Procesy, wątki i wielozadaniowość

Definicja Procesu

- **Proces** – jedno z najbardziej podstawowych pojęć w informatyce, definiowane jako egzemplarz wykonywanego programu. W celu wykonania programu system operacyjny przydziela procesowi zasoby (pamięć, czas procesora i inne – szczegółowa lista zasobów znajduje się dalej), ale także może być konieczne współbieżne wykonywanie pewnych fragmentów programu.
- W systemie operacyjnym każdy proces posiada proces nadrzędny, z kolei każdy proces może, poprzez wywołanie funkcji systemu operacyjnego, utworzyć swoje procesy potomne; w ten sposób tworzy się swego rodzaju drzewo procesów.

Stany procesu

- Proces wykonywany pod nadzorem systemu operacyjnego przebywa w jednym z następujących stanów:

  a) nowy – w tym stanie proces znajduje się zaraz po utworzeniu;

  b) wykonywanie – proces ma przydzielony procesor, który wykonuje jego instrukcje;

  c) oczekiwanie – proces oczekuje na zdarzenie (np. na zakończenie operacji wejścia-wyjścia);

  d) gotowość – proces oczekuje na przydział procesora w kolejce procesów gotowych do wykonywania;

  e) zakończony – proces zakończył działanie, lecz wciąż pozostaje w systemie (np. nie przekazał wyników).

Stany procesu i przejścia między nimi

- "Najwazniejszym" stanem jest **TASK_RUNNING**. Proces znajdujacy sie w tym stanie jest gotowy do wykonania, i to ze (byc moze) sie akurat nie wykonuje spowodowane jest faktem, ze procesor (procesory) wykonuje akurat inne zadanie (zadania).
- **TASK_UNINTERRUPTIBLE**, kiedy to proces w tym stanie nie odpowiada na zadne sygnaly. Proces "wpada" w taki stan z przyczyn od niego niezaleznych (bezposrednio niezaleznych). I np. tak kiedy proces jest tworzony, gdy wisi na semaforze zwiazanym z wewnetrzna organizacja jadra (dostep jadra do krytycznych fragmentow pamieci (zmienne))
- Proces konczacy swoje zycie przechodzi w stan **TASK_ZOMBIE** i czeka (tak naprawde to czekaja "pozostalosci" po nim) az zostana usuniete ze struktur utrzymywanych przez jadro.
- **TASK_STOPPED** – jak wiemy proces moze zostac zatrzymany przez wlasciciela grupy do ktorej nalezy. Gdy tak sie stanie, to pole state przyjmuje wlasnie wartosc TASK_STOPPED i proces pozostaje w stanie zatrzymania do momentu otrzymania sygnalu kontynuacji, badz zakonczenie (kill).

Wykonywanie procesów

- Dany proces rozpoczyna wykonywanie w momencie przełączenia przez Jądro systemu operacyjnego przestrzeni adresowej na przestrzeń adresową danego procesu oraz takie zaprogramowanie procesora, by wykonywał kod procesu. Wykonujący się proces może żądać pewnych zasobów, np. większej ilości pamięci. Zlecenia takie są na bieżąco realizowane przez system operacyjny.
- Wykonanie procesu musi przebiegać sekwencyjnie

Kończenie procesów

- Proces wykonuje ostatnią instrukcję – zwraca do systemu operacyjnego kod zakończenia. Jeśli proces zakończył się poprawnie zwraca wartość 0, w przeciwnym wypadku zwraca wartość kodu błędu.
- W momencie zwrotu do systemu operacyjnego kodu zakończenia, system operacyjny ustawia stan procesu na przeznaczony do zniszczenia i rozpoczyna zwalnianie wszystkich zasobów, które w czasie działania procesu zostały temu procesowi przydzielone.
- System operacyjny po kolei kończy wszystkie procesy potomne w stosunku do procesu macierzystego.
- System operacyjny zwalnia przestrzeń adresową procesu. Jest to dosłowna śmierć procesu.
- System operacyjny usuwa proces z kolejki procesów gotowych do uruchomienia i szereguje zadania. Jest to ostatnia czynność wykonywana na rzecz procesu.
- Procesor zostaje przydzielony innemu procesowi.

Zasoby procesu

- W skład procesu wchodzi:
- kod programu,
- licznik rozkazów,
- stos,
- sekcja danych
- Każdemu procesowi przydzielone zostają zasoby, takie jak:
- procesor,
- pamięć,
- dostęp do urządzeń wejścia-wyjścia,
- pliki

Wątek

Część programu wykonywana współbieżnie w obrębie jednego procesu; w jednym procesie może istnieć wiele wątków.

- Różnica między zwykłym procesem a wątkiem polega na współdzieleniu przez wszystkie wątki działające w danym procesie przestrzeni adresowej oraz wszystkich innych struktur systemowych (np. listy otwartych plików, gniazd, itp.) – z kolei procesy posiadają niezależne zasoby.

Wielowątkowość

- Cecha systemu operacyjnego, dzięki której w ramach jednego procesu może wykonywać kilka wątków lub jednostek wykonawczych. Nowe wątki to kolejne ciągi instrukcji wykonywane oddzielnie. Wszystkie wątki tego samego procesu współdzielą kod programu i dane.

Współbieżność

- Metoda umożliwiająca równoczesne korzystanie z bazy danych przez kilku użytkowników bez ryzyka zniszczenia swoich danych. Jednoczesne wykonywanie procesów w systemie wieloprogramowym.

Wady i zalety wątków

- Wątki wymagają mniej zasobów do działania i też mniejszy jest czas ich tworzenia.
- Dzięki współdzieleniu przestrzeni adresowej (pamięci) wątki jednego zadania mogą się między sobą komunikować w bardzo łatwy sposób, niewymagający pomocy ze strony systemu operacyjnego. Przekazanie dowolnie dużej ilości danych wymaga przesłania jedynie wskaźnika, zaś odczyt (a niekiedy zapis) danych o rozmiarze nie większym od słowa maszynowego nie wymaga synchronizacji (procesor gwarantuje atomowość takiej operacji).

Bezpieczeństwo wątków

- W programowaniu współbieżnym jest pojęciem stosowanym w kontekście programów wielowątkowych. Fragmenty kodu są "wątkowo-bezpieczne", jeżeli funkcje wywoływane jednocześnie przez wiele wątków wykonują się prawidłowo. W szczególności musi być spełniony następujący warunek: współbieżne wątki mają dostęp do tych samych współdzielonych danych oraz dostęp do fragmentu współdzielonych danych jest możliwy dla dokładnie jednego wątku w danym czasie.

Wielozadaniowość

- Zdolność do wykonywania przez komputer więcej niż jednego zadania naraz. Wielozadaniowy system operacyjny potrafi wykorzystywać zasoby i dzielić je pomiędzy uruchomionymi procesami. W rzeczywistości procesor nie jest w stanie wykonywać dwóch operacji naraz, ale dzieli czas swojej pracy na krótkie odcinki dla każdego procesu, tak, aby nie było to zauważalne dla użytkownika, po czym wykonuje zadane mu zadania..

Planista

- **Algorytm szeregowania** to algorytm rozwiązujący jedno z najważniejszych zagadnień informatyki – jak rozdzielić czas procesora i dostęp do innych zasobów pomiędzy zadania, które w praktyce zwykle o te zasoby konkurują.

Rodzaje planistów

- **1)Planista długoterminowy lub planista zadań –**
- wybiera procesy, które powinny być sprowadzone do pamięci z kolejki procesów gotowych, jest wołany rzadko(min,sek)dlatego może być wolniejszy

- **2)Planista krótkoterminowy lub planista przydziału procesora**

- wybiera proces następny do wykonania z kolejki procesów gotowych i przydziela mu procesor jest wołany bardzo często (milisekundy) dlatego musi być bardzo szybki

- **3)Planista średnioterminowe – w celu uzyskania lepszego doboru** procesów.