

OPERATING SYSTEMS BEISPIEL 2

Aufgabenstellung – hashsum

Schreiben Sie ein Programm, welches für alle Dateien in einem angegebenen Verzeichnis jeweils eine Zeile mit dem dazugehörigen MD5-Hashwert und dem Dateityp ausgibt. Dafür sollen die Standardtools `ls` (mit Parameter `-la`), `md5sum`, und `file` verwendet werden, indem diese als Kindprozesse per `fork()/exec()` aufgerufen werden.

SYNOPSIS:

```
hashsum [-i ignorepraefix] <directory>
```

Mit dem Parameter `-i` ("ignore") werden Dateien, die mit dem darauffolgenden Muster (nicht case-sensitive!) beginnen, ausgenommen; so führt ein Aufruf `hashsum -i . ~` dazu, dass alle versteckten Dateien im Home-Verzeichnis nicht behandelt werden.

Anleitung

Nachdem mit dem Kindprozess `ls` der gesamte Inhalt des angegebenen Verzeichnisses ermittelt und an den Elternprozess zurückgegeben wird, erzeugt man in einer Schleife für jede der von `ls` empfangenen Dateinamen jeweils einen Kindprozess, der `md5sum` mit diesem Namen als Argument ausführt, gleiches danach für `file`. Weiters werden die gewonnenen Informationen jeweils in einer Zeile (Format: *Dateiname md5summe dateityp*) ausgegeben. Beachten Sie, dass es sich nicht bei allen von `ls` gelisteten Einträgen um Dateien handelt (z.B. `.` und `..`); in dem Fall liefert dann `md5sum` einen nicht-erfolgreichen Return-Code zurück (welcher mittels `wait()/waitpid()` ermittelt werden kann) und der Eintrag soll in diesem Fall ignoriert werden.

Beispielaufruf

```
~ $ mkdir hashtests
~ $ cd hashtests
~/hashtests $ touch emptyfile
~/hashtests $ echo "das ist das haus vom nikolaus" > textfile
~/hashtests $ echo "compressme" | gzip -cf > compressed.gz
~ $ cd ..
~ $ ./hashsum ./hashtests
compressed.gz 35a0ec3087535c23863a774497881ada gzip compressed data, from Unix, [...]
emptyfile d41d8cd98f00b204e9800998ecf8427e empty
textfile 536b30bbea5ee48aca04807038e25875 ASCII text
```

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.