

Pamięć współdzielona

Pamięć współdzielona jest specjalnie utworzonym segmentem wirtualnej przestrzeni adresowej, do którego dostęp może mieć wiele procesów.

W zależności od praw dostępu procesy mogą odczytywać i/lub zapisywać wartości w pamięci współdzielonej, przy czym w przypadku współbieżnie działających procesów konieczne jest najczęściej synchronizowanie dostępu np. za pomocą semaforów .

Istnieje struktura opisująca obiekty pamięci współdzielonej o nazwie **shmid_ds**.

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

Funkcja SHMGET

```
PROTOTYPE: int shmget ( key_t key, size_t size, int
                        shmflags );
```

RETURNS: success : identyfikator segmentu pamięci
współdzielonej

error: -1

errno = **EACCES** (brak praw dostępu)

ENOENT (segment pamięci nie istnieje)

EIDRM (segment pamięci został usunięty)

EINVAL (nieprawidłowy rozmiar segmentu pamięci)

ENOMEM (nie ma wystarczająco dużo miejsca by
stworzyć segment pamięci współdzielonej)

EEXIST (segment pamięci współdzielonej istnieje)

PARAMETRY:

1. key - wartość klucza, który identyfikuje segment pamięci współdzielonej (ftok, IPC_PRIVATE)
2. size - wielkość segmentu pamięci współdzielonej (w bajtach)
3. shmflags - prawa dostępu do pamięci współdzielonej (IPC_CREAT, IPC_EXCL)

UWAGI:

Funkcja tworzy segment pamięci współdzielonej

Funkcja SHMCTL

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

RETURNS: success : 0

error: -1

errno = **EACCES** (nie ma praw do odczytu oraz cmd jest ustawiona na IPC_STAT)
EFAULT (adres wskazywany przez buf jest nieprawidłowy)
EIDRM (kolejka została usunięta)
EINVAL (msgqid nieprawidłowe, lub msgsz mniejsze od 0)
EPERM (komendy IPC_SET lub IPC_RMID zostały wydane podczas gdy process nie ma praw dostępu do zapisu)

PARAMETRY:

1. shmids - identyfikator pamięci współdzielonej
2. cmd - stała specyfikująca rodzaj operacji

cmd = **IPC_STAT** pozwala uzyskać informację o stanie pamięci współdzielonej
cmd = **IPC_SET** pozwala zmienić związane z kolejką ograniczenia
cmd = **IPC_RMID** pozwala usunąć pamięć współdzieloną z systemu
(shmctl(shmid, IPC_RMID, 0))

3. buf - wskaźnik na zmienną strukturalną przez którą przekazywane są parametry operacji

UWAGI:

Funkcja odpowiada funkcji msgctl. Przy próbie usunięcia segmentu odwzorowanego na przestrzeń adresową procesu system odpowiada komunikatem o błędzie.

Funkcja SHMAT

PROTOTYPE: char* shmat (int shmids, char* shmaddr, int shmflg);

RETURNS: success : adres, pod którym proces będzie „widział” przyłączony obszar pamięci.

error: -1

errno: **EACCES** (brak praw do odczytu)
EINVAL (msgqid nieprawidłowe, lub msgsz mniejsze od 0)

PARAMETRY:

1. shmids - identyfikator pamięci współdzielonej zwracany przez funkcję shmget

2. `shmaddr` – adres dla tworzonego segmentu pamięci współdzielonej lub wartość `NULL`, która powoduje, że segment dołączany jest w miejscu wybranym przez system (nie trzeba znać rozmieszczenia programu w pamięci)
3. `shmflg`
 - SHM_RDONLY** - segment jest przyłączany tylko dla odczytu
 - SHM_RND** - gdy jest ustawiony powoduje że przy wywołaniu funkcji adres `shmaddr` jest zaokrąglany w dół do granicy strony w pamięci, w przeciwnym razie pobierana jest wartość podana jako argument wejściowy.

UWAGI :

Zanim proces zacznie operować na pamięci współdzielonej musi przyłączyć jej segment. Zapisanie danych do pamięci współdzielonej odbywa się przez wykorzystanie adresu zwracanego przez funkcji `shmat`. W segmencie pamięci można umieścić każdy typ danych

oprócz wskaźników - przy dołączeniu do segmentu dwóch procesów z innymi adresami, wskaźnik wskazywać może różne obszary dla każdego z procesów.

Uwagi:

- ponieważ pamięć jest alokowana przy wywołaniu funkcji `shmat()` nie trzeba używać funkcji `malloc()` przy umieszczaniu danych w segmencie.

Funkcja SHMDT

PROTOTYPE: `char* shmdt (char* shmaddr);`

RETURNS: success : 0

error: -1

errno = **EINVAL** (nieprawidłowy adres)

PARAMETRY:

1. `shmaddr` – adres stworzonego segmentu pamięci współdzielonej

UWAGI :

Odłączenie segmentu pamięci współdzielonej. Odłączenie to powinno nastąpić po zakończeniu pracy z danym segmentem. Po wywołaniu funkcji `shmdt` licznik dołączeń do segmentu jest zmniejszany o 1.

- Po skończeniu korzystania z segmentu powinien on być usunięty

Przykład usunięcia segmentu pamięci:

```
struct shm_id_ds shm_desc;  
shmctl(shm_id, IPC_RMID, shm_desc)
```