

# Szeregowanie procesów | Koncepcja wykonania zadania

Damian D'Souza

1. Algorytm szeregowania dzielący procesy użytkownika na trzy grupy: A, B i C. Opracować funkcję systemową umożliwiającą przenoszenie procesów pomiędzy grupami. Procesy w grupie C otrzymują dwa razy więcej czasu niż procesy z grupy B.

Aby zaimplementować grupy procesów należy :

- Dodać pole zawierające informację o grupie do struktury `proc` w pliku `usr/src/kernel/proc.h`
- Zmodyfikować funkcję `do_fork` w pliku `usr/src/kernel/system.c`, aby domyślnie przydzielała procesom użytkownika grupę A
- Ustawić grupę każdego procesu na A w funkcji `main` w pliku `usr/src/kernel/main.c`
- Zdefiniować stałe czasowe dla grup w pliku `usr/src/kernel/clock.c` oraz sprawdzać grupę i zmieniać czas procesu w funkcji `do_clocktick` oraz `clock_handler`
- Dodać pętlę, która ustawi procesy w kolejce aby wykonywały się na przemian modyfikując funkcję `sched` w pliku `usr/src/kernel/proc.c`

Aby zaimplementować funkcję systemową umożliwiającą nadanie grupy procesom trzeba:

- Dodać wywołanie w `usr/include/minix/callnr.h`
- Dodać stałą wywołań w `usr/include/minix/com.h`
- Wykonać czynności wymagane do dodania wywołania z zad. 1, dodanie `no_sys` w tabeli fs, dodanie wywołania do tablicy, prototypu oraz implementacji wołającej `_taskcall` w module mm.
- Dodać implementację wywołania `do_setgroup` oraz wprowadzić wywołanie do wyrażenia switch w pliku `usr/src/kernel/system.c`

## 2. Testowanie

Aby przetestować implementację można za pomocą skryptu sh uruchomić kilka programów zawierających nieskończoną pętlę równolegle oraz przy użyciu F1 sprawdzać czas działania każdego procesu w kolumnie user, dla grupy C wartość ta powinna być zauważalnie większa niż B.