Procesy, wątki i wielozadaniowość

Proces

Jedno z najbardziej podstawowych pojęć w informatyce, definiowane jako egzemplarz wykonywanego programu, jednak każdy nowo powstały proces otrzymuje unikalny numer, który go jednoznacznie identyfikuje, tzw. numer PID (ang. process IDentifier).

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.

Aby to zrealizować program może zażądać utworzenia określonej liczby wątków, wykonujących wskazane części programu - o ich współbieżne wykonanie dba system operacyjny (albo sam program, wówczas mówi się o zielonych wątkach).

WAŻNE!

Wątki współdzielą prawie wszystkie zasoby zarezerwowane dla danego procesu.

Wyjątkiem jest czas procesora, który jest przydzielany indywidualnie każdemu wątkowi.

Za zarządzanie procesami odpowiada jądro systemu operacyjnego, sposób ich obsługi jest różny dla różnych systemów operacyjnych.
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.

Każdy proces otrzymuje od systemu operacyjnego odrębne zasoby, w tym odrębną przestrzeń adresową, listę otwartych plików, urządzeń itp.

Wątek (ang. thread)

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

PROCES vs WĄTEK

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.

Ta cecha ma dwie ważne konsekwencje:

1. Wątki wymagają mniej zasobów do działania i też mniejszy jest czas ich tworzenia.

2. Dzięki współdzieleniu przestrzeni adresowej (pamięci) wątki jednego zadania mogą się między sobą komunikować w bardzo łatwy sposób, niewymagający pomocy ze strony systemu operacyjnego.

Wielozadaniowość

Cecha systemu operacyjnego umożliwiająca mu równoczesne wykonywanie więcej niż jednego procesu. Zwykle za poprawną realizację wielozadaniowości odpowiedzialne jest jądro systemu operacyjnego.

Wielozadaniowość zapewniona jest między innymi przez planistę, czyli część systemu operacyjnego realizującą algorytm szeregowania zadań w kolejce do przyznania czasu procesora.

UWAGA!!!

Równoczesność jest pozorna, gdy system ma dostępnych mniej procesorów niż zadań do wykonania. Wówczas dla uzyskania wrażenia wykonywania wielu zadań jednocześnie, konieczne staje się dzielenie czasu.

PROCES TWORZENIA PROCESU:

* Użytkownik za pomocą powłoki zleca uruchomienie programu

* System operacyjny tworzy przestrzeń adresową dla procesu oraz strukturę opisującą nowy proces w następujący sposób:

  - wypełnia strukturę opisującą proces,

  - kopiuje do przestrzeni adresowej procesu dane i kod, zawarte w pliku wykonywalnym,

  - ustawia stan procesu na działający,

  - dołącza nowy proces do kolejki procesów oczekujących na procesor (ustala jego priorytet),

  - zwraca sterowanie do powłoki użytkownika.

Stany procesów, w których mogą się znajdować:

* aktualnie wykonywany przez procesor,
* czekający na udostępnienie przez system operacyjny zasobów,
* Uśpiony
* przeznaczony do zniszczenia,
* proces zombie,
* właśnie tworzony itd.

Działanie wielu wątków w obrębie danego procesu wymaga ich synchronizacji – trzeba zapewnić, że w trakcie gdy np. jeden wątek czyta wartość zmiennej X, inny wątek nie zmieni w tym samym czasie jego wartości.

Przykład błedów do jakich może doprowadzić brak synchronizacji:

Stan SYSTEMU	**Zadanie realizowane przez watek1**	**Zadanie realizowane przez wątek2**
Stan początkowy: Stan_konta = 500zł Bez zmian: Stan_konta = 500zł Zapisano nową wartość stanu konta: Stan_konta = 600zł Zapisano nową wartość stanu konta: Stan_konta = 400zł	1.Odczyt stanu konta do swojej lokalnej zmiennej nie widoczenej przez inne wątki: ZMIENNA_LOKALNA1:=Stan_Konta (500zł) 2.Interakcja z użytkownikiem 3. Zmiejszenie wartości stanu konta zapisanej w zmiennej lokalnej o 100zł (ZMIENNA_LOKALNA1 – 100zł = 400zł) 4. Zapis nowej wartości Stanu_konta do globalnej zmiennej Stan_konta:=ZMIENNA_LOKALNA1 (400zł)	1. Odczyt stanu konta do swojej lokalnej zmiennej nie widoczenej przez inne wątki: ZMIENNA_LOKALNA2:=Stan_Konta (500zł) 2. Zwiększenie lokalnej zmiennej ZMIENNA_LOKALNA2 o 100zł. ZMIENNA_LOKALNA = 600zł; 5. Zapisanie nowej wartości stanu konta ze zmiennej lokalnej ZMIENNA_LOKALNA2: Stan_konta:=ZMIENNA_LOKALNA2 (600zł)
Stan końcowy: Stan_konta = 400zł POWINNO BYĆ 500zł	--	--

Jak widać z powyższej tabelki dwa wątki działały w tym samym czasie wykonując pewne operacje odczytu i zapisu na współdzielonej zmiennej Stan_konta. Ponieważ brak było tu mechanizmów zabezpieczających – synchronizujących działanie dwóch wątków na jednej, wspólnej zmiennej doszło do ogromnego przekłamania.

Zmienna Stan_konta powinna mieć wartość końcową 500zł. Na początku wartość Stan_konta wynosiła 500zł, a następnie wątek1 dokonał operacji WYPŁATY pieniędzy zdejmując z konta 100zł. W tym samym czasie wątek2 dokonał WPŁATY gotówki, zwiększając saldo konta o 100zł. Jak łatwo obliczyć wartość końcowa Stanu_konta powinna wynosić 500zł (startowo było 500zł, wypłacono 100zł, wpłacono 100zł). Jednak brak było mechanizmów synchronizacji spowodował, że doszło do zakłamania danych.