

## SOI Zadanie na laboratorium 2.

Grupa pn. 10:15-12:00 CS500

1. Zaimplementować nowy algorytm szeregowania procesów w systemie Minix.
2. Zademonstrować działanie algorytmu przy pomocy odpowiednich programów testowych.

Wymagania odnośnie algorytmu:

- Zmieniane jest wyłącznie szeregowanie procesów użytkownika – specjalne zasady szeregowania procesów systemowych (np. zadań) nie ulegają zmianie.
- Należy wprowadzić podział procesów użytkownika na trzy klasy: zwykłe, uprzywilejowane i tła.
  - Procesy uprzywilejowane otrzymują (łącznie) połowę całego czasu procesora (algorytm daje przyspieszenie gdy jest tylko jeden taki proces, co jest domyślnym zastosowaniem, jednak nie blokujemy użytkownikowi możliwości uruchomienia większej liczby takich procesów, choć to mało sensowne).
  - Procesy tła otrzymują czas procesora wyłącznie wtedy, gdy nie ma żadnych innych procesów czekających na wykonanie.
- Nie należy marnować czasu procesora – niedopuszczalna jest sytuacja, w której w kolejce są procesy gotowe do wykonania, a żaden nie otrzymuje czasu procesora.
- Zasady podziału czasu procesora między procesy tłumaczy bardziej szczegółowo poniższa tabela (OBECNE znaczy nie tylko, że procesy danego typu istnieją, ale że są w kolejce gotowe do wykonania):

L.p.	Procesy zwykłe	Procesy uprzywilejowane	Procesy tła	Zasady
1	<b>OBECNE</b>	brak	brak	Procesy wykonują się normalnie, jak w oryginalnym algorytmie przed modyfikacją.
2	brak	<b>OBECNE</b>	brak	
3	brak	brak	<b>OBECNE</b>	
4	<b>OBECNE</b>	<b>OBECNE</b>	brak	Połowa czasu procesora alokowana jest dla procesów uprzywilejowanych. Wewnątrz każdej z grup szeregowanie działa normalnie.
5	<b>OBECNE</b>	brak	<b>OBECNE</b>	Wykonują się tylko procesy zwykłe, wg normalnego algorytmu.
6	brak	<b>OBECNE</b>	<b>OBECNE</b>	Wykonują się tylko procesy uprzywilejowane, wg normalnego algorytmu.
7	<b>OBECNE</b>	<b>OBECNE</b>	<b>OBECNE</b>	Połowa czasu procesora alokowana jest dla procesów uprzywilejowanych, druga połowa obsługuje procesy zwykłe. Procesy tła nie wykonują się. Wewnątrz każdej z grup szeregowanie działa normalnie.

Wszystkie powyższe reguły mają charakter przybliżony. Nie jest błędem, jeśli np. przy pięciu procesach zwykłych i jednym uprzywilejowanym, proces uprzywilejowany dostanie raz 45% a raz 60% czasu. Podobnie, jeśli proces testowy wykonuje też operacje we/wy, to możliwe jest, że proces

tła dostanie przez chwilę procesor mimo istnienia procesów uprzywilejowanych, bo akurat żaden z nich nie jest gotowy do wykonania – to jednak zdarzenie rzadkie.

Należy zapewnić możliwość ustawiania klasy procesu – najlepiej w formie programu umożliwiającego jej zmianę dla istniejącego procesu (patrz działanie kill, czy nice), albo programu uruchamiającego (patrz nohup). W ostateczności dopuszczalne jest dostarczenie biblioteki umożliwiającej procesowi samodzielne ustawienie swojej klasy.

Koncepcja powinna opisywać:

- sposób realizacji wymagań – gdzie będą potrzebne zmiany i jak będą realizowane,
- sposób demonstracji rezultatów – w jaki sposób zostanie wykazane prawidłowe działanie algorytmu.