

10주차 1차시 메모리 관리

【학습목표】

1. 메모리 관리 기법을 설명할 수 있다.
2. 메모리 관리 함수를 사용할 수 있다.

학습내용1 : 메모리 관리

1. 프로세스간 통신(IPC : InterProcess Communication)

* 동일한 유닉스/리눅스 시스템에서 수행 중인 프로세스끼리 데이터를 주고받는 것
 파이프(pipe) 특수 파일 이용
 메모리 매핑이나 공유 메모리 영역을 이용
 유닉스 V : 메시지 큐, 공유 메모리, 세마포어 등

* 넓은 의미

종료 상태와 시그널 같은 정수값을 주고 받는 것
 시그널 : 미리 정의된 상수를 프로세스간 주고받는 것

* 리눅스에서 IPC

① 이름 없는 익명의 파이프

기능	함수원형
간단한 파이프 생성	FILE *popen(const char *command, const char *mode); int pclose(FILE *stream)
복잡한 파이프 생성	int pipe(int fildes[2])

② 이름 있는 익명의 파이프

기능	함수원형
FIFO 생성 명령	mknod name p mkfifo [-m mode] path...
FIFO 생성 함수	int mknod(const char *path, mode_t mode, dev_t dev); int mkfifo(const char *path, mode_t mode);

2. 네트워크를 이용한 통신

TCP/IP 프로토콜을 기본으로 소켓 라이브러리 이용
 TLI(Transport Layer Interface)

학습내용2 : 메모리 관리 함수

1. 메모리 매핑 관련 함수

기능	함수원형
메모리 매핑	<code>void *mmap(void *addr, size_t len, int prot, int flags, int fildes, off_t off);</code>
메모리 매핑 해제	<code>int munmap(void *addr, size_t len);</code>
파일 크기 조정	<code>int truncate(const char *path, off_t length);</code> <code>int ftruncate(int fildes, off_t length);</code>
매핑된 메모리 동기화	<code>int msync(void *addr, size_t len, int flages);</code>

2. 메모리 매핑 : mmap(2)

- fildes가 가리키는 파일에서 off로 지정한 오프셋부터 len크기만큼 데이터를 읽어 addr가 가리키는 메모리 공간에 매핑
- 성공 시 : 매핑된 메모리 시작 주소 리턴
- 실패 시 : 상수 MAP_FAILED 리턴

3. 메모리 매핑 해제 함수

- * 메모리 매핑 해제: munmap(2)
- addr이 가리키는 영역에 len 크기만큼 할당해 매핑한 메모리 해제
- 해제한 메모리에 접근하면 SIGSEGV 또는 SIGBUS 시그널 발생

【학습정리】

1. 프로세스간 통신

- 프로세스간 통신(IPC : InterProcess Commnunication)란 유닉스/리눅스 시스템안에서 수행중인 프로세스끼리 데이터를 주고 받는것이다.
- 메모리 매핑, 파이프, 메시지 큐, 공유 메모리, 세마포어 등의 방법이 있다.

2. 메모리 관리 함수

기능	함수원형
메모리 매핑	<code>void *mmap(void *addr, size_t len, int prot, int flags, int fildes, off_t off);</code>
메모리 매핑 해제	<code>int munmap(void *addr, size_t len);</code>
파일 크기 조정	<code>int truncate(const char *path, off_t length);</code> <code>int ftruncate(int fildes, off_t length);</code>
매핑된 메모리 동기화	<code>int msync(void *addr, size_t len, int flages);</code>

3. RAM (Random Access Memory)

- * 휘발성 메모리, 읽기 쓰기가 가능
- * 종류 :
 - SRAM(Static RAM)
 - DRAM(Dynamic RAM)
 - SDRAM(Synchronous RAM)
 - DDRSDRAM(Double Data rate SDRAM, 보통 DDR램이라 함)
 - DDR2RAM, RDRAM(Rambus RAM)