

Zadanie 2
Systemy Operacyjne

Autor
Maciej Proszak

1 Koncept

1.1 Opis 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.

1.2 kernel/proc.h

Dodane zostaną definicje: GROUP_A, GROUP_B, GROUP_DELETED odpowiadające konkretnym grupom procesów.

Umieszczone zostaną również liczby reprezentujące kwanty czasu dla każdej grupy. Dla grupy A będzie to wartość 1, a dla grupy B wartość 2. Grupa Deleted otrzyma liczbę 0. W zadaniu jest określenie "dwa razy więcej czasu", więc równie dobrze grupa A może mieć np. wartość 5, a grupa B wartość 10.

1.3 kernel/proc.c

Funkcja sched() w tym pliku będzie miała największe znaczenie. W niej zaimplementujemy nowy algorytm szeregowania. Będziemy wyszukiwać odpowiednio procesu z grupy A lub B (w zależności jaką grupę aktualnie rozpatrujemy) i wrzucać na początek kolejki. Jeżeli odpowiedni proces nie zostanie znaleziony, wybieramy proces z drugiej grupy.

1.4 kernel/system.c

Do funkcji do_fork() dodam ustawienie odpowiedniej grupy dla nowo utworzonego procesu. W naszym przypadku nowy proces otrzyma domyślnie grupę A.

Również zostanie dodana obsługa funkcji SYS_SETPRI, która ustawi proces na odpowiednią grupę.

W przypadku do_fork oraz SYS_SETPRI musimy sprawdzić czy nowo dodany proces powinien usunąć najstarszy proces (zmienić grupę na "Deleted").

1.5 MM - memory manager

Analogicznie jak w poprzednim zadaniu (t1.txt) dodamy nowe wywołanie systemowe. Wywołanie systemowe (do_setpri) będzie dość proste i będzie polegało na wywołaniu innej funkcji za pomocą _taskcall(). Program testujący będzie uruchamiać to wywołanie w celu zmiany przydzielonej grupy.

1.6 Programy weryfikujące

Należy sprawdzić odpowiednie przydzielanie procesora do odpowiednich grup. Grupa B powinna z założenia mieć dostęp do czasu procesora dwa razy więcej niż grupa A. Uruchomione zostaną dwa procesy: jeden z grupy A drugi z grupy B, które będą posiadały pętle nieskończone. Po 10 sekundach zmierzemy ilość przydzielonych kwantów. Jeżeli algorytm jest poprawny, to ilość przydzielonych kwantów do procesu B będzie dwa razy większa niż dla drugiego procesu.

Drugi program testujący sprawdzi czy poprawnie usuwamy procesy z odpowiednich grup. Najpierw zostanie przetestowana grupa B. Utworzone zostaną 4 procesy grupy B. Proces, który został stworzony jako pierwszy nie będzie otrzymywać dostępu do procesora i ilość kwantów będzie stała. Na końcu zostanie sprawdzony wynik przekroczenia ilość procesów w grupie A.