

프로세스 간의 통신 기법

유튜브 : <https://youtu.be/UoPgQPOjoUo>

사이버보안트랙 1971362 이준희

<목차>

1. 시스템 V IPC의 공통 요소

2. 메시지 큐 (Message Queue)

3. 공유 메모리 (Shared Memory)

4. 세마포어 (Semaphore)



프로세스 간 '통신 기법'이 필요한 이유

- fork()를 하여 부모 프로세스와 자식 프로세스를 생성하여도
- 메모리 주소공간이 다른 독립된 메모리 공간을 가진다.
- => 그러므로, 부모, 자식 프로세스 간에도 전역변수 및 지역변수를 공유하지 않는다.
- 그러므로 프로세스 간에 통신을 하기 위해서는
- 통신 기법이 필요하다.

프로세스 통신 기법의 종류

- 1. 시그널 (Signal)
- 2. 파이프 (Pipe)
- 3. 메시지 큐 (Message Queue)
- 4. 공유메모리 (Shared memory)
- 5. 세마포어 (Semaphore)
- 6. 소켓 (Socket)



시스템 V IPC의 공통 요소

- 시스템 V IPC(Interprocess Communication)
- 유닉스(UNIX)는 크게 BSD 계열과 시스템 V 계열로 구분할 수 있음
- 메시지 큐, 공유 메모리, 세마포어 세개를 묶어 시스템 V IPC라고 함
- 시스템 V 계열 유닉스에서 개발해 제공하는 프로세스 간 통신 방법
- 리눅스: 시스템 V 계열에 속하는 운영체제
- 시스템 V는 AT&T 벨 연구소에서 개발한 UNIX 운영체제의 버전 중 하나로, 리눅스 커널은 시스템 V의 특징을 일부 포함

IPC 객체

- 시스템 V IPC를 사용하려면 IPC 객체를 생성해야 하는데 이를 위해 공통으로 사용하는 기본 요소가 **키와 식별자(ID)**
- 객체를 생성하고 현재 사용 중인 각 IPC의 상태를 확인하고 사용을 마친 객체는 삭제할 수 있도록 관리 명령을 제공

표 11-1 키 생성 및 IPC 관리 명령

기능	함수 및 명령
상수	IPC_PRIVATE
키 생성	key_t ftok(const char *pathname, int proj_id);
IPC 정보 검색	ipcs [-ihVmqsac1ptu]
IPC 생성	ipcmk [options]
IPC 삭제	ipcrm [options]

메시지 큐 (Message Queue)

- 시스템 V IPC 중 하나로, 프로세스 통신 기법 중 하나.
- 메시지 큐는 파이프와 유사하나 파이프는 '스트림' 기반으로 동작하고 메시지 큐는 '메시지(또는 패킷)' 단위로 동작
- 우편함처럼 메시지 큐를 만든 후 이를 통해 메시지를 주고받음
- 메시지 큐는 '커널'에 존재

표 11-2 메시지 큐 함수

기능	함수
메시지 큐 식별자 생성	<code>int msgget(key_t key, int msgflg);</code>
메시지 전송	<code>int msgsnd(int msqid, const void *msgp, size_t msgsz, int msgflg);</code>
메시지 수신	<code>ssize_t msgrcv(int msqid, void *msgp, size_t msgsz, long msgtyp, int msgflg);</code>
메시지 제어	<code>int msgctl(int msqid, int cmd, struct msqid_ds *buf);</code>

공유 메모리 (Shared Memory)

- 시스템 V IPC 중 하나로, 프로세스 통신 기법 중 하나.
- 한 프로세스의 일부분을 다른 프로세스와 공유하는 것을 의미
- 메모리의 일부 공간을 두 독립적인 프로세스에서 공유하고 해당 메모리를 통해 데이터를 주고받을 수 있음

표 11-3 공유 메모리 함수

기능	함수
공유 메모리 식별자 생성	<code>int shmget(key_t key, size_t size, int shmflg);</code>
공유 메모리 연결	<code>void *shmat(int shmid, const void *shmaddr, int shmflg);</code>
공유 메모리 연결 해제	<code>int shmdt(const void *shmaddr);</code>
공유 메모리 제어	<code>int shmctl(int shmid, int cmd, struct shmid_ds *buf);</code>

세마포어 (Semaphore)

- 시스템 V IPC 중 하나로, 프로세스 통신 기법 중 하나.
- 프로세스 사이의 동기를 맞추는 기능을 제공
- 공유 메모리에 여러 프로세스가 동시에 쓰기를 시도하면 데이터가 손상되는 현상이 발생
- 따라서 여러 프로세스 사이에서 동작 순서를 지정해줘야 하는데, 세마포어가 접근 순서를 정하는 방법을 제공

표 11-4 세마포어 함수

기능	함수
세마포어 생성	<code>int semget(key_t key, int nsems, int semflg);</code>
세마포어 제어	<code>int semctl(int semid, int semnum, int cmd, ...);</code>
세마포어 연산	<code>int semop(int semid, struct sembuf *sops, size_t nsops);</code>

Q & A

