

PUNKTEVERTEILUNG:

1	2	Σ

Aufgabe (1)

1.
 - a) **BSD** steht für Berkeley Software Distribution.
 - b) **Linux** wurde irgendwann zwischen den Releases 1.1.52 und 1.1.95 von 4.4BSD Lite 1 inspiriert, möglicherweise ging es dort um die `socket`-API. Linux entstand als Ersatz von Minix, wobei es aber nicht von dessen Code abstammt, sondern eine komplette Eigenentwicklung ist. In den Jahren hat es verglichen mit anderen Unices kaum Inspirationen aufgenommen, wobei aber das Filesystem-in-Userspace-Konzept von Plan9 oder der XFS-Code von IRIX genannt werden sollte. Eine recht neue Inspiration ist auch Solaris' ZFS, das zur Entwicklung von btrfs führte.
Mac OS X wurde nicht nur von BSD beeinflusst, es nutzt den XNU-Kernel der teils Mach-Code und teils BSD-Code nutzt. Der BSD-Code entstammt 4.3BSD und wurde später auf den Stand von FreeBSD 3.1 aktualisiert.
2.
 - a) Unmittelbare Vorfahr von **Mach 2.0** ist Mach, aus dem Jahre 1985.
 - b) **NeXTSTEP** wurde am 12. Oktober 1988 in Version 0.8 veröffentlicht. Das erste Mac OS X-Release, DP1 stammt von Mach 3 sowie vom NeXTSTEP und OPENSTEP-Nachfolger Rhapsody DR2 ab.
 Die Mac OS X-Entwicklung ging danach linear weiter wobei in Release 10.2 weiterer Code aus FreeBSD 4.4 importiert wurde. Das Release 10.3 wurde von 10.2.6 abgespalten; 10.4 wurde ursprünglich von 10.3.4 abgespalten und danach nochmal mit neuem Code aus 10.3.9 aktualisiert. Das Release 10.5 wurde jedoch nicht von einem 10.4.x-Release abgespalten, sondern von der Open Source-Variante des Mac OS X, Darwin 8.6.
 - c) **Darwin** ist die Open Source-Variante des Mac OS X. Die erste von einem Vorgänger von Mac OS X beeinflusste Version von Darwin ist 0.1, das von Rhapsody DR2 abgespalten wurde.
3.
 - a) Linux 1.0 wurde am 14. März 1994 released. Das erste Linux Release, 0.01 wurde am 1. August 1991 veröffentlicht, gefolgt von einer Menge weiterer Releases innerhalb von kürzester Zeit unter anderem 0.12 am 16. Januar 1992 0.95 am 8. März 1992 und etlichen 0.99-Releases im laufe des Jahres 1993 und 1994.
 - b) Der erste "Split" hat nach Linux 1.0.6 stattgefunden, wo die Entwicklung sich in die kurzlebige 1.0.x-Linie und die 1.1.x-Linie aufgeteilt hat.
 - c) Das letzte 2.0-Release war 2.0.40 am 8. Februar 2004. Zu dieser Zeit existierte bereits die Nachfolgerserie 2.2, mit dem Release 2.2.25 sowie deren Nachfolgerserie 2.4 mit Release 2.4.24. Das 2.6er-Release 2.6.1 war ebenfalls bereits freigegeben.
4.
 - a) Solaris ist der Nachfolger des SunOS, das in Version 4.0 von 4.3BSD Tahoe beeinflusst wurde und davor in SunOS 1.2 von 4.2BSD. Ursprünglich stammte SunOS von 4.1BSD ab.
 - b) Die einzigen im Diagramm erkennbaren Einflüsse sind die von UNIX Interactive 4.1 in Solaris 2.0 und viele Jahre später, Einflüsse von Linux 2.4.5

in Solaris 9 EA (wobei es sich wohl nicht um Code gehandelt hat, da Linux GPL2-lizenziert ist und eine Übernahme in Solaris nicht möglich gewesen wäre).

Aufgabe (2)

Es wird davon ausgegangen, dass die Arithmetischen Ausdrücke bereits Tokenisiert sind und in die richtigen Typen gewandelt wurden. Zudem wird angenommen, dass `lookup()` den Wert einer Variable bestimmen kann.

(a) P_1 :

```
def calculate(expression):
    # filter out parenthesis
    expression = [token for token in expression if token not in ('(', ')')]

    if expression[1] == '-':
        return lookup(expression[0]) - lookup(expression[2])
    elif expression[1] == '+':
        return lookup(expression[0]) + lookup(expression[2])
    else:
        raise ValueError('Such an expression is not allowed')
```

P_2 :

```
def calculate(expression):
    if len(expression) == 1:
        return lookup(expression[0])

    if expression[2] == '-':
        return calculate(expression[1]) - calculate(expression[3])
    elif expression[2] == '+':
        return calculate(expression[1]) + calculate(expression[3])
    elif expression[2] == '*':
        return calculate(expression[1]) * calculate(expression[3])
    elif expression[2] == '/':
        return calculate(expression[1]) / calculate(expression[3])
    else:
        raise ValueError('Such an expression is not allowed')
```

(b) P'_1 :

```
def calculate(expression):
    # filter out parenthesis
    expression = [token for token in expression if token not in ('(', ')')]

    concurrent_begin
        left = lookup(expression[0])
        right = lookup(expression[2])
    concurrent_end;
```

```

if expression[1] == '-':
    return left - right
elif expression[1] == '+':
    return left + right
else:
    raise ValueError('Such an expression is not allowed')

```

P'_2 :

```

def calculate(expression):
    if len(expression) == 1:
        return lookup(expression[0])

    concurrent_begin
        left = calculate(expression[1])
        right = calculate(expression[3])
    concurrent_end;

    if expression[2] == '-':
        return left - right
    elif expression[2] == '+':
        return left + right
    elif expression[2] == '*':
        return left * right
    elif expression[2] == '/':
        return left / right
    else:
        raise ValueError('Such an expression is not allowed')

```