속하여 늘려갈 수록 처리속도가 이상적인 경우처럼 일정한 비율로 향상되어 나타나지를 않았는데, 이는 스레드의 개수가 늘어날수록 스레드 간 동기화를 위한 서로의 상태를 감시하고 필요시 대기하는 등의 동작이 점점 더 많은 비중을 차지하게 되기 때문이라고 볼 수 있다.

Ⅳ. 결론 및 추후 연구

본 논문에서는 정보 검색 시스템의 색인 시스템으로서 사용될 수 있는, 다중 처리가 가능한 형태소 해석기로서 다중 스레드를 도입한 형태소 해석기를 제안하였다.

다중 스레드 기법은 새로운 프로세서를 매번 생성하는 것보다 적은 자원 소모를 한다는 점이나, 프로세스 간 동기화보다 비교적 쉽게 스레드 간 제어를 할 수 있다는 점 등의 장점으로 인해 다중 처리를 요하는 시스템에서 매우 유용하게 여겨지고 있다.

반면 다수의 스레드들이 자원을 공유하기 때문에 발생하는 예기치 못한 상황 발생 가능성 때문에 이 기법을 부작용 없이 사용하기 위해서는 매우 조심스러운 프로그래밍 사고가 필요하였다.

실제 실험 결과, 작은 크기의 입력에 대한 처리에 있어서는 문제가 없었으나, 스레드 생성 개수가 커지고 처리량이 커질수록 예상했던 만큼의 처리 향상을 보이지 않았다. 이는 스레드 간 동기화를 위한 제어의 영향도 있겠고, 한정된 자원이 존재하는 환경하에서 한꺼번에 많은 수의 스레드들이 자원을 공유하여 작업을 처리하여야 하기 때문이기도 하다.

본 논문에서 소개한 다중 스레드 기법에 대한 개념 및 다중 스레드 방식을 도입하기 위해 고려해야 하는 사항들, 간단한 실험 결과를 바탕으로 다중 스레드를 도입한 형태소 해석기의 성능이 개선될 수 있도록 원인 분석 및 수정을 가할 것이다.

나아가서는 실제 정보 검색 시스템의 색인기로서 형태소 해석기가 사용될 수 있도록 다량의 웹문서를 입력으로 하는 테스트를 수행하여 시스템을 검증해 볼 계획이다.

참고문헌

[1] 최신 정보검색 시스템의 특성 및 동향, KOSTI 2000 워크샵 한글정보검색 학술발표 논문집, pp.3-30, 2000.12

[2] 통합 정보 검색을 위한 과학기술문서 색인 및 요약 시스템의 개발, KOSTI 2000 워크샵 한글정보검색 학술발표 논문집, pp.116-133, 2000.12

[3] Beginning Linux Programming, Matthew, Neil, 정보문화사, 2000

[4] http://it.soongsil.ac.kr/

[5] http://fox2000.com.ne.kr/info/thread.html

[6] http://namhae.duksung.ac.kr/~ucpark/c-program/node29.html

[7] http://namhae.duksung.ac.kr/~ucpark/c-program/node30.html

[8] http://cosmos.soongsil.ac.kr/course/os/os1997-2/thread.html

[9] http://cse.sch.ac.kr/~doh/os/hello_world.html

[10] http://unix.or.kr/oldstudy/network/13-1.HTM