2018 November 11

김 경훈

201533695 컴퓨터공학과

데이터베이스 박양재 교수님

페이징 교체 알고리즘과 트리탐색 알고리즘의 종류와 기능

페이지 교체 알고리즘

페이지 교체 알고리즘이란 페이징 기법으로 메모리를 관리하는 운영체제에서, 페이지 부재가 발생하여 새로운 페이지를 할당하기 위해 현재 할당된 페이지 중 어느 것과 교체할 지를 결정하는 방법이다. 이때 페이징 기법이란 컴퓨터가 메인 메모리에서 사용하기 위해 2차 기억 장치로부터 데이터를 저장하고 검색하는 메모리 관리 기법이다. 이때의 일정한 크기를 가진 블록을 페이지라 하며, 메모리에 적재 된 페이지중에 사용 페이지가 없는 경우를 페이지 부재라고 한다. 페이지 부재가 발생하여 새로운 페이지를 적재해야 하는데 적재할 공간이 없어 이미 적재된 페이지 중 하나를 교체해야 할 때 이 페이지 교체 알고리즘이 사용된다.

1. FIFO 알고리즘

FIFO 알고리즘은 FIRST-IN FIRST-OUT이다. 말그대로 메모리에 먼저 올라온 페이지를 먼저 내보낸다는 것이다. 이 방법은 페이징 교체 알고리즘 중 가장 간단한 방법으로 이미 할당된 페이지 프레임에 모두 페이지가 적재되 있는 상황에서 새로운 페이지가 적재되어야 할 때 주기억장치에 적재되 있는 페이지 중 가장 오래된 페이지부터 교체한다. FIFO의 알고리즘은 프레임의 수가 적을수록 페이지 결함이 더 많이 일어나게 된다. 계속 교체를 해주어야 하기 때문이다. 하지만 프레임의 수가 많아질수록 페이지 결함의 횟수는 감소하게 된다. FIFO는 간단하지만, 페이지가 오래 있었던 이유 하나만으로 교체되므로 불합리하다는 단점이 있다. 왜냐하면 오래 있었던 페이지는 앞으로 계속 사용될 가능성이 높기 때문이다.

1. OPT 알고리즘

앞으로 가장 사용하지 않을 페이지를 가장 우선적으로 내려 보내는 알고리즘. 가장 사용하지 않게 될 페이지를 예측하는 것은 거의 불가능하여, 예측이 가능한 특정 상황에서만 사용 가능. OPTimal한 알고리즘이라는 이름 답게 가장 적절한 알고리즘이라고 할 수도 있지만, 비현실적이라는 말이 많음

1. LFU 알고리즘

LFU 알고리즘은 Lease Frequently Used 알고리즘으로 주기억장치에 적재되어 있는 페이지들에 대해 참조된 횟수를 기준으로 교체될 페이지를 선정하는 기법이다. 페이지 프레임 모두에 페이지들이 적재되어 있는 상황에서 새로운 페이지가 적재되어야 할 때 주기억장치에 적재된 페이지 중 가장 참조 횟수가 적은 페이지를 교체한다. 교체 대상이 여러 개일 경우, 가장 오랫동안 참조되지 않은 페이지를 교체한다(LRU알고리즘 적용).

참조될 가능성이 많음에도 불구하고 횟수에 의한 방법이므로 최근에 들어온 프로그램이 쓸모가 많아도 교체시킬 가능성이 있다.

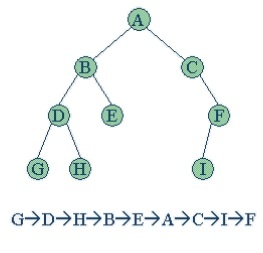
1. LRU 알고리즘

LRU 알고리즘은 Lease Recently Used 알고리즘으로 페이지 프레임에 페이지가 모두 적재된 상황에서 새로운 페이지가 적재되어야 할 때, 주기억장치에 적재되어 있는 페이지 중 최근 가장 오랫동안 참조되지 않은 페이지를 교체한다. 최근에 참조된 페이지는 앞으로도 참조될 가능성이 높다고 판단하여, 페이지를 교체하는 알고리즘. 프로세스가 주기억장치에 접근할 때 마다 참조된 페이지에 대한 시간을 기록해야 하여 큰 오버헤드가 발생한다. OPT알고리즘은 미래를 보고 판단하지만 LRU 알고리즘은 과거를 보고 판단하므로 실질적으로 사용 가능한 알고리즘이다.

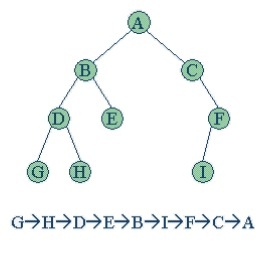
트리 탐색 알고리즘

트리 탐색 문제는 그래프나 트리로 표현되는 자료구조에서 목적지를 찾아가기 위한 경로를 구하는 것이다. 트리 탐색 알고리즘은 그 경로를 찾아가는 알고리즘을 말하는데, 대부분의 트리 탐색 알고리즘에서 트리는 이진 트리로 표현이 된다. 그 이유는 이진 트리가 가장 알고리즘을 표현하기 쉽기 때문이다. 하지만 이 알고리즘 들은 다른 모든 트리에서도 일반화될 수 있다.

트리는 선형이 아닌 비선형 구조이기 때문에 모든 노드를 거쳐가기 위한 방법이 필요하다. 그리하여 4가지 순회 방법이 나오게 되었다. 반대로 트리가 아닌 list같은 경우 선형 구조이기 때문에 특별한 노드를 거쳐가는 방법을 필요로 하지 않는다.

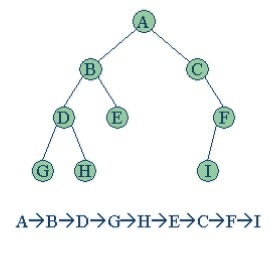
1. 중위 탐색 알고리즘

중위 탐색은 왼쪽 트리-> 루트 노드 -> 오른쪽 트리 순으로 방문한다. 이때 왼쪽 트리를 방문 할 때 왼쪽 트리 역시 트리기 때문에 이 트리 안에서도 중위 탐색 한다. 이렇게 중위 탐색을 반복하는 순환 알고리즘의 형태이다. 그림에서처럼 왼쪽의 왼쪽 트리를 중위 탐색하며 계속 순환-> 루트 노드 -> 오른쪽 트리 순으로 방문한다.

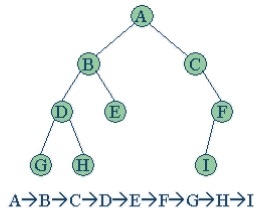
1. 후위 탐색 알고리즘

후위 탐색은 왼쪽 트리-> 오른쪽 트리-> 루트 노드 순으로 방문한다. 마찬가지로 왼쪽 트리를 방문할 때 왼쪽 트리 자체가 트리이기 때문에 다시 후위 탐색을 한다. 이것 역시 순환 알고리즘이라고 볼 수 있다.

1. 전위 탐색 알고리즘

전위 탐색은 루트노드->왼쪽트리->오른쪽 트리 순으로 방문하게 된다.

마찬가지로 왼쪽 트리도 트리이기 때문에 왼쪽 트리에서도 다시 전위 탐색을 한다. 위의 2가지와 마찬가지로 순환 알고리즘으로 볼 수 있다.

1. 레벨 탐색 알고리즘

레벨 탐색 알고리즘은 트리마다 레벨1->레벨2-> 레벨3 -> …. 레벨 n순으로 탐색을 한다. 레벨 n으 탐색할 때 방문 노드의 자식 노드를 저장을 해놓은 다음 n+1레벨을 방문하면 저장해놓은 노드를 꺼내어 방문한다. 그림에서는 A가 레벨 1 노드로 그 자식 노드를 저장 해 두었다가 (B,C) 레벨 2를 방문할 때 저장해 놓았던 B,C를 꺼내어 방문한다.

출처

<https://ko.wikipedia.org/wiki/%ED%8E%98%EC%9D%B4%EC%A7%80_%EA%B5%90%EC%B2%B4_%EC%95%8C%EA%B3%A0%EB%A6%AC%EC%A6%98>(페이징 교체 알고리즘의 개념)

<http://copycode.tistory.com/122>(페이징 알고리즘의 종류)

[https://m.blog.naver.com/PostView.nhn?blogId=kyung778&logNo=60164009610&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F (LRU](https://m.blog.naver.com/PostView.nhn?blogId=kyung778&logNo=60164009610&proxyReferer=https%3A%2F%2Fwww.google.co.kr%2F%20(LRU)기법에대한 참고)

<http://3dmpengines.tistory.com/423> 트리 탐색 알고리즘 그림 + 내용 출처