

OPERATING SYSTEMS BEISPIEL 2

Aufgabenstellung – encr

Implementieren Sie ein Programm, welches kontinuierlich Strings (Passwörter) von *stdin* einliest und diese von einem Kindprozess verschlüsseln lässt.

SYNOPSIS:

encr

Jedes Mal, wenn der Elternprozess eine Zeile eingelesen hat, wird ein eigener Kindprozess erzeugt, der das Passwort (exklusive des abschließenden Newline-Zeichens) mittels *crypt(3)* verschlüsselt. Danach soll der Kindprozess mittels *sleep(3)* eine längere (pseudozufällige) Verarbeitungsdauer simulieren, um das nichtblockierende Einlesen des Elternprozesses zu demonstrieren, und schließlich das verschlüsselte Passwort auf *stdout* ausgeben.

Anleitung

Lesen Sie in einer Schleife zeilenweise Strings von *stdin* ein und entfernen Sie das abschließende Newline (`\n`). Erzeugen Sie mittels *fork(2)* einen Kindprozess und verschlüsseln Sie in diesem das Passwort; der Elternprozess liest zu diesem Zeitpunkt bereits wieder von *stdin* ein. Beachten Sie, dass in dieser Situation mehrere Kindprozesse asynchron und in nicht vorhersehbarer Reihenfolge terminieren können!

Der Exit-Status jedes Kindes muss so früh wie möglich abgeholt werden. Behandeln Sie das Signal *SIGCHLD*, um das zu gewährleisten. Achten Sie darauf, dass *fgets(3)* den Fehlercode *EINTR* liefert, wenn es durch ein Signal unterbrochen wird. In diesem Fall muss *fgets(3)* erneut aufgerufen werden.

Testen

Testen Sie ihr Programm z.B. wie folgt

```
$ ./encr
jamesbond
micky
encr: jamesbond -> aCqHs040YLnq.
encr: micky -> aCSHDy093k9FQ
turing07
encr: turing07 -> aCJvjJDrpvXWE
asdfqwer
encr: asdfqwer -> aCZpdSZB6ohMQ
```

Die Ausgabe ihres Programmes muss nicht genau wie in diesem Beispiel formatiert sein. Die Reihenfolge von Input und Output wird vermutlich anders sein, ebenso wie das verschlüsselte Wort je nach gewähltem Salt variiert.

Beachten Sie beim Testen auch den Fall, in dem der Elternprozess durch ein Signal terminiert wird, während noch Kindprozesse aktiv sind. Auch in diesem Fall muss der Elternprozess auf das Terminieren seiner Kinder warten.

Richtlinien

Bitte beachten Sie auch die *Richtlinien für die Erstellung von C-Programmen* sowie die *Allgemeinen Hinweise zur Beispielgruppe 2* auf der Übungswebsite.

Insbesondere ist es ab dieser Beispielgruppe notwendig, die Dokumentation in Doxygen zu führen. Es muss zumindest das HTML-Output generierbar sein. Bitte dokumentieren Sie ausnahmslos alle Funktionen (auch `static`-Funktionen; siehe `EXTRACT_STATIC` in der Doxyfile). Eine kurze Einführung haben wir Ihnen auf http://wiki.vmars.tuwien.ac.at/index.php/Doxygen_Primer bereitgestellt. Achten Sie weiters darauf, dass nach außen hin sichtbare Funktionen (exportierte Funktionen) in der Header-Datei und lokale (`static`) Funktionen nur in der C-Datei dokumentiert werden. Sie sollten auch Ihre Typen (insbesondere `structs`), Konstanten und globale Variablen dokumentieren.