Chapter 11. 프로세스간 통신(Inter Process Communication) 내용 확인 문제

1. 프로세스간 통신이 의미하는 바는 무엇인가? 이를 개념적으로, 그리고 메모리의 관점에서 각각 설명해 보자.

프로세스간 통신은 두 프로세스가 서로 데이터를 주고 받을 수 있음을 의미한다. 프로세스는 서로 다른 메모리 구조를 같기 때문에 공유하는 메모리가 존재하지 않는다. 이에 대해서 ‘파이프(pipe)’라는 통신 기법을 활용하여 파이프의 입출력 파일 디스크립터를 사용해 데이터를 쓰고 읽음으로써 데이터를 주고 받을 수 있게 됨을 의미한다.

1. 프로세스간 통신에는 IPC라는 별도의 메커니즘이 요구된다. 그리고 이는 운영체제에 의해서 지원되는 별도의 기능이다. 그렇다면 프로세스간 통신에 있어서 이렇듯 운영체제의 도움이 필요한 이유는 무엇인가?

프로세스간의 메모리 구조를 살펴보면, 서로 공유하는 메모리가 존재하지 않는 독립적인 관계를 이루고 있다. 이러한 상황에서 프로세스간 데이터를 주고 받기 위해서는 ‘파이프(pipe)’ 통신이 필요하다. 이는 pipe( ) 함수 호출을 하여 운영체제의 자원인 pipe에 접근이 가능하게 된다. 파이프의 생성은 프로세스에서 이루어지는 것이 아니라 운영체제가 생성하며, 이는 운영체제의 시스템 자원이다. 따라서 두 프로세스 간의 통신이 이루어지기 위해 운영체제의 도움이 필요하게 된다.

1. 대표적인 IPC 기법으로 ‘파이프(pipe)’라는 것이 있다. 파이프의 IPC 기법과 관련해서 다음 질문에 답해보자.
2. 파이프는 프로세스간에 데이터를 송수신하는 경로를 의미한다. 그렇다면 이 경로는 어떻게 해서 생성되며, 누구에 의해서 만들어지는가? Pipe( ) 함수 호출에 따라, 운영체제가 생성한다.
3. 프로세스간 통신을 위해서 통신의 주체가 되는 두 프로세스 모두 파이프에 접근이 가능해야 한다. 그렇다면 하나의 파이프에 두 프로세스는 어떻게 해서 모두 접근이 가능한가?

동일한 작업을 수행하는 프로세스는 복제하는 경우, 파이프는 운영체제의 시스템 자원이므로 복제의 대상이 되지 않는다. 다만, 프로세스는 파이프에 접근 가능한 2개의 파일 디스크립터를 pipe( ) 함수의 호출 결과로 얻게 된다. 이후 fork( )를 진행하여 복제하게 되면, 두 프로세스가 2개의 파일 디스크립터를 통해 파이프에 데이터를 읽고 쓰기 할 수 있다.

1. 파이프는 두 프로세스간에 양방향 통신이 가능하게 한다. 그렇다면 양방향 통신을 진행하는데 있어서 특히 주의해야 할 사항은 무엇인가?

프로세스가 파이프에 데이터를 쓰게 되면, 파이프에 남은 데이터는 어느 프로세스라도 read( )를 통해 읽어 들일 수 있다. 이것은 데이터를 write( )한 프로세스 조차도 자신이 입력한 데이터를 읽어들일 수 있음을 의미한다. 그렇기 때문에, 단일 파이프를 사용하여 통신하는 경우, 프로세스의 흐름을 잘 분석하여 적절한 시점에 write(), read() 가 이루어져야 한다.

1. IPC 기법을 확인하는 차원에서, 두 프로세스 사이에서 총 3회에 걸쳐서 문자열을 한 번씩 주고받는 예제를 작성해 보자. 물론 두 프로세스는 부모, 자식의 관계로 형성이 되며, 주고 받을 문자열의 종류는 프로그램상에서 여러분이 임의로 결정하기 바란다.

텍스트이(가) 표시된 사진

자동 생성된 설명