인덕대학교

스케줄링 및 IPC
- IPC

컴퓨터소프트웨어학과

김병국 교수

학습목표

- 프로세스간 통신을 위한 운영체제의 역할을 알 수 있다.
- IPC기능을 위한 기법들을 알 수 있다.
- 실습을 통해 몇가지 IPC 기술을 안다.

목차

1. 프로세스 간 통신의 개념 (1/2)

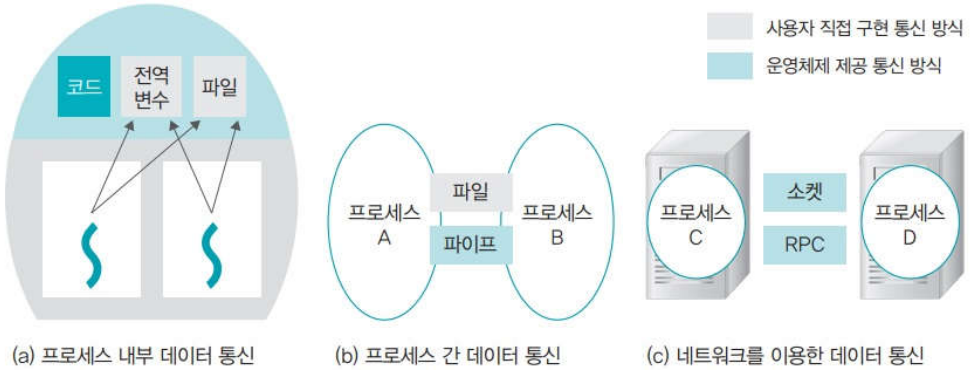
❑정의

- IPC : Inter-Process Communication
- 프로세스간 데이터를 공유하기 위한 방법
- 운영체제는 프로세스의 관리 및 각각의 동작을 보호
  - 프로세스는 자신의 할당메모리 영역 외에는 접근 불가
  - 타프로세스의 영역을 독단적으로 접근할 수 없음
- 프로세스간 통신을 위해 운영체제는 몇가지 자원을 제공
  - 예: 시그널, 인터럽트, 공유 파일, 파이프, 공유메모리, 메시지큐 등

1. 프로세스 간 통신의 개념 (2/2)

❑ 프로세스간 통신(IPC)

- 프로세스 내부 데이터 통신
  - 프로세스 내 스레드간 통신
    - 스레드는 **전역 변수**나 **파일**을 이용하여 데이터를 공유



(a) 프로세스 내부 데이터 통신 (b) 프로세스 간 데이터 통신 (c) 네트워크를 이용한 데이터 통신

- 프로세스 간 데이터 통신
  - 같은 컴퓨터(동일 호스트)에 있는 프로세스간 통신
  - **공용 파일** 또는 운영체제가 제공하는 **자원**을 이용

- 네트워크를 이용한 데이터 통신
  - 여러 컴퓨터가 네트워크로 연결되어 있을 때 통신
  - 소켓을 이용한 데이터 공유

3. IPC 기법 (1/8)

❑종류

- 시그널
- 인터럽트
- 공유 파일
- 파이프
- 공유메모리
- 메시지큐

3. IPC 기법 (2/8)

❑ 시그널(Signal)

- 운영체제는 프로세스를 관리하기 위한 시그널을 정의
- 시그널을 수신한 프로세스는 그에 맞는 행위를 수행
  - 예1: 9(SIGKILL)는 프로세스를 강제 종료
  - 예2: 14(SIGALRM)은 잠들어있는 프로세스를 깨움

```
[kali@kali:~]$kill -l
 1) SIGHUP       2) SIGINT       3) SIGQUIT      4) SIGILL       5) SIGTRAP
 6) SIGABRT      7) SIGBUS       8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGALRM     15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
[kali@kali:~]$
```

① **sleep** 명령 실행

```
[kali@kali:~]$sleep 100
Alarm clock
[kali@kali:~]$
```

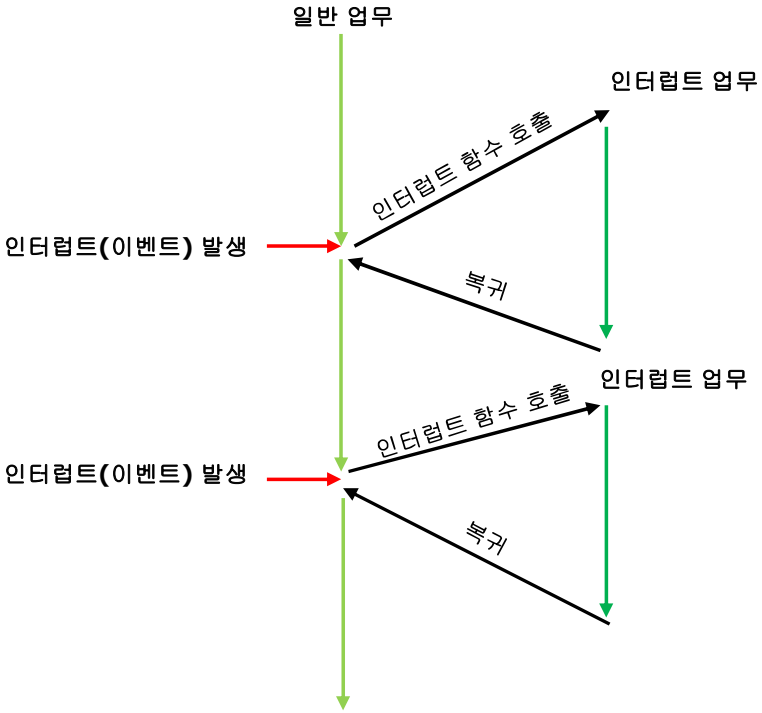
③ 깨어남 확인

```
[kali@kali:04_1]$ps f
    PID TTY      STAT   TIME COMMAND
   1268 pts/1    Ss     0:00 /usr/bin/bash
   1451 pts/1    S+     0:00  \_ sleep 100
   1139 pts/0    Ss     0:00 /usr/bin/bash
   1452 pts/0    R+     0:00  \_ ps f
[kali@kali:04_1]$kill -14 1451
[kali@kali:04_1]$
```

② **sleep** 프로세스에 **14**번 시그널 전송

❑ 인터럽트

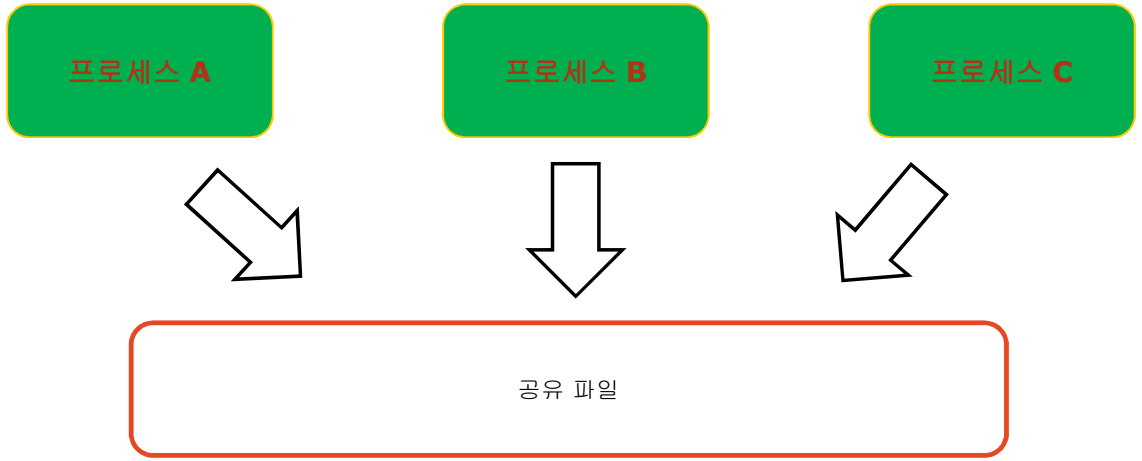
- 이벤트 발생에 따른 처리를 위한 방식
- 프로세스가 동작 중 인터럽트가 발생하면 해당 기능을 수행



3. IPC 기법 (4/8)

❑ 공유 파일

- 서로 다른 프로세스가 동일한 파일을 공유하여 사용하는 방식
- 프로세스 처리 성능은 기록 장치의 읽기/쓰기 속도에 영향을 받음



3. IPC 기법 (5/8)

❑ 파이프

- A 프로세스의 출력을 B프로세스의 입력으로 처리
- 실습 예: 특정프로그램의 출력 결과를 입력을 받아서 출력

[파일명: pipe_r.c]

```c
#include <unistd.h>
#include <stdio.h>
#include <string.h>

int
main(void)
{
    char pBuffer[BUFSIZ];
    int nLen = 0;

    do
    {
        memset(pBuffer,0,BUFSIZ);
        nLen = read(0, pBuffer, BUFSIZ);
        if(nLen>0){
            printf("%s", pBuffer);
        }
        else
            break;
    } while(1);

    return 0;
}
```

[결과]

```
[kali@kali:04_1]$gcc pipe_r.c -o pipe_r
[kali@kali:04_1]$ls -l | ./pipe_r
total 52
-rwxr-xr-x 1 kali kali 16736 Mar 17 20:32 a.out
-rw-r--r-- 1 kali kali   155 Mar 17 08:16 job.c
-rwxr-xr-x 1 kali kali 16712 Mar 17 21:06 pipe_r
-rw-r--r-- 1 kali kali   275 Mar 17 21:06 pipe_r.c
-rw-r--r-- 1 kali kali   277 Mar 17 21:06 pipe_r.c~
[kali@kali:04_1]$
```

3. IPC 기법 (6/8)

❑공유메모리

- 운영체제에서 별도로 할당해준 공유 영역을 활용
- 각 응용프로그램을 해당 영역을 포인터로 접근
- 공유메모리 확인 명령: ipcs

❑ 메시지큐

- 큐(Queue) 방식으로 데이터를 접근
- 큐는 행위에 따라 En-queue와 De-queue로 분류
  - En-queue는 큐에 데이터를 삽입
  - De-Queue는 큐에서 데이터를 추출

3. IPC 기법 (8/8)

❑ 공유메모리 실습

- 관련 명령어: ipcs, ipcmk, ipcrm
  - Ipcs : IPC관련 운영체제내 상태를 확인
  - Ipcmk : IPC관련 자원 생성
  - Ipcrm : IPC관련 자원 삭제

```
[kali@kali:04_1]$ipcmk -M 1024
Shared memory id: 32768
[kali@kali:04_1]$ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x323b47aa 32768      kali       644        1024       0
0x00000000 14         kali       600        524288     2          dest
0x00000000 15         kali       600        2097152    2          dest
```

```
[kali@kali:04_1]$ipcrm -m 32768
[kali@kali:04_1]$ipcs -m

------ Shared Memory Segments --------
key        shmid      owner      perms      bytes      nattch     status
0x00000000 14         kali       600        524288     2          dest
0x00000000 15         kali       600        2097152    2          dest
0x00000000 16         kali       600        524288     2          dest
0x00000000 22         kali       600        524288     2          dest
```

```
[kali@kali:04_1]$ipcs -mp

------ Shared Memory Creator/Last-op PIDs --------
shmid      owner      cpid       lpid
14         kali       956        573
15         kali       956        573
16         kali       947        1262
22         kali       959        573
24         kali       964        573
30         kali       965        573
33         kali       966        573
37         kali       1003       573
39         kali       963        573
```

4. 방향성에 따른 분류

❑ 단방향 통신

- 한쪽 방향으로만 데이터를 전송할 수 있는 구조
- 대표적 예: 파이프(PIPE) 기법이 있음

❑ 양방향 통신

- 데이터를 동시에 양쪽 방향으로 전송할 수 있는 구조
- 대표적 예: 소켓 통신

❑ 반양방향 통신

- 양쪽 방향으로 전송이지만 동시 전송은 불가
- 특정 시점에 한쪽 방향으로만 전송할 수 있는 구조
- 대표적 예: 메시지 큐, 공유메모리

5. 동시성에 따른 분류

❑대기가 있는 통신

- 동기화를 지원하는 통신 방식
- 데이터를 받는 쪽은 데이터가 **도착할 때까지 자동 대기 상태(블록킹, Blocking)**

❑대기가 없는 통신

- 동기화를 지원하지 않는 통신 방식
- 다른 작업을 하는 중에 데이터가 수신되면 처리
  - 인터럽트(Interrupt, 이벤트) 처리 방식
  - 폴링(Polling) 처리 방식

수고하셨습니다.