

# Laboratorium 2. - SOI

## Robert Wojtaś

### Koncepcja wykonania zadania

## 1. Co należy zrobić?

Proszę zrealizować algorytm szeregowania dzielący procesy użytkownika na grupy: A, B.

Zakłada się, że proces jest umieszczony w grupie:

A - gdy jego identyfikator procesu jest podzielny bez reszty przez 2,

B - gdy jego identyfikator procesu nie jest podzielny bez reszty przez 2.

Wykonać niezbędne modyfikacje funkcji systemowych umożliwiającą przenoszenie procesów pomiędzy powyższymi grupami.

Proszę także wykonać usługę systemową o prototypie:

```
int set_scheduler(int x);
```

Która ustala proporcje czasowe w jakich scheduler ma wybierać do wykonania między zadaniami z grupy A i B.

Argument x może przyjmować wartości 0...100, i oznacza on ile procent czasu dostanie zadanie A (zadanie B dostanie odpowiednio 100-x procent czasu).

Usługa ma zwracać 0 gdy udało się wykonać zmianę proporcji czasowych, gdy zwróci -1, oznaczać będzie, że proces wywołujący tę usługę został przydzielony do klasy B i procesom tej klasy nie wolno zmieniać proporcji.

## 2. Koncepcja realizacji zadania

Każdy proces typu USER przypisany zostanie do jednej z grup: A lub B. Grupa jest określana w trakcie tworzenia procesu na podstawie przydzielonego identyfikatora. W późniejszej fazie życia procesu możliwa będzie zmiana grupy procesu.

Szeregowanie procesów realizowane będzie w ramach jednej kolejki. Różnicę stanowił będzie czas wykonania procesu – zależny od grupy, do której proces został przypisany. Proporcje czasowe ustalić można w odpowiednim wywołaniu systemowym. Przydzielenie kwantu czasu odbywać się będzie zgodnie z treścią zadania.

### EDYCJA PLIKÓW SYSTEMOWYCH:

- w strukturze **proc.h**
  - dodanie pola "group" wypełnianego jedna z dwóch predefiniowanych wartości GROUP\_A - 0 i GROUP\_B - 1.
- w pliku **kernel/proc.c**:
  - \* w funkcji 'sched()': zachowuję jedną kolejkę dla procesów z grupy A i B
  - \* w funkcji 'ready(rp)': należy umieścić proces w kolejce
- w pliku **kernel/system.c**:
  - Dla procesów typu USER w funkcji "do\_fork(m\_ptr)" zadbać o zainicjowanie grupy dla tworzonego procesu poprzez sprawdzenie identyfikatora procesu.
- w pliku **kernel/clock.c**

na końcu znajduje się procedura wywłaszczająca procesy grupy USER, które wykonują się za długo. Należy dokonać modyfikacji tej funkcji tak aby brała pod uwagę istnienie grup procesów oraz różne długości przydzielonego kwantu czasu.

### **USŁUGI SYSTEMOWE:**

Oba wywołania zostaną zrealizowane w module Memory Menager.

- **int set\_scheduler(int x);** - ustala proporcje czasowe w jakich scheduler ma wybierać wykonania między zadaniami z grup A i B.
- **int change\_group(void);** - wywołanie umożliwiające zmianę grupy na przeciwną, zwraca 0 gdy zmiana grupy przebiegła pomyślnie, -1 gdy operacja się nie powiodła.

### **TESTY:**

//TODO