

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра электронных вычислительных машин

Лабораторная работа №3
по дисциплине ОСиСП

«Взаимодействие и синхронизация процессов»

Выполнил:
студент группы 150501
Климович А.Н.

Проверил:
старший преподаватель
_____ Поденок Л.П.

Минск 2023

1 ПОСТАНОВКА ЗАДАЧИ

Разработать две программы – `parent` и `child`.

Синхронизация процессов с помощью сигналов и обработка сигналов таймера. Управление дочерними процессами и упорядочение вывода в `stdout` от них, используя сигналы `SIGUSR1` и `SIGUSR2`.

Действия родительского процесса:

По нажатию клавиши «+» родительский процесс (P) порождает дочерний процесс (C_k) и сообщает об этом.

По нажатию клавиши «-» P удаляет последний порожденный C_k, сообщает об этом и о количестве оставшихся.

При вводе символа «l» выводится перечень родительских и дочерних процессов.

При вводе символа «k» P удаляет все C_k и сообщает об этом.

При вводе символа «s» P запрещает всем C_k выводить статистику.

При вводе символа «g» P разрешает всем C_k выводить статистику.

При вводе символов «s<num>» P запрещает C_<num> выводить статистику.

При вводе символов «g<num>» P разрешает C_<num> выводить статистику.

При вводе символов «r<num>» P запрещает всем C_k вывод и запрашивает C_<num> вывести свою статистику. По истечению заданного времени (5 с, например), если не введен символ «g», разрешает всем C_k снова выводить статистику.

По нажатию клавиши «q» P удаляет все C_k, сообщает об этом и завершается.

Действия дочернего процесса:

Дочерний процесс во внешнем цикле заводит будильник и входит в вечный цикл, в котором заполняет структуру, содержащую пару переменных типа `int`, значениями {0, 0} и {1, 1} в режиме чередования.

При получении сигнала от будильника проверяет содержимое структуры, собирает статистику и повторяет тело внешнего цикла.

Через заданное количество повторений внешнего цикла (например, через 101) дочерний процесс, если ему разрешено, выводит свои `PPID`, `PID` и 4 числа — количество разных пар, зарегистрированных в момент получения сигнала от будильника.

Вывод осуществляется посимвольно (`fputc(3)`).

C_k запрашивает доступ к `stdout` у P и осуществляет вывод после подтверждения. По завершению вывода C_k сообщает P об этом.

Следует подобрать интервал времени ожидания и количество повторений внешнего цикла, чтобы статистика была значимой.

Сообщения выводятся в `stdout`.

Сообщения процессов должны содержать идентифицирующие их данные, чтобы можно было фильтровать вывод утилитой `grep`.

2 АЛГОРИТМ

- 1) начало.
- 2) установка собственных обработчиков на сигналы `SIGALRM`, `SIGUSR1` и `SIGUSR2`.
- 3) начало бесконечного цикла для работы с процессами.
- 4) ввод опции родительского процесса.
- 5) выполнение соответствующей опции.
- 6) вывод на экран результатов работы выбранной опции.
- 7) конец бесконечного цикла для работы с процессами.
- 8) конец.

3 ТЕСТИРОВАНИЕ ПРОГРАММЫ

```
$ ./parent
h - help
Enter:
+
Child C_1 is created, pid=6353
+
Child C_2 is created, pid=6354
+
Child C_3 is created, pid=6355
pid = 6353, ppid = 6352, {0, 0} = 0, {0, 1} = 0, {1, 0} = 2, {1, 1} = 1
pid = 6354, ppid = 6352, {0, 0} = 1, {0, 1} = 1, {1, 0} = 0, {1, 1} = 1
pid = 6355, ppid = 6352, {0, 0} = 0, {0, 1} = 0, {1, 0} = 1, {1, 1} = 2
-
Child C_3 is succesfully killed
2 processes are still running
l
Parent P, pid=6352
Child C_1, pid=6353
Child C_2, pid=6354
pid = 6353, ppid = 6352, {0, 0} = 2, {0, 1} = 1, {1, 0} = 2, {1, 1} = 1
pid = 6354, ppid = 6352, {0, 0} = 3, {0, 1} = 2, {1, 0} = 0, {1, 1} = 1
s
All child processes are blocked
g
All child processes are unblocked
pid = 6353, ppid = 6352, {0, 0} = 5, {0, 1} = 1, {1, 0} = 2, {1, 1} = 2
pid = 6354, ppid = 6352, {0, 0} = 4, {0, 1} = 2, {1, 0} = 0, {1, 1} = 4
p1
pid = 6353, ppid = 6352, {0, 0} = 5, {0, 1} = 1, {1, 0} = 3, {1, 1} = 2
k
Child C_2 is succesfully killed
Child C_1 is succesfully killed
All childs are killed
q
All childs are killed
```

4 ВЫВОД

Были разработаны две программы, одна из которых является родительским процессом, а вторая – ее дочерним процессом. Были реализованы следующие опции: «+», «-», «k», «h», «g», «s», «s<num>», «g<num>», «p<num>», «q», «l». В ходе работы были получены знания по работе с процессами и сигналами.