

## Treść zadania

---

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

Dodatkowo, proszę opracować funkcję systemową umożliwiającą przenoszenie procesów pomiędzy grupami.

Procesy w grupie B otrzymują dwa razy więcej czasu niż procesy z grupy A.

Grupa B może zawierać maksymalnie 3 procesy, grupa A maksymalnie 5, dodawanie kolejnych procesów powoduje usunięcie najwcześniej dodanego.

Zakładamy, że nowy proces domyślnie znajduje się w grupie A oraz że w grupie A znajduje się co najmniej 1 proces.

Opracować również łatwą metodę weryfikacji poprawności rozwiązania.

## Interpretacja

---

Sam podział na grupy A i B jest zrozumiały, lecz problem pojawia się przy usuwaniu najwcześniej dodanego po wykorzystaniu limitów. Aby nie ograniczać procesów użytkownika do 8 (i tym samym uniknąć przypadkowego zabicia ważnego procesu), założyłem, że usunięcie oznacza przydzielenie czasu procesora równego 0. Tym samym tak naprawdę będą 3 grupy:

B - mająca dwa razy więcej czasu niż grupa A

A - mająca "standardową" ilość czasu

C - mająca zerową ilość czasu

## Przydział do grup

---

Aby rozróżnić grupy dodam pole 'proc\_grp' do struktury 'proc' w jądrze. Będzie ona przyjmowała różne wartości w zależności od przynależności procesu do grupy.

- proc\_grp = 2 dla grupy B
- proc\_grp = 1 dla grupy A
- proc\_grp = 0 dla grupy C

Oznaczenie jest zgodne z założonymi proporcjami kwantów czasu dla danych grup.

## Zarys algorytmu

---

Aby odpowiednio zmienić szeregowanie zmodyfikuję nieco funkcję sched() w taki sposób aby sprawdzała przez ile kwantów czasu wykonywał się dany proces. Jeśli osiągnął on swój limit (zależny od grupy) - wrzucamy go na koniec kolejki. Jeśli nie - na początek. Potrzeba do tego jeszcze jednego pola w strukturze proc - 'proc\_cnt', która będzie przechowywać ile razy wykonał się proces. Jeśli 'proc\_cnt' zwiększone o 1 będzie mniejsze lub równe od 'proc\_grp' - wrzucamy proces na początek kolejki i inkrementujemy 'proc\_cnt'.

W przeciwnym wypadku - na koniec. W tak zaimplementowanym algorytmie procesy z grupy B powinny otrzymywać dwa kwanty czasu, z A jeden a z grupy C powinny być wrzucane zawsze na koniec kolejki.

## Domyślne wartości

---

Po utworzeniu procesu będzie on miał domyślnie ustawioną grupę A - `proc_grp = 1`.

## Przenoszenie procesów między grupami

---

Aby dobrze zaimplementować przenoszenie i uwzględnić limity będziemy potrzebować zmiennych trzymających liczbę procesów w danej grupie. Niech będą to 'group\_a\_count' i 'group\_b\_count'. Przy próbie dodania do grupy, która osiągnęła już limit, najstarszy proces będzie przenoszony do grupy C. Przenoszenie będzie umożliwiało wywołanie systemowe 'set\_proc\_gr'. Aby dowiedzieć się w jakiej grupie jest proces - dodam również wywołanie 'get\_proc\_gr'. Tym samym:

- przenoszenie procesów do grup A i B odbywa się manualnie przez syscall
- przenoszenie procesów do grupy C odbywa się automatycznie, jeśli próbujemy dodać proces do grup A lub B i został osiągnięty limit

Należy zapamiętać również czas powstawania procesu w strukturze 'proc' - tak aby wiedzieć jaki proces przenieść do grupy C.

## Planowane pliki do zmiany

---

### Wywołania systemowe

Analogicznie jak na laboratorium pierwszym:

- include/minix/callnr.h - zwiększenie liczby syscalli i dodanie nowych
- src/fs/table.c - dodanie pustych funkcji wywołań
- src/mm/proto.h - dodanie prototypów wywołań
- src/mm/table.c - dodanie nowych funkcji wywołań
- src/mm/main.c - definicja nowych wywołań

Dodawanie taskcalli i edycja plików jądra:

- include/minix/com.h - dodanie wpisów taskcalli
- src/kernel/system.c - dodanie prototypów taskcalli i ich definicje, przydzielanie standardowych wartości procesom w do\_fork()
- src/kernel/proc.h - dodanie pól do struktury proc
- src/kernel/glo.h - dodanie zmiennych globalnych określających ilość procesów w danej grupie
- src/kernel/proc.c - zmiana funkcji sched() zgodnie z ustalonym algorytmem

## Testowanie

---

### Testowanie czasów wykonania

Napisanie programu w C z użyciem funkcji time i sprawdzanie w jakim czasie wykona się proces z grupy A, a w jakim z grupy B.

## Testowanie zmian grup

Utworzenie kilku procesów i próba zmiany grup - sprawdzenie na tabeli procesów czy usuwa się najstarszy.

## Autor

---

Konrad Wojda, 310990