

Zadanie: kontrolowanie zakończenia programu

Należy przygotować system współpracujących ze sobą procesów (spokrewnionych). Proces główny jest odpowiedzialny za „zdobycie” zasobów, koniecznych do wykonania zadania, i uruchomienie podwykonawców (osobnych procesów). Podwykonawcy działają równolegle, przez określony czas (dla każdego osobno). Po zakończeniu wszystkich podwykonawców, proces główny wypisuje podsumowanie.

Uwagi. Każdy proces powinien raportować swoje działania w celach diagnostycznych. Można wykorzystać strumień diagnostyczny albo dodatkowe pliki.

1. Zasoby

Zasobami może być cokolwiek. W tym wypadku będą to procesy potomne, które będą mogły odmówić działania.

Pozyskanie zasobu. Zasobem będzie podproces, uruchamiany przez program główny. Podproces ma załadować osobny program. W programie tym, na początku jest losowane, czy ma on działać dalej. Prawdopodobieństwo sukcesu ma być 40%.

Proces główny, aby pozyskać zasób, musi uruchomić podproces a następnie sprawdzić, czy zdecydował się on dalej działać. Powinien to zrobić w oparciu o *nieblokującą* wersję *waitid* lub *waitid*. (Uwaga. Jak zapobiec niekorzystnym zależnościom czasowym?)

Jeżeli nie udało się pozyskać zasobu, to proces główny ponawia próbę. Po trzech nieudanych próbach proces główny kończy się niepowodzeniem.

Zwolnienie zasobu. Dla każdego zdobytego zasobu rejestrowana jest właściwa procedura jego zwrócenia. Ponieważ w każdym przypadku będzie ona taka sama, należy zastosować jedną funkcję (parametryzowaną) i rejestrację za pomocą *on_exit*.

Procedura zwalniania zasobu polega na zabiciu właściwego procesu. Jednakże powinna być przygotowana na sytuacje awaryjne. W tym przypadku, na „zniknięcie” docelowego procesu. Taką sytuację powinna zaraportować.

2. Podwykonawcy

W systemie mają być dwaj podwykonawcy. Zadaniem podwykonawców jest raportowanie, co 1/3 sekundy, czy zasoby są dostępne.

Kod pierwszego podwykonawcy ma być zawarty w programie głównym, a drugiego ma być ładowany z innego pliku. (Jaśniej: w przypadku pierwszego **nie może**, a w przypadku drugiego **musi** być użyta jakaś funkcja z rodziny *exec**).

Obaj podwykonawcy znają identyfikatory zasobów oraz każdy ma osobny, dodatkowy parametr: ile czasu ma działać (w przybliżeniu). Parametry te są losowane przez proces nadrzędny. Dla uproszczenia:

- możliwe wartości, to 2 i 3 sekundy,
- podwykonawcy muszą mieć różne parametry.