

OPERATING SYSTEMS BEISPIEL 2

Aufgabenstellung – calc

SYNOPSIS:

```
calculator
$> <zahl1> <zahl2> <operator>
```

BNF:

```
<zahl>      ::= -?[0-9]+
<operator> ::= +|-|*|/
```

Implementieren Sie die Funktionalität eines Taschenrechners als Elternprozess, der seinen Input von *stdin* als String einliest und den String an einen Kindprozess weiterleitet. Dieser extrahiert aus dem String die Operanden und die Operation, berechnet das Ergebnis und gibt dieses als String wieder zurück. Das Ergebnis wird vom Elternprozess anschließend am Bildschirm wieder ausgegeben. Da als Zahlen Integerwerte (maximal 65535) angenommen werden sollen, kann die Eingabezeichenkette auf 15 Zeichen beschränkt werden.

Beispielsweise soll die Eingabe `10 15 +` das Ergebnis 25 liefern.

Anleitung

Das Programm soll nach dem Starten in einer Schleife wiederholt Rechnungen von der Tastatur (*stdin*) einlesen. Zur Berechnung soll Ihr Programm einen Kindprozess erzeugen, an den es die Eingabe über eine Pipe weiterleitet. Dieser Kindprozess leitet nach der Umwandlung und Berechnung das Ergebnis über eine zweite Pipe an den Elternprozess zurück. Der Elternprozess gibt dieses Ergebnis wieder am Bildschirm (*stdout*) aus.

Dieser Vorgang soll solange wiederholt werden, bis der Elternprozess *EOF* (**Ctrl+D** von der Tastatur) liest. In diesem Fall ist die Pipe zum Kindprozess zu schließen. Der Kindprozess erhält dadurch beim Lesen ebenfalls *EOF* und terminiert. Der Elternprozess soll auf die Terminierung des Kindprozesses warten, alle benötigten Ressourcen (Pipes) an das System zurückgeben (schließen) und dann ebenfalls terminieren.

Sie können davon ausgehen, dass alle Rechnungen richtig eingegeben werden – es ist also keine Syntaxüberprüfung notwendig. Auch können Sie die Ergebnisse abrunden (Kommastellen abschneiden) – die Eingabe `5 2 /` ergibt als Ergebnis daher 2 anstelle von 2.5.

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.