

UNIX 프로그래밍 실습 (2019/11/21)

1. 메모리 매핑을 이용하여 통신하는 두 개의 프로그램을 작성 합니다. 두 프로그램 모두 “data1” 파일을 메모리 매핑 합니다. 한 프로그램은 외부 입력을 읽어서 매핑된 메모리에 저장 하는 작업을 3회 실행 합니다. 다른 프로그램은 매핑된 메모리의 내용을 출력하는 작업을 3회 실행 합니다. 두 프로그램 사이의 읽기-쓰기 동기화를 위해 semaphore를 사용 합니다.

<pre>int main(void){ int fd, i, n=0; char *addr; fd=open("data1", O_RDWR O_CREAT, 0600); addr=mmap(NULL, 512, PROT_READ PROT_WRITE, MAP_SHARED, fd, 0); ftruncate(fd, 512); for (i=0; i<3; i++){ n=n+read(0, addr+n, 50); } exit(0); }</pre>	<pre>int main(void){ int fd, i, n=0, semid; char *addr; fd=open("data1", O_RDWR O_CREAT, 0600); addr=mmap(NULL, 512, PROT_READ PROT_WRITE, MAP_SHARED, fd, 0); ftruncate(fd, 512); for (i=0; i<3; i++){ n=n+write(1, addr+n, 50); } exit(0); }</pre>
--	--

2. 네 개의 프로세스가 동기화를 하며 자신의 프로세스 id를 5회 출력하는 프로그램을 작성 합니다. 이 프로그램은 main() 함수의 arguments로 동기화에 참여하는 전체 프로세스 중 자신의 출력 순서를 입력 받습니다. 프로그램이 시작되면, 순서대로 자신의 프로세스 id를 출력 합니다. 동기화 작업은 semaphore를 사용하여 수행 합니다.