운영체제 13주차

20135151 이갑성

|  |
| --- |
| [lab11\_3.c]  #include <stdio.h>  #include <string.h>  #include <unistd.h>  #include <sys/types.h>  #include <sys/ipc.h>  #include <sys/shm.h>  #include <stdlib.h>  #define SIZE 1024  int main()  {  struct check\_data{  int check;  char data[SIZE];  } \*shared\_data;  int shmid;  void \*shmaddr;  struct shmid\_ds shm\_stat;  //시스템에서 식별하는 공유 메모리 번호는 20135151, 공유메모리의 크기는 struck check\_data로 공유메모리 생성  //생성 성공시 공유메모리 식별자는 shmid에 저장된다. 만약 공유 메모리가 존재한다면 생성은 하지 않고 식별자만 shmid에 저장  if((shmid = shmget((key\_t)20135151, sizeof(struct check\_data), IPC\_CREAT|0666)) == -1)  {  perror("shmid failed");  exit(1);  }  //생성된 공유 메모리를 자신의 메모리 영역으로 설정하기 위해 shmat함수 사용  //성공시 shmaddr에 공유 메모리 주소가 들어간다.  if((shmaddr = shmat(shmid, (void \*)0, 0)) == (void \*)-1)  {  perror("shmat failed");  exit(1);  }    //구조체 shared\_data에 공유 메모리 주소 shmaddr을 참조한다.(주소 대입)  shared\_data = (struct check\_data \*)shmaddr;  shared\_data->check = 1;    //무한 반복문이 돌고  while(!0){  //위에서 shared\_data->check를 1로 설정했으니 true임으로 if문 해당  if(shared\_data->check){  //공유메모리에 있던 데이터를 가져와 출력  printf("data read from shared memory: %s \n", shared\_data->data);  sleep(1);  //0으로 재설정  shared\_data->check = 0;  //공유메모리에 있던 데이터가 "quit" 라면  if(!strncmp(shared\_data->data, "quit", 4)) {  //공유메모리의 정보를 가져와 shm\_stat에 저장한다.  if(shmctl(shmid, IPC\_STAT, &shm\_stat) == -1){  perror("shmctl failed");  exit(1);  }  //공유 메모리를 사용하는 프로세스의 개수가 1 이라면 무한반복문 종료  if(shm\_stat.shm\_nattch == 1){  break;  }  }  }  sleep(1);  }  //공유 메모리를 분리해주는 작업을 해준다.  if(shmdt(shmaddr) == -1)  {  perror("shmdt failed");  exit(1);  }  //공유 메모리를 제거해준다.  if(shmctl(shmid, IPC\_RMID, 0) == -1)  {  perror("shmctl failed");  exit(1);  }  exit(0);  } |
| [lab11\_4.c]  #include <stdio.h>  #include <string.h>  #include <stdlib.h>  #include <unistd.h>  #include <sys/types.h>  #include <sys/ipc.h>  #include <sys/shm.h>  #define SIZE 1024  int main()  {  struct check\_data{  int check;  char data[SIZE];  } \*shared\_data;  int shmid;  void \*shmaddr;  char buffer[SIZE];  //시스템에서 식별하는 공유 메모리 번호는 20135151, 공유메모리의 크기는 1024로 공유메모리 생성  //생성 성공시, 공유메모리 식별자는 shmid에 저장된다. 만약 공유 메모리가 존재한다면 생성은 하지 않고 식별자만 shmid에 저장  if((shmid = shmget((key\_t)20135151, SIZE, IPC\_CREAT|0666)) == -1){  perror("shmid failed");  exit(1);  }  //생성된 공유 메모리를 자신의 메모리 영역으로 설정하기 위해 shmat함수 사용  //성공시 shmaddr에 공유 메모리 주소가 들어간다.  if((shmaddr = shmat(shmid, (void \*)0, 0)) == (void \*)-1){  perror("shmat failed");  exit(1);  }  //공유 메모리의 주소를 shared\_data에 참조시켜준다.(주소 저장)  shared\_data = (struct check\_data \*)shmaddr;  //무한 반복문 돌고  while(!0){  //공유 메모리의 shared\_data의 check가 1일때 while문 실행  while(shared\_data->check){  sleep(1);  printf("waiting....\n");  }  //데이터를 입력받고, 입력 받은 데이터를 공유 메모리 shared\_data의 data에 넣어준다.  printf("input data ==> ");  fgets(buffer, SIZE, stdin);  strncpy(shared\_data->data, buffer, SIZE);  shared\_data->check = 1;  //공유 메모리에 입력한 값이 quit일때 무한 반복문 종료  if(!strncmp(shared\_data->data, "quit", 4)){  break;  }  }  //공유 메모리를 분리해주는 작업  if(shmdt(shmaddr) == -1){  perror("shmdt failed");  exit(1);  }  exit(0);  } |