

SYSProg HW7-2

-IPC 기법을 이용한 프로세스 간 데이터 교환-

학 과	컴퓨터공학과
학 번	201211704
이 름	김기홍
제 출 일	2015.11.23

목 차

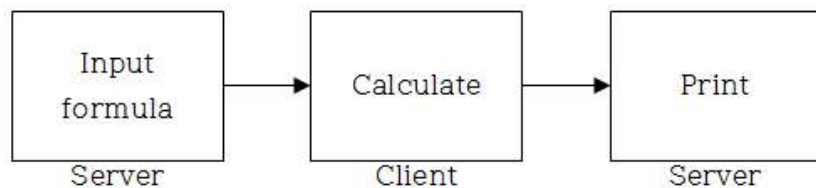
1. 개요

2. 구현

- Message Queue 방식
- Shared Memory 방식

1. 개요

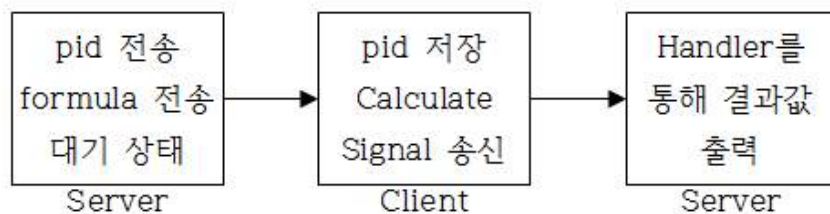
- HW7-1과 같은 기능을 하는 프로그램이며, 다른점은 IPC 기법이 다르다는 점이다. 이번 HW7-2에서의 IPC 기법은 유닉스 시스템 V에서 제공하는 기법이며 그 중에서 Message Queue 방식과 Shared Memory 방식을 이용하여 상호 프로세스간 데이터를 주고 받는 작업을 구현하였다. 프로그램은 Server 와 Client 로 나뉘며, Server은 계산 식(Formula)를 입력받고 그 식을 Client로 넘기고, Client는 그 식을 계산하여 결과값을 다시 Server로 넘기면 Server는 그 결과값을 출력하는 구조이다.



2. 구현

- Message Queue 방식 : 이 방식은 파이프와 유사하지만 파이프와는 달리 메시지 단위로 동작하고, 메시지의 최대 크기가 제한 되어있다. 하지만 운영체제에서 따로 제공하는 기법이기에 때문에 메시지 큐 생성 후, 어떤 프로세스든 간에 같은 키 값으로 생성한 메시지 큐라면, 손쉽게 메시지 (데이터)를 공유할 수 있다. msgsnd()와 msgrcv()를 통해 메시지 송신 및 수신이 쉽게 가능하다. 필자가 Message Queue 방식으로 구현한 방법은 서로 다른 프로세스간 시그널을 송신해야 했는데 이를 위해 다른 프로세스의 pid를 알아야만 했다. 필자는 그 방법 또한 Message Queue 방식을 이용했다.

쉽게 정리해서 말하자면, Server 프로그램에서 getpid()를 사용하여 Message Queue에 전송하고, 그 후에 사용자로부터 입력받은 식을 전송하였다. 그리고 시그널을 받을 때까지 대기상태로 두었다. 다시 Client 프로그램에서는 Message Queue에서 데이터를 꺼내어 pid 변수에 저장하고, 식을 이용하여 값을 계산하여 Message Queue에 전송 후, 앞서 얻은 pid와 kill()을 이용해 시그널을 보내고 종료하는 형태로 구현하였다. 간단하게 그림으로 살펴보자.



- Shared Memory 방식 : 이 방식은 같은 메모리 공간을 두 개 이상의 프로세스가 공유할 수 있으며 같은 메모리 공간을 사용하므로 이를 통해 데이터를 주고 받을 수 있다. Message Queue와 유사한 방법으로 같은 키값을 이용해 생성한 공유 메모리라면 아주 쉽게 데이터를 주고 받을 수

있다.(여기서는 주고 받는다기 보다는 같은 공간을 쓰기 때문에 한쪽에서 변경된 값이 다른 한쪽에서도 변경되어 사용할 수 있다는 표현이 더 맞는 것 같다.) 여기서도 위의 Message Queue방식을 구현할 때와 비슷하게 구현하였다. 다만 조금 다른 점은 시그널을 송신하기 위한 pid를 따로 받지 않고, struct shmid_ds형의 변수에 연결된 다른 프로세스의 정보를 담아 쉽게 pid를 얻을 수 있었다. 만약 구조체 변수의 이름이 shm_stat라면 pid값은 shm_stat.shm_cpid로 구할 수 있다. 나머지 세세한 구현방법은 위와 동일하므로 생략하도록 하겠다.