

Hardwarenahe Programmierung

Übungsaufgaben 2 – 31. Oktober 2024

Aufgabe 1: Seltsame Programme

Unter <https://gitlab.cvh-server.de/pgerwinski/hp/tree/2024ws/20241031> finden Sie (unter anderem) die Programme `test-1.c`, `test-2.c` und `test-3.c`.

Was bewirken diese Programme, und warum verhalten sie sich so?

Aufgabe 2: Kalender-Berechnung

Am 3. 1. 2009 meldete *heise online*:

Kunden des ersten mobilen Media-Players von Microsoft erlebten zum Jahresende eine böse Überraschung: Am 31. Dezember 2008 fielen weltweit alle Zune-Geräte der ersten Generation aus. Ursache war ein interner Fehler bei der Handhabung von Schaltjahren.

<http://heise.de/-193332>,

Der Artikel verweist auf ein Quelltextfragment (Datei: `aufgabe-2.c`), das für einen gegebenen Wert `days` das Jahr und den Tag innerhalb des Jahres für den `days`-ten Tag nach dem 1. 1. 1980 berechnen soll:

```
year = ORIGINYEAR; /* = 1980 */
```

```
while (days > 365)
{
    if (IsLeapYear (year))
    {
        if (days > 366)
        {
            days -= 366;
            year += 1;
        }
    }
    else
    {
        days -= 365;
        year += 1;
    }
}
```

Dieses Quelltextfragment enthält schlechten Programmierstil, nämlich mehrere Code-Verdopplungen:

- Die Anweisung `year += 1` taucht an zwei Stellen auf.
- Es gibt zwei unabhängige Abfragen `days > 365` und `days > 366`: eine in einer `while`- und die andere in einer `if`-Bedingung.
- Die Länge eines Jahres wird nicht durch eine Funktion berechnet oder in einer Variablen gespeichert; stattdessen werden an mehreren Stellen die expliziten numerischen Konstanten 365 und 366 verwendet.

Diese Probleme führten am 31. Dezember 2008 zu einer Endlosschleife.

Gut hingegen ist die Verwendung einer Konstanten `ORIGINYEAR` anstelle der Zahl 1980 sowie die Kapselung der Berechnung der Schaltjahr-Bedingung in einer Funktion `IsLeapYear()`.

- (a) Erklären Sie das Zustandekommen der Endlosschleife.
- (b) Schreiben Sie das Quelltextfragment so um, daß es die beschriebenen Probleme nicht mehr enthält.

Hinweis 1: Verwenden Sie für `IsLeapYear()` Ihre eigene Funktion aus Aufgabe 1 der letzten Übung.

Hinweis 2: Schreiben Sie zusätzlich eine Funktion `DaysInYear()`.

Aufgabe 3: Strings

Strings werden in der Programmiersprache C durch Zeiger auf **char**-Variable realisiert.

Wir betrachten die folgende Funktion (Datei: [aufgabe-3.c](#)):

```
int fun_1 (char *s1, char *s2)
{
    int result = 1;
    for (int i = 0; s1[i] && s2[i]; i++)
        if (s1[i] != s2[i])
            result = 0;
    return result;
}
```

- (a) Was bewirkt die Funktion?
- (b) Welchen Sinn hat die Bedingung „s1[i] && s2[i]“ in der **for**-Schleife?
- (c) Was würde sich ändern, wenn die Bedingung „s1[i] && s2[i]“ in der **for**-Schleife zu „s1[i]“ verkürzt würde?
- (d) Schreiben Sie eine eigene Funktion, die dieselbe Aufgabe erledigt wie `fun_1()`, nur effizienter.

Aufgabe 4: Programm analysieren

Wir betrachten das folgende C-Programm (Datei: [aufgabe-4.c](#)):

```
char*f="char*f=%c%s%c;main(){printf(f,34,f,34,10);}%;c";main(){printf(f,34,f,34,10);}
```

- (a) Was bewirkt dieses Programm?
- (b) Wofür stehen die Zahlen?
- (c) Ergänzen Sie das Programm derart, daß seine `main()`-Funktion `int main (void)` lautet und eine **return**-Anweisung hat, wobei die in Aufgabenteil (a) festgestellte Eigenschaft erhalten bleiben soll.