**Документация**

**Предмет:** CSCB534 Проект: Операционни системи (UNIX/Linux)

**Преподавател:**  гл. ас. д-р Росица Голева

**Изготвил:** Илина Йорданова Гергова

**Факултетен номер:** 103273

**Тема:** *„Направете скрипт с меню, който търси по ключови думи дадени във файл процеси и ги спира. Проверявайте за тези ключови думи cron jobs.“*

За целите на проекта съм използвала пакета Cygwin и Visual studio code за редактор.

Всеки файл е конвертиран в UNIX формат с командата dos2unix.

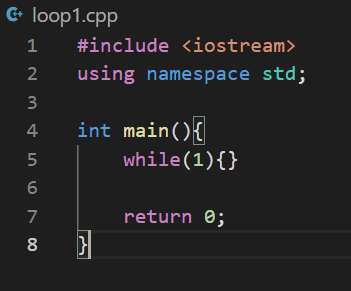
За скрипта е дадено разрешение за изпълнение

chmod +x projectScript.sh

Файлове в проекта:

* loop1.cpp
* loop2.cpp
* projectScript.sh
* helpFile – създава се от скрипта
* keyWords.txt

За целите на тестването на скрипта съм направила два cpp файла loop1.cpp и loop2.cpp с по един безкраен цикъл.



Компилирам ги с командата

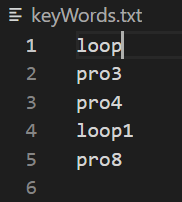
g++ -o loop1.exe loop1.cpp

(и g++ -o loop2.exe loop2.cpp ) и ги изпълнявам с

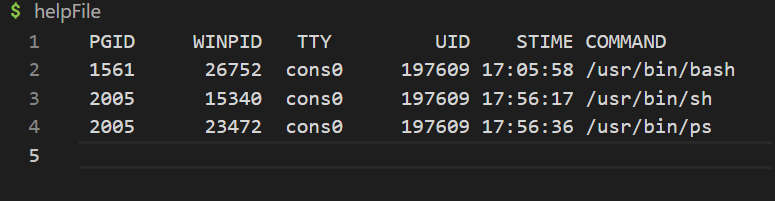
./loop1.exe &

(и ./loop2.exe & ) като знакът „&“ позволява процесите да действат на заден фон и да позлваме същия терминал, през който сме ги стартирали без да ги спираме.

Във файла keyWords.txt се съдържат всички ключови думи.



Използвам за помощен helpFile файл, където да запазя действащите прозеси.



За да видя тези процеси използвам командата ps aux, за да запаза резултатът на командата във файл, се използва < helpFile.

Цялата команда изглежда така

ps aux < helpFile

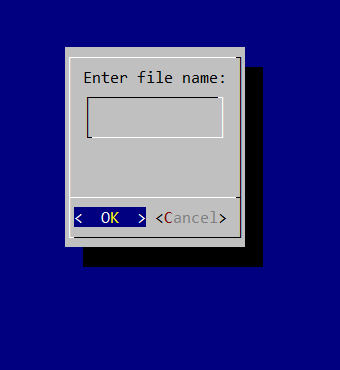
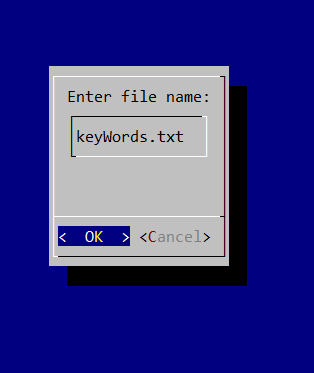
и се намира в скрипта.

Скриптът (намира се във файла projectScript.sh) представлява един цикъл, който обхожда файла с ключовите думи и меню, създадено чрез dialog.

След стартирането на скрипта

./projectScript.sh

се появява меню с поле за писане, където трябва да се посочи името на файла, съдържащ ключовите думи. Ако файлът не съществува, ще се изведе съответната грешка.

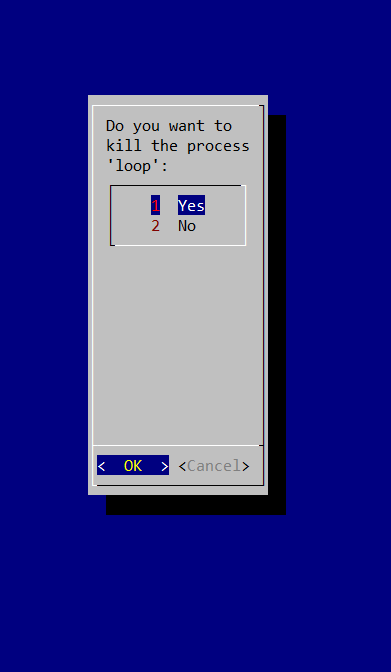
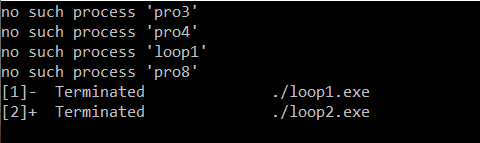


След успешното въвеждане и разпознаване на файла, цикълът в скрипта започва да итерира по редовете на файла с ключовите думи. Чрез

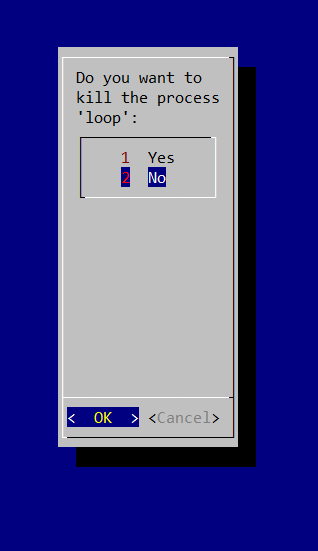
grep -q "$line" helpFile

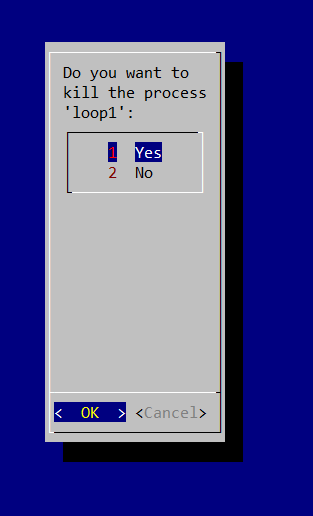
се проверява дали думата от текущия ред се съдържа във файла с процесите, тоест дали има процес с такова име. Ако е така се появява меню на екрана, което пита дали искаме да изтрием процеса. В противен случай се извежда текст, който казва, че няма процес със съответното име.

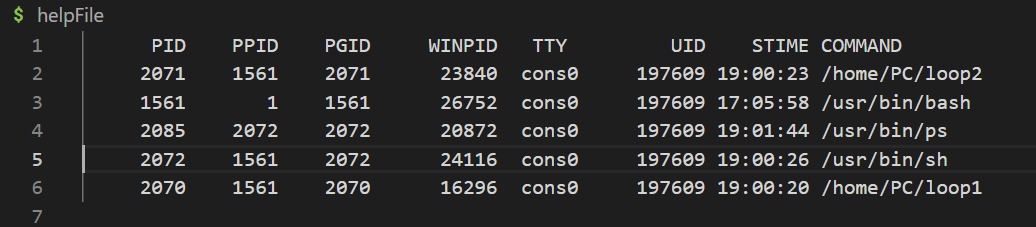
В моя пример първата ключова loop дума може да спре и двата процеса, които съм пуснала.



Ако откажем, тоест изберем опция 2, кодът ще продължи да търси и ще намери другата ключова дума loop1 , която е име на действащ процес



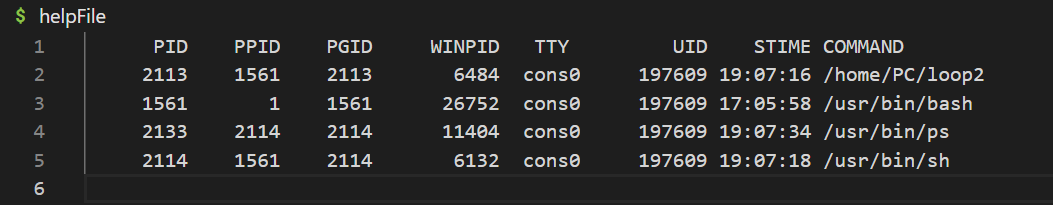


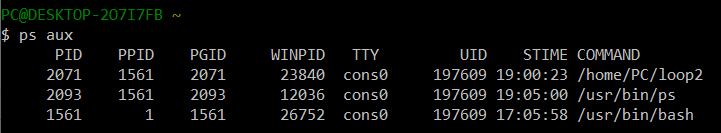
Ако откажем и тук, кодът ще продължи и ще попита пак, ако засече име на действащ процес, а тези процеси, които сме отказали, ще продължат да работят.

Ако приемем да изтрием loop1, то този процес ще бъде прекратен, но другия loop2 ще продължи да работи.

Съобщение, че процесът е спрян.





Информацията в помощния файл също е актуална.

В началото на цикъла се извършва проверяването на действащите процеси и запазването им във файл. Използвам помощен файл, защото за да намеря процес с име Х трябва да използвам командата

ps aux | grep –q “X” “$line”

но тази команда е поместена като условие на if, а тя винаги ще дава истина, защото grep винаги ще е действащ за ps.

За подобряването на кода първо е добре да се добави другата част от условието за намирането на cron jobs. След това, за да не се използва допълнително памет и да се премахне помощния файл, е възможно използването на командата pgrep, която показва и проверява едновременно. Може да се добавят проверки за колко думи има на ред и дали съществуват процеси с тези думи. Възможна е и промяна свързна с менюто и функционалността му – да се добавят повече опции. Например ако намери два процеса от една ключова дума, да попита индивидуално за всеки процес дали да бъде изтрит, а не и двата едновременно както е сега.

Източници:

* учебник Fundamentals of the UNIX System Version G.02 Hewlett Packard
* <https://stackoverflow.com/questions/8965509/why-if-ps-aux-grep-always-succeeds-in-bash>
* <https://unix.stackexchange.com/questions/104641/how-can-i-suppress-output-from-grep-so-that-it-only-returns-the-exit-status>
* <https://phoenixnap.com/kb/how-to-kill-a-process-in-linux>
* <https://askubuntu.com/questions/157779/how-to-determine-whether-a-process-is-running-or-not-and-make-use-it-to-make-a-c>
* <https://askubuntu.com/questions/420981/how-do-i-save-terminal-output-to-a-file>
* <https://askubuntu.com/questions/157075/why-does-ps-aux-grep-x-give-better-results-than-pgrep-x>
* <https://www.cyberciti.biz/faq/run-execute-sh-shell-script/>
* <https://www.cyberciti.biz/faq/run-execute-sh-shell-script/>
* <https://tldp.org/LDP/Bash-Beginners-Guide/html/sect_09_02.html>
* <https://www.cyberciti.biz/faq/unix-howto-read-line-by-line-from-file/>
* <https://askubuntu.com/questions/612315/how-do-i-search-for-a-process-by-name-without-using-grep>
* <https://www.cygwin.com/>