Systemy Operacyjne

Jakub Robaczewski

## Należy stworzyć trzy wywołania systemowe oraz funkcje je wywołujące

* + void setgroup(pid, group) -> Zmienia grupę (A, B, C) dla danego procesu
  + int set\_scheduler\_a(int x) -> Ustala % czasu działania grupy A
  + int set\_scheduler\_b(int x) -> Ustala % czasu działania grupy B

W szczególności wprowadzić odpowiednie zmiany do:

/usr/include/minix/callnr.h -> Definicje odwołań

/usr/src/mm/proto.h -> Prototypy funkcji

/usr/src/mm/table.c -> Odwołanie do funkcji

/usr/src/fs/table.c -> Adresy pusty

/usr/src/tools/main.c -> procedury obsługi

## kernel/proc.h

Dodać pole grupy (A, B lub C) do struktury

## kernel/system.c

**do\_fork()** -> zaimplementować domyślną wartość pola grupy

## glo.h

Dodajemy 2 zmienne globalne TIME\_A i TIME\_B, ustawiane przez set\_scheduler\_a() i set\_scheduler\_b()

## clock.c

**do\_clocktick()** -> należy zaimplementować wybór nowej grupy na podstawie pola grupy i zmianę sched\_tick. Nowy sched\_tick będzie przemnożony przez % czasu działania grupy.

**clock\_handler()** -> W tej funkcji są obsługiwane procesy niedziałające przez sched()

## Testowanie

* Po wywołaniu funkcji zmiany wyświetla grupę początkową i grupę końcową i PID procesu.
* Wypisywanie obecnego procesu i jego grupy po każdym ticku.
* Zliczenie ticków dla każdego procesu i wypisywanie po zmianie aktywnego procesu