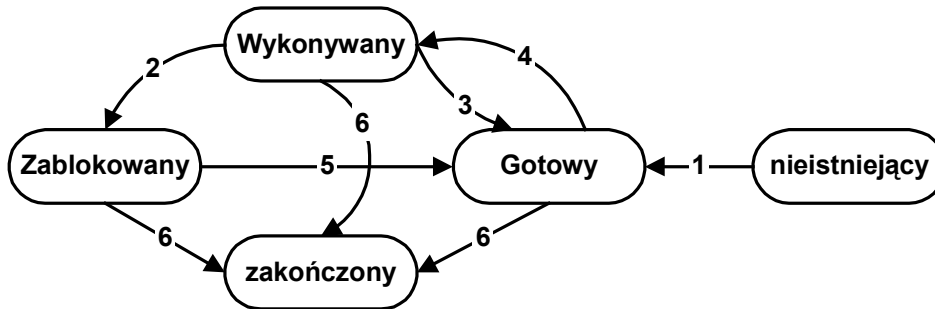


Procesy – pojęcia podstawowe

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

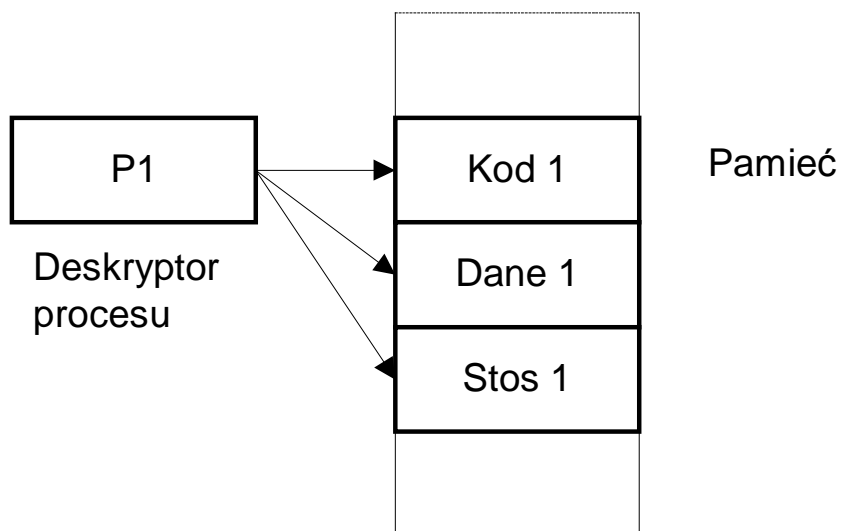
Pokazane na rysunku przejścia mają miejsca w następujących sytuacjach.

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

2 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. Segment sterty - (*ang. heap*) segment pamięci w którym przechowywane są chwilowe dane procesu jawnie przydzielane i zwalniane przez programistę.
5. Deskryptor procesu - (*ang. process descriptor*) rekord w którym system operacyjny utrzymuje wszystkie informacje niezbędne do zarządzania procesem.



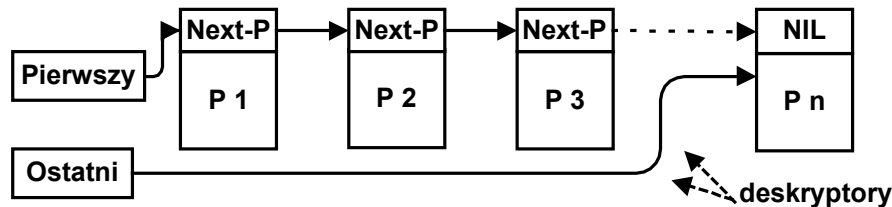
Struktury danych procesu

3 Deskryptor procesu

System operacyjny musi prowadzić administrację procesami.

Procesy są tworzone, wykonywane, wznowiane, zawieszane i kończone.

Muszą być utrzymywane struktury danych zawierające wszystkie informacje ku temu niezbędne.



Rys. 2 Kolejka deskryptorów procesów

Zawartość deskryptora:

Zarządzanie procesami:

- Identyfikator procesu – PID (*ang. Process Identifier*)
- Bieżący stan procesu (wykonywany, gotowy, zablokowany, itd. ...)
- Wskaźniki do poprzedniego i następnego deskryptora w kolejce wszystkich deskryptorów.
- Wskaźniki do poprzedniego i następnego procesu w danej kolejce (procesów gotowych, zablokowanych, itd...).
- Informacje dla procedur szeregowania (priorytet procesu, typ szeregowania).
- Informacje dotyczące obsługi sygnałów (sygnały dostarczone, zablokowane, ...).
- Informacje na temat hierarchii procesów (proces macierzysty, potomne, itd...).
- Kontekst sprzętowy procesu (rejstry procesora).
- Informacje rozliczeniowe o czasie procesora zużytym przez proces.
- Nazwa pliku z którego utworzono proces.
- Informacje uwierzytelniające jak rzeczywisty i efektywny identyfikator użytkownika i grupy (UID, GID, EUID, EGID).

Zarządzanie pamięcią:

- Rozmiar segmentu kodu, danych, stosu.
- Położenie segmentu kodu, danych, stosu.
- Informacje o stronach zajmowanych przez proces.

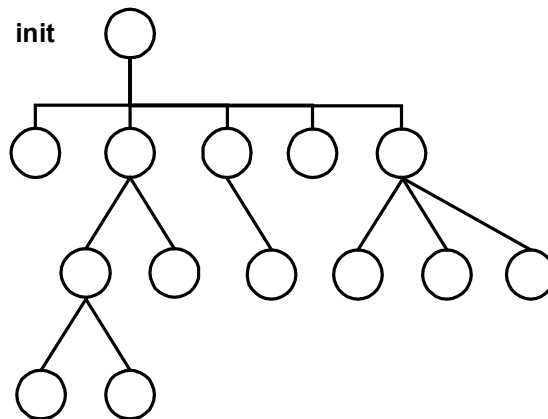
Zarządzanie plikami:

- Katalog bieżący.
- Katalog macierzysty.
- Informacja o terminalu sterującym.
- Wzorzec tworzenia nowych plików (UMASK)
- Wskaźnik na tablicę deskryptorów otwartych plików.

4 Drzewo procesów

Procesy tworzą drzewo

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

**5 Elementarne operacje na procesach**

Aby można było zbudować pełny system zarządzania procesami należy dysponować następującymi operacjami:

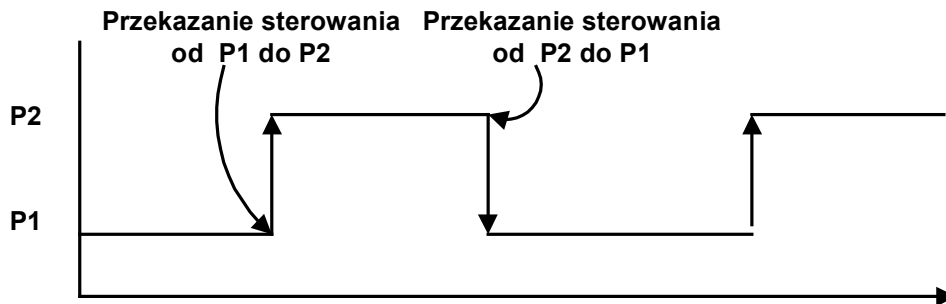
- Utworzenie procesu
- Synchroniczne przekazanie sterowania od procesu bieżącego do procesu P_i .
- Asynchroniczne przekazanie sterowania od procesu bieżącego do procesu P_i .
- Zakończenie procesu

Utworzenie procesu

Operacja powoduje utworzenie deskryptora i alokację pamięci niezbędnej dla procesu.

Synchroniczne przekazanie sterowania od procesu bieżącego do procesu P_i .

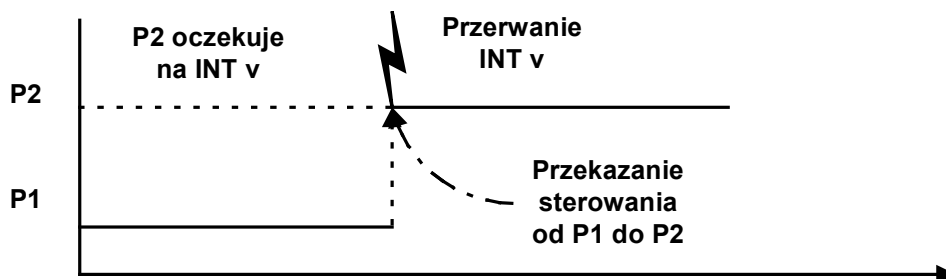
Przekazanie sterowania od procesu bieżącego do procesu P_i następuje na żądanie procesu bieżącego. Kontekst procesu bieżącego zostaje zachowany a kontekst procesu P_i odtworzony.



Rys. 3 Synchroniczne przekazanie sterowania pomiędzy procesami

Asynchroniczne przekazanie sterowania od procesu bieżącego do procesu P_i .

Przekazanie sterowania ma miejsce gdy występuje przerwanie. Proces bieżący jest zawieszany a sterowanie przekazywane jest do procesu P_i zainstalowanego wcześniej do obsługi tego przerwania.



Rys. 4 Asynchroniczne przekazanie sterowania pomiędzy procesami

Zakończenie procesu

Przy zakończeniu procesów następuje

- zwrot zasobów
- skasowanie deskryptora procesu
- przekazanie sterowania do innego procesu