Kolejki komunikatów

Systemy Operacyjne 2

Piotr Zierhoffer

10 grudnia 2011

Mechanizmy IPC

IPC — Inter Process Communication

- kolejki komunikatów,
- pamięć współdzielona
- semafory
- polecenia bash: `ipcs, ipcrm`

Kolejki komunikatów

- identyfikowane kluczem
- wyszczególnione typy i długość wiadomości
- dostęp niekoniecznie sekwencyjny
- wiadomości utrzymywane nawet po zakończeniu procesu nadawcy
- nadawca może nadawać nawet, gdy nie ma odbiorców

Stworzenie kolejki

- Funkcja:
  `int msgget(key_t key, int msgflg)`
- Parametry:
  - `key` — klucz identyfikujący kolejkę (“nazwa”)
  - `msgflg` — flagi
- Wartość zwracana:
  - -1 w wypadku błędu
  - identyfikator kolejki powiązany z kluczem `key`

Stworzenie kolejki, cd.

Opis parametrów:

- `key` — klucz identyfikujący kolejkę ("nazwa")
  - `IPC_PRIVATE` — stwórz nową kolejkę o dowolnym identyfikatorze, ignoruj `msgflg` oprócz praw dostępu
  - zwyczajowo wartość heksadecymalna
- `msgflg` — flagi, sumowane bitowo
  - prawa dostępu — np. 0600
  - `IPC_CREAT` — tworzenie kolejki
  - `IPC_EXCL` — "jeżeli kolejka istnieje i użyto `IPC_CREAT`, zwróć błąd"

Wysłanie komunikatu

- Funkcja:
  `int msgsnd(int msqid, struct msgbuf* msgp, int msgsz, int msgflg)`
- Parametry:
  - `msqid` — identyfikator kolejki zwrócony przez `msgget`
  - `msgp` — struktura zawierająca komunikat
  - `msgsz` — rozmiar komunikatu (bez informacji o jego typie)
  - `msgflg` — flagi
- Wartość zwracana:
  - 0 lub -1 w wypadku błędu

Wysyłanie komunikatu, cd.

- `msgflg` — flagi specyfikujące zachowanie funkcji
  - `IPC_NOWAIT` — jeżeli brakuje miejsca w kolejce, nie czekaj, zwróć błąd
- `msgp` — struktura zawierająca komunikat

```
struct msgbuf {
        long mtype; //typ wiadomości, większy od 0
        char mtext[1]; //dane komunikatu - dowolna tablica,
                                struktura lub lista zmiennych
}
```

- `msgsz` — rozmiar struktury `mtext`

Odbiór komunikatu

- Funkcja:
  `int msgrcv(int msqid, struct msgbuf* msgp, int msgsz, long msgtyp, int msgflg)`
- Parametry:
  - `msqid` — identyfikator kolejki zwrócony przez `msgget`
  - `msgp` — struktura zawierająca komunikat
  - `msgsz` — rozmiar komunikatu (bez informacji o jego typie)
  - msgtyp — typ komunikatu do odbioru
  - `msgflg` — flagi
- Wartość zwracana:
  - -1 w wypadku błędu
  - długość struktury `mtext`

Odbiór komunikatu, cd.

- `msgtyp` — typ komunikatu do odbioru
  - 0 — odbierz pierwszy komunikat z kolejki
  - n>0 — odbierz komunikat o `mtype` = n
  - n<0 — odbierz komunikat o **najniższym** `mtype` < |n| (priorytety)

- `msgflg` — flagi specyfikujące zachowanie funkcji
  - `IPC_NOWAIT` — jeżeli w kolejce nie ma komunikatu, nie czekaj i zwróć błąd
  - `MSG_EXCEPT` — jeżeli `msgtyp` > 0, odbierz komunikat o `mtype` != `msgtyp`
  - `MSG_NOERROR` — jeżeli komunikat jest dłuższy niż `msgsz`, przytnij go do odpowiedniej wielkości

Zarządzanie kolejką

- Funkcja:
  `int msgctl(int msqid, int cmd, struct msqid_ds *buf)`
- Parametry:
  - `msqid` — identyfikator kolejki zwrócony przez `msgget`
  - `cmd` — operacja do wykonania
    - IPC_RMID — usunięcie kolejki (i wznowienie wszystkich procesów oczekujących)
    - IPC_SET — zmiana parametrów kolejki: właściciela, praw, pojemności, etc. na podstawie `buf`
    - IPC_STAT — pobranie parametrów kolejki do `buf`
  - `buf` — struktura na parametry operacji
- Wartość zwracana:
  - 0 lub -1 w wypadku błędu

Przykład

```
struct msgbuf {
    long type;
    char text[1024];
};
struct msgbuf my_msg_send, my_msg_recv;

int mid = msgget(0x123, 0600 | IPC_CREAT);
strcpy(my_msg_send.text, "Text");
my_msg_send.type = 5;

msgsnd(mid, &my_msg_send, strlen(my_msg_send.text)+1, 0);
msgrcv(mid, &my_msg_recv, 1024, 5, 0);

printf("%s\n", my_msg_recv.text);
msgctl(mid, IPC_RMID, NULL);
```