

Lokalna komunikacja procesów

Zbigniew Suski

Lokalna komunikacja procesów

1

Semafor - narzędzie synchronizacji

Semafor jest zmienną całkowitą z dwoma wyróżnionymi operacjami:

- opuszczenie semafora (P, *signal*)
- podniesienie semafora (V, *wait*)

Opuszczenie semafora P(S):

Jeżeli $S > 0$, to $S--$, w przeciwnym przypadku wstrzymaj działanie procesu wykonującego tą operację.

Podniesienie semafora V(S):

Jeżeli są procesy wstrzymane w wyniku wykonania operacji opuszczania semafora S, to wznów jeden z nich, w przeciwnym przypadku $S++$;

Niepodzielność operacji

Semafor ogólny

Semafor binarny

Zbigniew Suski

Lokalna komunikacja procesów

2

UNIX - semafony

```
int semget ( key_t nr_zbioru,
             int   liczba_semaforów_w_zbiorze,
             int   flagi_dostępu_do_semafora)

int semctl(  int   identyfikator_zbioru_semaforów,
             int   nr_semafora_w_zbiorze,
             int   kod_operacji,
             union semun      argument)

union semun {
    int          val;      /* tylko dla SETVAL */
    struct semid_ds *buf; /* dla IPC_STAT, IPC_SET */
    ushort       *array; /* dla GETVAL, SETVAL */
};
```

Zbigniew Suski

Lokalna komunikacja procesów

3

UNIX - semafony cd.

```
int semop(
    int          identyfikator_zbioru_semaforów,
    struct sembuf *tablica_operacji_na_semaforze,
    int          ilość_elementów_tablicy_operacji)

struct sembuf {
    ushort       sem_num; /* nr_semafora */
    short        sem_op;  /* operacja na semaforze */
    short        sem_flg; /* znaczniki operacji */
};
```

Zbigniew Suski

Lokalna komunikacja procesów

4

UNIX - semafony - flagi

dla funkcji *semop*:

IPC_NOWAIT wyłączenie czekania na zakończenie operacji

dla funkcji *semget*:

IPC_CREAT utworzenie zbioru semaforów,
brak sygnalizacji błędu, gdy zbiór już istnieje
IPC_EXECL sygnalizacja błędu, gdy zbiór już istnieje
(łącznie z IPC_CREAT)

dla funkcji *semctl*:

IPC_RMID usunięcie zbioru semaforów
GETVAL pobranie wartości semafora
SETVAL - nadanie wartości (np. początkowej)

Zbigniew Suski

Lokalna komunikacja procesów

5

UNIX – pamięć dzielona

```
int shmget(
    key_t klucz,          /* klucz kanału komunikacyjnego */
    int  rozmiar,        /* rozmiar segmentu w bajtach */
    int  flagi_dostępu);

char *shmat(
    int  identyfikator_segментu_pamięci,
    char *adres,          /* jeżeli 0 to jądro określa adres */
    int  flagi);

int shmdt( char * adres_segментu);

int shmctl(
    int  identyfikator_segментu_pamięci,
    int  polecenie,       /* należy wstawić IPC_RMID */
    struct shmid_ds *buf); /* zwykle 0 */
```

Zbigniew Suski

Lokalna komunikacja procesów

6

UNIX - semafony i pamięć dzielona

Przykład:

plik [Komunikacja lokalna-przyklady](#)

Zbigniew Suski

Lokalna komunikacja procesów

7

UNIX - kolejki komunikatów

```
int msgget(
    key_t nr_kanału,
    int  flagi_dostępu)

int msgsnd(
    int  identyfikator_kolejki,
    struct msgbuf *komunikat,
    int  długość_komunikatu,
    int  flagi)

struct msgbuf {
    long mtype; /* typ komunikatu (>0) - obowiązkowy */
    char mtext[1]; /* dane komunikatu - opcjonalne */
};
```

Zbigniew Suski

Lokalna komunikacja procesów

8

UNIX - kolejki komunikatów cd

```
int msgrcv(  
    int    identyfikator_kolejki,  
    struct msgbuf    *komunikat,  
    int    długość_komunikatu,  
    int    pożądany_typ_komunikatu,  
    int    flagi)
```

```
int msgctl(  
    int    identyfikator_kolejki,  
    int    polecenie,  
    struct msqid_ds    *opis_kolejki)
```

Zbigniew Suski

Lokalna komunikacja procesów

9

UNIX - kolejki komunikatów cd

Niektóre flagi dla funkcji *msgcnd*, *msgrcv*:

IPC_NOWAIT - natychmiastowy powrót z funkcji w przypadku braku miejsca, lub braku komunikatu w buforze

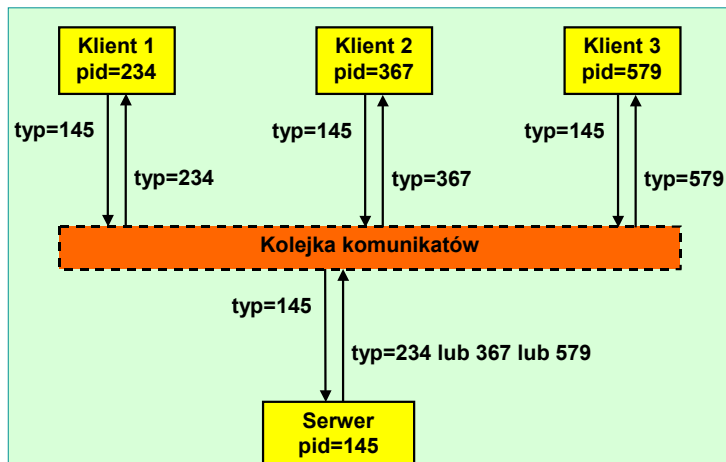
MSG_NOERROR - jeżeli komunikat, który przyszedł jest większy niż założono, to zostanie obcięty i błąd nie będzie sygnalizowany

Zbigniew Suski

Lokalna komunikacja procesów

10

Klient-serwer - kolejki komunikatów



Zbigniew Suski

Lokalna komunikacja procesów

11

UNIX - kolejki komunikatów

Przykład:

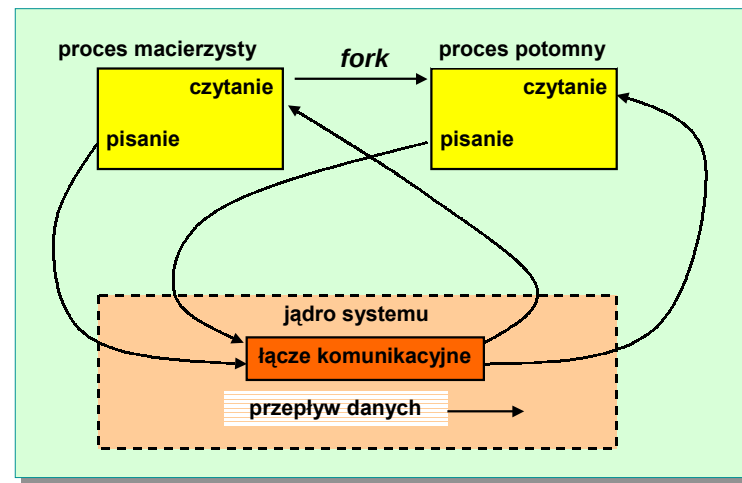
plik [Komunikacja lokalna-przykłady](#)

Zbigniew Suski

Lokalna komunikacja procesów

12

UNIX - łącza nienazwane (pipes)

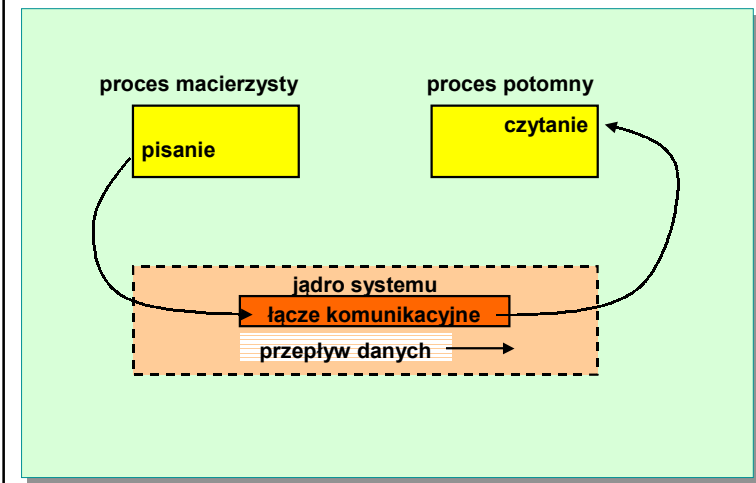


Zbigniew Suski

Lokalna komunikacja procesów

13

UNIX - łącza nienazwane cd



Zbigniew Suski

Lokalna komunikacja procesów

14

UNIX - łącza nienazwane cd

```
int deskryptor_pliku[2];
int pipe (    int    *deskryptor_pliku)
```

deskryptor_pliku[0] - otwarty do czytania

deskryptor_pliku[1] - otwarty do pisania.

Zbigniew Suski

Lokalna komunikacja procesów

15

UNIX - łącza nienazwane cd

```
int main()
{
    if (pipe(deskryptor) < 0) return(1);
    if ((f=fork())==0) { /* POTOMNY: konsument */
        close(deskryptor[1]);
        while( read(deskryptor[0], buf, LEN)) {
            uzyj( buf);
        }
        close(deskryptor[0]);
        return(0);
    } else if (f == -1) {
        printf("\nBłąd powołania procesu \n");
        return(1);
    } else { /* MACIERZYSTY: producent */
        close(deskryptor[0]);
        while !koniec_produkcji() {
            produkuj(buf);
            write(deskryptor[1],buf,LEN);
        }
        close(deskryptor[1]);
        return(0);
    }
}
```

Zbigniew Suski

Lokalna komunikacja procesów

16

UNIX - komunikacja dwukierunkowa

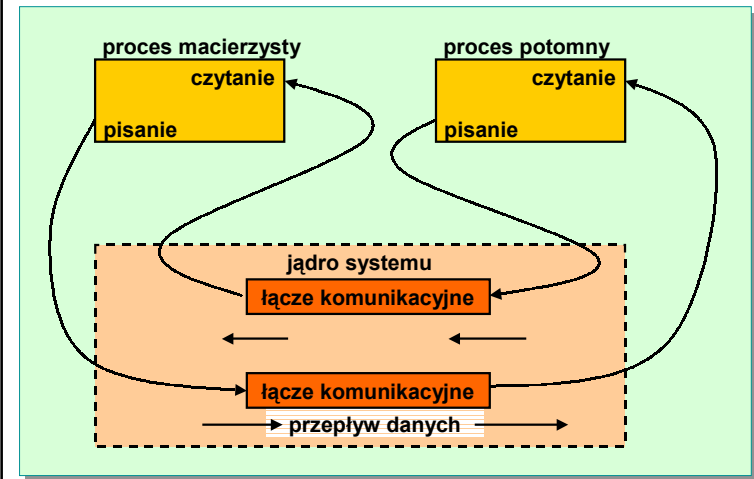
- ❑ Tworzenie łącza 1 i łącza 2.
- ❑ Wywołanie funkcji *fork*.
- ❑ Przodek zamyka łącze 1 do czytania.
- ❑ Przodek zamyka łącze 2 do pisania.
- ❑ Potomek zamyka łącze 1 do pisania.
- ❑ Potomek zamyka łącze 2 do czytania.

Zbigniew Suski

Lokalna komunikacja procesów

17

UNIX - komunikacja dwukierunkowa



Zbigniew Suski

Lokalna komunikacja procesów

18

UNIX - łącza nienazwane cd

Przykład:

plik [Komunikacja lokalna-przykłady](#)

Zbigniew Suski

Lokalna komunikacja procesów

19

UNIX – łącza nazwane (kolejki FIFO)

```
int mknod (char *ścieżka, int tryb, int urządzenie)
tryb = S_IFIFO | 0666      (przykładowo)
```

producent

```
int fh;
char buf[LEN];
if ((fh = open("/tmp/fifo1", O_WRONLY) < 0)) exit(1);
while (!koniec_produkcji()) {
    produkuj(buf);
    write(fh, buf, LEN);
}
close(fh);
```

Zbigniew Suski

Lokalna komunikacja procesów

20

UNIX – łącza nazwane (kolejki FIFO)

konsument

```
int fh;  
char buf[LEN];  
if ((fh = open("\\tmp\\fifo1", O_RDONLY) < 0))  
    exit(1);  
while (read(fh,buf,LEN)) {  
    użyj(buf);  
}  
close(fh);
```

UNIX – łącza nazwane (kolejki FIFO)

Przykład:

plik [Komunikacja lokalna-przyklady](#)