

Задача за многопроцесна работа

Да се разработи система от 5 процеса, работещи в среда Linux, всеки от които има следната функционалност:

Процес #1:

Това е основния процес, защото изпълнението на задачата започва с неговото стартиране и приключва при неговото завършване. Процесът реализира меню-програма, която има следните функции:

- Точките 1, 2 и 3 на менюто са за стартиране на процес 2, 3 или 4 (съответно).
 - При стартирането на всеки от процесите се задава цвят на текст, цвят на фон и брой милисекунди за пауза на между циклите на работа – тези данни се предават през командния ред към всеки от процесите.
 - Данните се избират от потребителя в процедурата за стартиране на съответния процес.
- Точките 4, 5 и 6 са за спиране и премахване (англ. kill) на процеси 2, 3 или 4 (съответно).
- Точка 7 е край на работата на програмата.
 - За да се може да се стартира тази точка от менюто е необходимо да няма други работещи процеси, т.е. процесите 2, 3, 4 и 5 да са спрени и премахнати.
- Стартирането на процес 5 става при стартиране на първия от процесите 2, 3 и 4.
 - Премахването на процес 5 става след премахване на последния от процесите 2, 3 или 4, но след като процес 5 завърши своята работа.
 - Всеки от процесите 2, 3 и 4 трябва да използва различен механизъм за комуникация с процес 5.

Процес #2:

Това е процес, чиято задача е да прочита цели числа, въвеждани от клавиатурата и да ги изпраща към Процес #5. След завършване на изпращането на прочитането цяло число, процесът трябва да изчака зададен брой милисекунди преди отново да чете ново цяло число (времетраенето на изчакването се задава като параметър от командния ред).

Особености на работата:

- Процесът се стартира от команда на Процес #1.
- Процесът се премахва от команда на Процес #1 или получава команда да завърши и да се премахне.
- През командния ред процесът получава три задължителни параметъра: цвят на текст, цвят на фон и цяло число, задаващо брой милисекунди за изчакване между отделните итерации на работния цикъл.
- При стартирането си, процесът извежда надпис на екрана с зададения цвят на текст и цвят на фон за да покаже, че е започнал работа.
- При завършването си, процесът извежда надпис на екрана с зададения цвят на текст и цвят на фон за да покаже, че е завършил работа.
- Комуникацията между Процес #2 и Процес #5 се реализира по произволно избран от студента метод за междупроцесна комуникация.

Процес #3:

Този процес има същата функционалност като Процес #2, но чете реално число от клавиатурата (Процес #2 чете цяло число).

Процес #4:

Този процес има същата функционалност като Процес #2, но чете стринг от клавиатурата (Процес #2 чете цяло число).

Процес #5:

Този процес има за задача да получава от процеси 2, 3 и 4 въведени от клавиатурата данни и да ги записва в предварително зададен текстов файл.

Особености на работата:

- Процесът се стартира от Процес #1.
- Името на файла, в който ще се работи се получава като параметър на командния ред. Ако вече съществува такъв файл, то трябва да се продължи записът във файла.
- Процесът трябва винаги да работи, когато работи някой от процесите 2, 3 и 4.
- Ако никой от процесите не работи, то тогава и този процес трябва да завърши своята работа.
- Когато процесът получи данни от процеси 2, 3 или 4, то тези данни се записват в текстов файл – всяка на самостоятелен ред с оказване на номера на процеса, от който е получена конкретната стойност.

Допълнителна литература:

1. Д. Минчев, Линукс – Системно програмиране, ISBN: 978-619-7126-69-3
а. Глава 4-7

<https://dimitarminchev.gitbooks.io/linux-system-programming/content/>

<https://dimitar-minchev.gitbook.io/linux-system-programming>

2. Аргументи от командния ред

<https://www.thegeekstuff.com/2013/01/c-argc-argv/>

3. Оцветяване на текст в терминалния прозорец:

<https://www.codeproject.com/Articles/5329247/How-to-change-text-color-in-a-Linux-terminal>