

RAPORT

SOI – Laboratoria, Zadanie nr 2

Dominik Szaciłowski

Dołączone pliki:

- /usr/include/minix/callnr.h
- /usr/src/mm/mm_main.c
- /usr/src/kernel/system.c
- /usr/src/kernel/main.c
- /usr/src/kernel/clock.c
- /usr/src/kernel/proc.c
- /usr/src/kernel/proc.h
- /usr/src/fs/fs_main.c
- /usr/src/mm/mm_main.c
- /usr/src/mm/proto.h
- /root/set.c
- /root/long_program.c
- /usr/include/setsched.h

Wykonane kroki:

1. Modyfikacja funkcji pick_proc() w proc.c tak, aby przyznawała procesom ilość czasu w zmiennej sched_ticks w zależności od rodzaju grupy.
2. Modyfikacja funkcji do_clocktick w clock.c tak, aby ona nie przyznawała już wartości sched_ticks.
3. Nadanie początkowych wartości zmiennej precent i sched_ticks w main.c.
4. Zmiana wartości stałej SCHED_RATE na 100 i deklaracji do proc.h
5. Modyfikacja funkcji do_fork() w system.c tak, aby to ona nadawała wartości pola p_group.
6. Zaimplementowanie wywołania systemowego SETSCHED :
 - Dodanie numeru w com.h
 - Dodanie prototypu oraz case w system.c
 - Dodanie stałej w callnr.h
 - Modyfikacja fs_table.c
 - Dodanie nazwy w mm_table.c
 - Dodanie prototypu w proto.h
 - Implementacja funkcji w mm_main.c
 - Implementacja funkcji przyjaznej dla użytkownika w setsched.h oraz set.c
7. Dodanie funkcji testującej long_program.c

Opis wykonanych testów:

Do testu został wykorzystany program long_program.c, który składa się z 2 dużych pętli. Test polega na odpaleniu dwóch takich programów jednocześnie oraz pomiar ich czasów funkcją time w powłoce shell (time ./long_program.out & ; ls ; time ./long_program.out). ls został wywołany dla zmiany parzystości pidu drugiego odpalonego programu. Pomiary wykonujemy dla różnych wartości zmiennej precent, którą możemy zmieniać za pomocą programu ./set.out. Czas wykonania programu przy całkowitym użyciu procesora to 36.8s.

Wnioski:

Różnica czasów zwiększa się wraz z oddalaniem zmiennej precent od wartości 50. Jeden z programów będzie dążył do wartości oryginalnej czyli 36.8s, drugi będzie się od niej oddalał do wartości $2 \cdot 36.8$. Różnica czasów prawidłowo dąży do wartości 36.8s. Wyniki są zgodne co do zakładanego zachowania jednak nie całkowicie równe z zakładanymi wartościami. Zgodnie z założeniem dla $\text{precent} = p$ czas wykonania jednego z programów powinien wynosić $36.8 + 36.8 \cdot p/100$, drugiego $36.8 + 36.8 \cdot (100-p)/100$. Np. dla wartości $p = 70$, czas powinien wynosić 62s, a wynosi 66s. Opóźnienie pierwszego z procesów może wynikać z odpalenia równoległe do dwóch programów testowych również funkcji ls, która zabiera czas procesora. Następnym powodem przekłamania czasów jest to, że gdy pierwszy z procesów zakończy swoją pracę to drugi otrzymuje

100% czasu działania procesora przez co przyspiesza. Różnica wynika również z tego, że programy nie są odpalane dokładnie w tym samym momencie. Zaburza to dokładny pomiar.

Tabela Pomiarów

Wartość zmiennej precent	Czas działania pierwszego programu	Czas działania drugiego programu	Różnica
95	53s	80s	27s
90	57s	74s	17s
70	66s	75s	9s
50	71s	71s	0s
30	75s	65s	10s
10	74s	57s	17s
5	81s	53s	28s