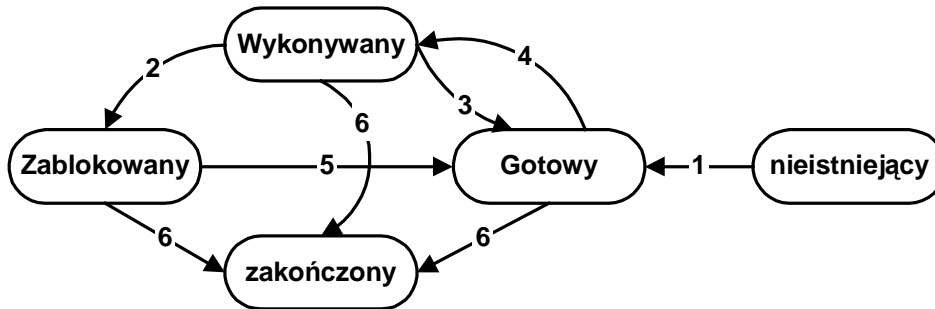


1 Procesy – pojęcia podstawowe

1.1 Kanoniczne stany procesów

Proces może być w jednym z trzech podstawowych stanów:

- Wykonywany (*ang. Running*),
- Gotowy (*ang. Ready*)
- Zablokowany (*ang. Blocked*).



Rys. 1 Przejścia pomiędzy podstawowymi stanami procesów

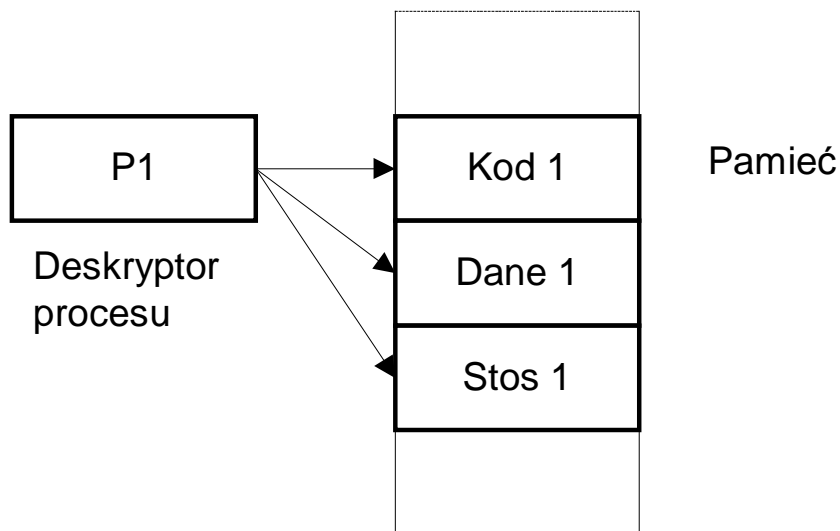
Pokazane na rysunku przejścia mają miejsca w następujących sytuacjach.

Utworzenie procesu

1. Proces utrzymuje w pamięci następujące struktury danych:
2. Proces żąda zasobu który nie jest dostępny.
3. Wystąpiło przerwanie (proces został wyłączone) lub też proces dobrowolnie zwolnił procesor.
4. Procedura szeregująca zdecydowała że ten proces ma być wykonywany.
5. Zasób którego brakowało do kontynuacji procesu stał się dostępny. Przejście zostało zainicjowane przez przerwanie od urządzenia wejścia / wyjścia lub też proces aktualnie wykonywany.
6. Zakończenie procesu

Struktury danych używane przez proces

1. Segment kodu - (*ang. code segment*) segment pamięci w którym przechowywane są instrukcje kodu maszynowego procesu.
2. Segment danych - (*ang. data segment*) segment pamięci w którym przechowywane są statyczne dane procesu (statyczne znaczy tyle że dane te istnieją poprzez cały czas istnienia procesu)
3. Segment stosu - (*ang. stack segment*) segment pamięci w którym przechowywane są chwilowe dane procesu. Na stosie utrzymywane są zmienne lokalne procedur, parametry procedur i inne chwilowe dane. Przydział i zwalnianie pamięci na stosie odbywa się automatycznie.
4. Deskryptor procesu - (*ang. process descriptor*) rekord w którym system operacyjny utrzymuje wszystkie informacje niezbędne do zarządzania procesem.



Rysunek 1-1 Struktury danych procesu

1.2 Deskryptor procesu

System operacyjny musi prowadzić administrację procesami. Procesy są tworzone, wykonywane, wznawiane, zawieszane i kończone. Muszą być utrzymywane struktury danych zawierające wszystkie informacje ku temu niezbędne.

Zawartość deskryptora:

Tożsamość procesu:

- Identyfikator procesu – PID procesu, grupa procesów
- Uwierzytelnienia – identyfikator użytkownika i grupy
- Indywidualność (ang. *personality*) – z cechy tej korzystają biblioteki emulacji

Środowisko procesu

Na środowisko procesu składają się dwa wektory:

- Wektor argumentów – wykaz argumentów wywołania programu
- Wektor środowiska – zestaw par NAZWA=WARTOŚĆ

Środowisko procesu dziedziczone jest z procesu macierzystego.

Kontekst procesu

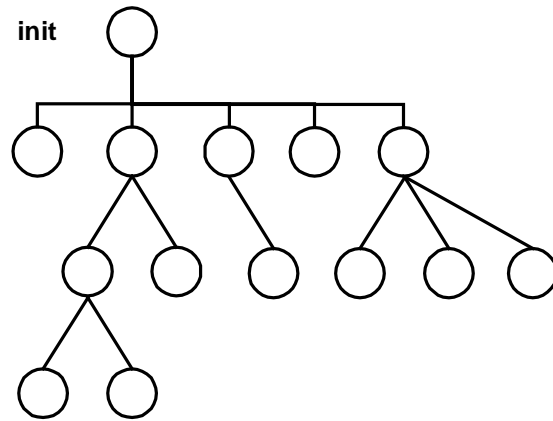
Kontekst procesu zawiera informacje:

- Kontekst planowania: rejestry, priorytet bieżący, nie obsłużone sygnały
- Informacje rozliczeniowe: skumulowany czas procesora
- Tablica plików: zawiera wskaźniki do otwartych plików utrzymywanej przez jądro.
- Tablica obsługi sygnałów: tablica specyfikująca akcje mające być wykonane przy nadejściu sygnałów
- Kontekst pamięci wirtualnej: opisuje zawartość przestrzeni adresowej procesu

1.3 Drzewo procesów

Procesy tworzą drzewo

- Każdy proces ma dokładnie jeden proces macierzysty
- Proces może mieć wiele procesów potomnych



Rysunek 1-2 Drzewo procesów