Mechanizmy z grupy IPC

Podobnie jak łącza, **IPC** (**Inter Process Communication**) jest grupą mechanizmów komunikacji i synchronizacji procesów działających w ramach tego samego systemu operacyjnego. W skład pakiety wchodzą:

- kolejki komunikatów — umożliwiają przekazywanie określonych porcji danych,
- pamięć współdzielona — umożliwiają współdzielenie kilku procesom tego samego fragmentu wirtualnej przestrzeni adresowej,
- semafory — umożliwiają synchronizacje procesów w dostępie do współdzielonych zasobów (np. do pamięci współdzielonej).

Kolejki komunikatów

Komunikat jest pakietem informacji przesyłanym pomiędzy różnymi procesami. Każdy komunikat ma określony typ i długość. Nadawca może wysyłać komunikaty, nawet wówczas gdy żaden z potencjalnych odbiorców nie jest gotów do ich odbioru. Komunikaty są w takich przypadkach buforowane w kolejce oczekiwania na odebranie. Odbiorca może oczekiwać na pierwszy przybyły komunikat lub na pierwszy komunikat określonego typu. Komunikaty w kolejce są przechowywane nawet po zakończeniu procesu nadawcy, tak długo, aż nie zostaną odebrane lub kolejka nie zostanie zlikwidowana.

```
#include <sys / types.h>
#include <sys / ipc.h>
#include <sys / msg.h>
```

Funkcja MSGGET

```
PROTOTYPE: int msgget( key_t key, int  msgflg );
  RETURNS: success : identyfikator kolejki komunikatów
             error: -1
               errno =EACCES (kolejka skojarzona z wartością
                              key, istnieje, ale proces
                              wołający funkcję nie ma
                              wystarczających praw dostępu
                              do kolejki)
                      EEXIST (kolejka skojarzona z wartością
                              key istnieje i msgflg zawiera
                              flagi IPC_CREAT i IPC_EXCL)
                      EIDRM  (kolejka została przeznaczona
                              do usunięcia)
                      ENOENT (kolejka skojarzona z
                              wartością key nie istnieje,
```

zaś msgflg nie zawiera flagi **IPC_CREAT**)

ENOMEM (brak pamięci na utworzenie nowej kolejki)

ENOSPC (próba przekroczenia systemowego ograniczenia na ilość istniejących w systemie kolejek komunikatów)

PARAMETRY:

1. key- liczba, która identyfikuje kolejkę
2. msgflg– jest sumą bitową stałej IPC_CREAT i dziewięciu bitów określających prawa dostępu do kolejki.

UWAGI:

IPC_PRIVATE nie jest flagą tylko szczególną wartością key_t**.** Jeśli wartość ta zostanie użyta jako parametr key, to system uwzględni jedynie bity uprawnień parametru msgflg i zawsze będzie próbować utworzyć nową kolejkę.
Istnienie flag IPC_CREAT i IPC_EXCL w parametrze msgflg znaczy tyle samo, w przypadku kolejki komunikatów, co istnienie flag O_CREAT i O_EXCL w argumencie mode wywołania open , tzn. funkcja msgget nie wykona się prawidłowo jeśli msgflg będzie zawierać flagi IPC_CREAT i IPC_EXCL , zaś kolejka komunikatów skojarzona z kluczem key już będzie istnieć.

Funkcja MSGSND

PROTOTYPE: int msgsnd(int msgid, struct msgbuf *msgp, int msgs, int msgflg);

RETURNS:success : 0
error: -1

errno =**EACCES**(kolejka skojarzona z wartością key, istnieje, ale proces wołający funkcję nie ma wystarczających praw dostępu do kolejki)

EAGAIN(kolejka jest pełna, a flaga IPC_NOWAIT była ustawiona)

EFAULT(niepoprawny wskaźnik msgp)

EIDRM(kolejka została przeznaczona do usunięcia)

EINTR(otrzymano sygnał podczas oczekiwania na operację zapisu)

EINVAL(niepoprawny identyfikator kolejki, lub ujemny typ wiadomości, lub nieprawidłowy rozmiar wiadomości)

PARAMETRY:

1. `msgid` - identyfikator kolejki
2. `msgp` – wskaźnik do obszaru pamięci zawierającego treść komunikatu
3. `msgs` - rozmiar właściwej treści komunikatu
4. `msgflg` – flagi specyfikujące zachowanie się funkcji w warunkach nietypowych
   Wartość ta może być ustawiona na 0 lub **`IPC_NOWAIT`** (jeśli kolejka komunikatów jest pełna wtedy wiadomość nie jest zapisywana do kolejki, a sterowanie wraca do procesu. Gdyby flaga nie była ustawiona, proces jest wstrzymywany tak długo, aż zapis wiadomości nie będzie możliwy)

UWAGI:

Jeśli w kolejce komunikatów nie ma miejsca proces jest blokowany.
Ogólna struktura komunikatu wygląda następująco:

```
struct msgbuf{
     long mtype;        // typ komunikatu (>0)
     char mtext[1];     // treść komunikatu
}
```

1. Parametr `msgs` jest rozmiarem pola `mtext`,
2. Treść komunikatu może być dowolną strukturą
3. Pole `mtype` określa typ komunikatu, dzięki czemu możliwe jest przy odbiorze wybieranie z kolejki komunikatów określonego rodzaju. Typ komunikatu musi być wartością większą od 0.

Funkcja MSGRCV

```
PROTOTYPE: int msgrcv ( int msgid, struct msgbuf  *msgp, int
                        msgs, long msgtyp, int msgflg )
RETURNS:success : rozmiar odebranego komunikatu
           error: -1
            errno = analogicznie jak w funkcji msgsnd
```

PARAMETRY:

1. `msgid` - identyfikator kolejki
2. `msgp` – wskaźnik do obszaru pamięci zawierającego treść komunikatu
3. `msgs` - rozmiar właściwej treści komunikatu
4. `msgtyp` - typ komunikatu jaki ma być odebrany

`msgtyp >0` wybierany jest komunikat którego typ jest dokładnie taki jak `msgtyp`

`msgtyp <0` wybierany jest komunikat który ma najmniejszą wartość typu mniejszą lub równą bezwzględnej wartości z `msgtype`

msgtyp =0 typ komunikatu nie jest brany pod uwagę przy wyborze

5. msgflg – flagi specyfikujące zachowanie się funkcji w warunkach nietypowych
   Wartość ta może być ustawiona na 0 lub
   MSG_NOERROR, to komunikat przekraczający rozmiar bufora jest ucinany przy odbiorze, w przeciwnym razie odbierane są tylko komunikaty, których treść jest mniejsza od msgs
   IPC_NOWAIT, jeśli w kolejce nie ma komunikatów, i ustawiona jest w ten sposób flaga, to zwracana jest wartość –1, w przeciwnym wypadku proces jest blokowany, aż do czasu pojawienia się komunikatu

UWAGI:
Odebranie komunikatu oznacza pobranie go z kolejki i raz odebrany komunikat nie może zostać odebrany ponownie.

Funkcja MSGCTL

PROTOTYPE: int msgctl(int msgid, int cmd, struct msgid_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. msgid - identyfikator kolejki
2. cmd – stała specyfikująca rodzaj operacji

cmd = IPC_STAT pozwala uzyskać informację o stanie kolejki komunikatów
cmd = IPC_SET pozwala zmienić związane z kolejką ograniczenia

cmd = IPC_RMID pozwala usunąć kolejkę z systemu
(`msgctl(qid, IPC_RMID,0)`

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

UWAGI:
Funkcja służy do zarządzania kolejką (np. usuwania kolejki, zmiany praw dostępu itp.)