1 IPC

IPC (InterProcess Communication) to udostępniane przez jądro systemu operacyjnego mechanizmy służące komunikacji oraz współdzieleniu zasobów i informacji pomiędzy procesami. IPC Systemu V udostępnia następujące mechanizmy:

1. semafory,
2. kolejki komunikatów,
3. pamięć dzieloną.

2 Implementacja w systemie Linux 2.4

2.1 Synchronizacja poziomu jądra

2.1.1 Operacje atomowe

Operacje atomowe (niepodzielne) to udostępniane przez jądro proste, niepodzielne funkcje arytmetyczne i arymetyczno-logiczne, takie jak dodawanie, odejmowanie, ustawianie i testowanie wartości zmiennych i bitów. Funkcje te realizowane są za pomocą niepodzielnych instrukcji asemblera (instrukcje pobierające dane z pamięci, instrukcje modyfikujące dane w pamięci – *inc* oraz *dec*, instrukcje poprzedzone bajtem blokady *lock (0xf0)*).

Operacje te zdefiniowane są w pliku *include/asm/atomic.h*.

2.1.2 Wyłączanie przerwań

Wyłączanie przerwań jest kluczowym mechanizmem gwarantującym wykonanie sekwencji instrukcji jądra jako sekcji krytycznej. Mechanizm ten pozwala na nieprzerwane działanie ścierzek wykonania nawet wtedy, gdy urządzenia zewnętrzne zgłąszają sygnały IRQ. Pozwala to na efektywną ochronę struktur danych wykorzystywanych przez jądro oraz procedury obsługi przerwań. Wyłączanie przerwań odbywa sie za pomocą makrodefinicji cli() a ich włączanie za pomocą makrodefinicji sti(), dodatkowo udostępniane są makrodefinicje save_flags i restore_flags które służą do zapisywania i przywracania stanu rejestru eflags. Wyczyszczenie flagi IF rejestru eflags powoduje wyłączenie przerwań a jej ustawienie - ponowne ich włączenie. Makra te są użyteczne w przypadku gdy dana sekcja krytyczna może być wywołana z innego przerwania (przypadek przerwań zagnieżdżonych).

2.1.3 Semafory jądra

Semafory to obiekty przyjmujące wartości całkowite, na których można wywoływać dwie niepodzielne operacje P()(up) i V() (down) Operacja P() zmniejsza o 1 wartość semafora i powoduje wstrzymanie procesu, jeżeli wartość jest mniejsza od 0. Operacja V() zwiększa wartość o 1; jeżeli wartość jest większa-równa od zera budzony jest proces oczekujący na danym semaforze.

Semafory jądra to stuktury (*struct semaphore*) posiadające następujące pola:

- count – licznik semafora (wartość całkowita) - wartość mniejsza lub równa zero oznacza,ze zasób jest zajęty. Operacje na tym polu wykonywane są za pomocą instrukcji atomowych,
- wait – wskaźnik do kolejki przechowującej uśpione procesy oczekujące na zasób,
- waking – pole pomocnicze służące wybudzaniu procesów.

Wybudzanie procesów oczekujących (w funkcji *down()*) polega na ustawieniu pola *waking* na 1 i wykonaniu przez każdy z oczekujących procesów testu *waking > 0*. Jeżeli wartośc testu jest prawdą pole waking jest zmniejszane o 1. Powoduje to wybudzenie tylko 1 procesu z kolejki procesów oczekujących. Regiony krytyczne funkcji *down* i *up* chronione są za pomocą globanej blokady pętlowej *semaphore_wake_lock*.

2.2 Semafory IPC

Semafory (poziomu użytkownika) implementowane są przy pomocy semaforów prostych (jądra). Każdy semafor IPC (a właściwie jego adres) przechowywany jest w tablicy (semary). Tablica ta może posiadać także wartości IP_UNUSED / IPC_NOID oznaczającej fakt nieużywania danego pola tablicy. Semafor IPC składa sie z deskryptora semafora (semid_ds - zawiera listę operacji oczekujących – FIFO, operacji do cofnięcia i liczbę semaforów (sem) w tablicy) oraz struktur *sem* opisujących semafory proste (licznik sem. prostego oraz PID ostatnigo procesu korzystającego z semafora).

2.3 Kolejki komunikatów IPC

Kolejka komunikatów jest mechanizmem pozwalającym na przekazywnie niewielkich porcji komunikatów pomiędzy procesami. Każdy komunikat posiada, poza danymi, także typ oraz priorytet. W systemie Linux 2.4 kolejka komunikatów jest synchronizowana za pomocą *globalnego semafora kolejki komunikatów*

(funkcje: sys_msgget, sys_smgctl – *IPC_SET, IPC_RMID*), oraz globalnej blokady wirującej kolejki (funkcje: sys_msgget,sys_smgctl – *PC_STAT, IPC_SET, IPC_RMID* , sys_msgsnd, sys_msgrcv, newque, freeque).

2.4 Pamięć dzielona IPC

Pamięć dzielona jest najszybszym sposobem komunikacji pomiędzy procesami. Jest to fragment pamięci fizycznej współdzielony przez wiele procesów. Jego obsługa po przyłączeniu do mapy pamięci procesu jest analogiczna do obsługi dowolnej innej pozycji w pamięci.

Tak jak semafory IPC segmenty pamięci przechowywane są w globalnej tablicy *shm_segs*. Tablica ta zawiera adresy deskryptorów współdzielonych segmentów pamięci IPC. Każdy z deskryptorów (struktura *shmid_kernel*) zawiera między innymi:

- wskaźnik do tablicy zawierającej pozycje tablicy stron bloków stronnicowanych (adresy stron tworzących dany segment pamięci IPC).
- czasy ostatniej operacji dołączenia i odłączenia
- rozmiar pamięci współdzielonej

Dostęp do powyższych struktur chroniony jest poprzez globalny semafor jądra – *shm_ids.sem*(pobranie segmentu pamięci dzielonej), oraz blokadę wirującą *shm_ids.ary*.

3 Konfiguracja testowa

3.1 klaster OpenSSI

Trzy maszyny każda z nich w następującej konfiguracji:

Procesor	2 x Intel(R) Pentium(R) 4 CPU 2.80GHz (HT)
Kompilator	gcc version 3.3.5 (Debian 1:3.3.5-12)
System operacyjny	Linux lab4-cs1 2.4.22-1.2199.nptl-ssi-686-smp

3.2 komputer porównawczy

Procesor	AMD Athlon(tm) 64 Processor 3200+
Kompilator	gcc version 3.4.3 20041125 (Gentoo Linux 3.4.3-r1, ssp-3.4.3-0, pie-8.7.7)
System operacyjny	Linux wieszak 2.6.10-gentoo-r5

4 Propozycje testów

4.1 Semafory

Badanie wydajności synchronizacji procesów na wielu semaforach – problem "Problem jedzących filozofów".

Szczegóły testu *N* filozofów, każdy z 1 widelcem,umieszczonych w kolejce obsługiwanych jest przez 33 procesy.
Stany każdego filozofa to:

1. *początek cyklu* (pobranie zadania z kolejki)
2. myślenie (zaśnięcie na losowy okres czasu)
3. kompletowanie dwóch widelców (synchronizacja na dwóch semaforach)
4. jedzenie (zaśnięcie na losowy okres czasu)
5. odłożenie widelców (zwolnienie semaforów)

Dodatkowo, w celu zwiększenia ilości operacji na semaforach każdy filozof podczas jednej iteracji zmienia kolumnę wartości w pomocniczej macierzy semaforów, która jest współdzielona pomiędzy wszystkie procesy.

Konfiguracje testów

1. 3 wezly; procesy porozmieszczane równomiernie na wszystkich trzech węzłach,
2. 2 wezly; zasoby IPC stworzone jednym węźle, procesy uruchomione na drugim,
3. komputer testowy; 1 procesor.

Tablica 1: Wyniki testów

Konfiguracja	min [s]	max [s]
1	3243	4307
2	4755	5095
3	128	130

4.2 Pamięć dzielona

2 grupy procesów: 1 grupa (1 proces) odczytuje dane z pliku (/dev/urandom) i zapisuje je do segmentu pamięci. Druga grupa (pozostałe 32) odczytuje dane z pamięci. Druga grupa poza odczytywaniem danych modyfikuje także licznik kolejki umieszczony w tym samym segmencie pamięci współdzielonej.

Konfiguracja programu: rozmiar kolejki umieszczonej w segm. pamięci dzielonej: 4096 Bajty, 1000 iteracji zapisu.

Konfiguracje testów

1. 3 wezly; czytelnicy rozmieszczeni rownomiernie na 3 węzłach , pisarz i zasoby IPC umieszczone na 1 nym samym węźle,
2. 3 wezly; czytelnicy umieszczeni na 1 węźle, zasoby IPC na 2 węźle, pisarz na 3 węźle,
3. 1 węzeł; wszystkie procesy na tym samym węźle,
4. 2 węzły – czytelnicy rozmieszczeni rownomiernie
5. 2 węzły – czytelnicy rozmieszczeni rownomiernie, brak modyfikacji segmentu pamięci (w tym przypadku zmniejszenia indeksu kolejki),
6. komputer testowy; 1 procesor.

Tablica 2: Wyniki testów

Konfiguracja	czas avg [s]
1	600
2	257
3	250
4	372
5	319
6	40

5 Implementacja IPC w systemie OpenSSI – wnioski i obserwacje

Operacje na obiektach IPC implementowane są przy pomocy wywołań odpowiednich metod RPC. Serwer obsługujący wywołania RPC znajduje się na komputerze który utworzył dany zasób IPC. W przypadku pamięci dzielonej, w celu zwiększenia szybkości działania, następuje migracja segmentów pamięci na węzeł aktualnie użytkujący dany zasób. powoduje to znaczne opóźnienia w przypadku gdy procesy najczęściej używające współdzielonego segmentu pamięci znajdują sie na różnych węzłach.

W przypadku, gdy procesy najczęściej używające segmentu dokonują TYLKO odczytów wzrost wydajności wynosi w granicach 15% w stosunku do użycia typu odczyt-zapis.