Betriebs- und Kommunikationssysteme

WS 14/15

Prof. Dr-Ing. Volker Roth

Freie Universität Berlin

1. Übung

Ausgabe	Abgabe
17.10.14	31.10.14

Bitte bei der Abgabe Name der Mitglieder einer Gruppe, Nummer der Übung/Teilaufgabe und Datum auf den Lösungsblättern nicht vergessen! Darauf achten, dass die Lösungen beim richtigen Tutor abgegeben werden. Achten Sie bei Programmieraufgaben außerdem darauf, dass diese im Linuxpool kompilierbar sind. Nutzen Sie dazu die Flags `-std=c99`, `-Wall` und `-pedantic`. Es sollten keine Warnungen auftauchen.
Zu spät abgegebene Lösungen werden nicht berücksichtigt!

Aufgabe 1: Entwicklung der Betriebssysteme (2 Punkte)

Wie bereits in der Vorlesung angesprochen beschränken sich Betriebssysteme nicht nur auf Android, Linux, Mac OS oder Windows. Mit dieser Aufgabe sollen Sie sich einen Überblick über weitere Betriebssysteme verschaffen und diese nach Begriffen aus der Vorlesung einordnen.

Beispielsweise gibt es noch Multics, KeyKOS, EROS, Plan9, OpenBSD,Minix3 und Mach microkernel. Zur Auswahl des Betriebssystems berechnen Sie die Quersumme ihrer Matrikelnummer und rechnen Sie diese modulo sieben. Bei zwei gleichen Betriebssystemen wählen Sie bitte das nächstfolgende

Betrachten Sie dieses Betriebssystem genauer und fassen Sie dieses zusammen. Versuchen Sie folgende Fragen jeweils zu beantworten:

- Wann wurde das Betriebssystem entwickelt? Wird es noch weiterentwickelt?
- Hat das Betriebssystem einen besonderen Anwendungszweck?
- Ist eine spezielle Hardware nötig?
- Stellt es bestimmte Dienste (eng: *services*) bereit?
- Worin hebt es sich von anderen Betriebssystemen ab?

Beschreiben Sie kurz ihren Rechercheweg. Haben Sie orginal Quellen benutzt? Geben Sie eine vollständige Liste ihrer Quellen an und beachten Sie das korrekte Zitieren. Der Abgabe der Aufgabe (für zwei vorgestellte Betriebessysteme) sollte nicht mehr als zwei Seiten Beumfassen.

Aufgabe 2: Einführung in C: Zahlentypen (3 Punkte)

Die erste Teilaufgabe befasst sich mit den verschiedenen Zahlentypen. Die Programmiersprache C bietet Zahlentypen mit unterschiedlichen Größen, Wertebereichen und Präzisionen an, z.B. `char`, `int`, `long`, `float` und `double`. In C gibt es eine automatische (implizite) Typkonversion. Man kann also z.B. folgende Zuweisung machen:

```
int i=3;
float f=2.5;
...
f = i;
```

Implizite Typkonversionen stellen eine erhebliche Fehlerquelle dar. Man sollte daher die Konvertierung explizit durch sog. *casting* anweisen:

```
f = (float)i;
```

Zu beachten ist, dass innerhalb von arithmetischen Ausdrücken auf der rechten Zuweisungsseite ebenfalls Typkonversionen auftauchen können.

a) Führen Sie folgende Codezeilen aus und notieren Sie sich ihre Beobachtungen. Welche Ergebnisse der Typkonversionen sind fehlerhaft oder unerwartet?

Hinweis: An den Poolrechern ist dies mit den Befehlen `gcc -std=c99 -Wall -pedantic file -o outputfile` und dann mit `./outputfile` möglich.

```c
#include <stdio.h>

int main()
{
    float f;
    int i;
    long l;
    i = 1024;
    l = (long) i;
    printf("%d = %ld \n",i,l);
    l = 23;
    printf("l = %ld\n",l);
    i = (int) l;
    printf("%ld = %d\n",l,i);
    l = 17179869184;
    printf("l = %ld\n",l);
    i = (int) l;
    printf("%ld = %d\n",l,i);
    f = 5/2;
    printf("5/2 = %6.2f\n",f);
    f = (float)5/2;
    printf("5/2 = %6.2f\n",f);
    f = 1.333;
    i = (int)f;
    printf("1.333 = %d\n",i);
    f = 5.0/2;
    i = (int)f;
    printf("2.5 = %d\n",i);
    return 0;
}
```

b) Schreiben Sie ein Programm zur Umrechnung von Grad Celsius in Fahrenheit nach der genährten Formel.

$$t_F = \frac{9}{5} \cdot t_C + 32$$

Achten Sie dabei auf die Verwendung passender Datentypen.

c) Erweitern Sie das Programm so, dass eine Celsius-Fahrenheit–Umrechnungstabelle ausgegeben wird. Es sollen die Temperaturen $-30, -20, -10, \ldots, 100$ Grad Celsius umgerechnet und tabellarisch ausgegeben werden.