Pamięć dzielona

Poznajemy sposób tworzenia i korzystania z pamięci dzielonej.

1.Wstęp teoretyczny

•Do tworzenia segmentu pamięci wspólnej służy funkcja semget

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
int shmget (key_t key, int size, int shmflag);
```

•zwraca identyfikator pamięci lub -1 w przypadku błędu

•key - klucz identyfikujący pamięć wspólną

•size - określa rozmiar pamięci w bajtach

•semflag - jest kombinacją stałych symbolicznych określających prawa dostępu

Wartość liczbowa	Stała Symboliczna	Znaczenie
0400	SEM_R	czytanie przez właściciela
0200	SEM_A	zmienianie przez właściciela
0040	SEM_R>>3	czytanie przez grupę
0020	SEM_A>>3	zmienianie przez grupę
0004	SEM_R>>6	czytanie przez innych
0002	SEM_A>>6	zmienianie przez innych
1000	IPC_CREAT	
2000	IPC_EXCL	

•shmget tworzy lub otwiera segment pamięci wspólnej ale by z niego skorzystać trzeba go dołączyć

```
#include <sys/types.h>
#include <sys/ipc.h>
#include <sys/shm.h>
char *shmat(int shmid, char *shmaddr, int shmflag);
```

•funkcja zwraca adres początkowy segmentu pamięci wspólnej lub -1 w przypadku błędu
•adres ustalany jest według następujących zasad:

•jeżeli shmaddr = 0 to system sam wybiera adres

•jeżeli shmaddr != 0 to przekazywany adres zależy od tego czy ustalony jest znacznik SHM_RND

•jeżeli nie jest ustawiony to segment pamięci wspólnej będzie podłączony od adresu określonego przez argument shmaddr

•jeżeli jest ustawiony to zacznie się od adresu zaokrąglonego w dół o wartość stałej SHMLBA (Lower Boundary Address)
•dla większości zastosowań wystarczy dawać 0

•shmid - identyfikator pamięci dzielonej zwróconej przez shmget

•shmflag - może mieć znacznik np SHM_RDONLY

•odłączenie pamięci wspólnej dokonuje się za pomocą

```
int shmdt(char *shmaddr);
```

Funkcja ta nie usuwa segmentu pamięci wspólnej

•by usunąć segment pamięci dzielonej trzeba użyć funkcji:

```
int shmctl(int shmid, int cmd, struct shmid_ds *buf);
```

•z argumentem cmd jako IPC_RMID

2.Napisać program serwera który:

•tworzy segment pamięci dzielonej o wielkości pozwalającej na zapis zmiennej typu int (0,4p)
•tworzy dwa semafory: 1 opuszczony, 2 opuszczony (0,4p)
•w pętli wykonuje: opuszcza semafor 2, odczekuje losowy czas od 0 do 1 sekundy, wpisuje kolejną (poczynając od 1) liczbę do zmiennej dzielonej, podnosi semafor 1 (1,5p)
•program powinien usuwać pamięć dzieloną i semafory po naciśnięciu ctrl^c (0,5p)

•należy zadbać by tylko jeden program serwera tworzył semafory i pamięć (tryb exclusive) (0,2p)

3.Napisać program klienta który:

•podłącza się do segmentów pamięci i semaforów (0,5)

•w pętli (10 razy) podnosi semafor 2, opuszcza semafor 1, odczytuje wartość z pamięci dzielonej, wyświetla wartość, odczekuje losowy czas od 0 do 1 sekundy (1)

•sprawdzić działanie kilku klientów naraz (0,5)

•czy istnieje niebezpieczeństwo, że dwóch klientów dostanie tą samą liczbę?

4.Sprawozdanie

Nie zapomnijcie o sprawozdaniu