Allg. Informatik für WiWi (WS 2000)

Abgabetermin: 21. Dezember 2000

Beispiellösung

1 Geheimsache (15 Punkte)

Wir wollen uns ein wenig mit Kryptographie, d. h. der Verschlüsselung von Daten befassen. Eine sehr einfache Art der Verschlüsselung ist die sogenannte Cäsar-Chiffre. Hierbei werden die Buchstaben zyklisch verschoben. Die Anzahl der zyklischen Verschiebungen ist der Schlüssel, d. h. die geheime Information, mittels der die Nachricht auch wieder entschlüsselt werden kann. Haben wir als Schlüssel beispielsweise 3 gewählt, so wird aus dem Buchstaben `a` der Buchstabe `d` und so weiter und aus dem Buchstaben `z` wird der Buchstabe `c`. Man sieht also, dass als Schlüssel nur die Zahlen von 1 bis 25 sinnvoll sind, da im Falle des Schlüssels 0 der Text nicht verändert wird und alle anderen Schlüssel (d. h. Zahlen) zu einem dieser Schlüssel äquivalent sind (dies liegt daran, dass die Verschiebung zyklisch erfolgt und die Länge des Zyklus 26 ist).

(a) Schreiben Sie zunächst ein Modul `Encrypt`, das Zeichen für Zeichen von der Standardeingabe liest (solange noch ein Zeichen vorhanden ist). Falls das eingelesene Zeichen ein Buchstabe ist, soll es wie beschrieben verschlüsselt (d. h. zyklisch nach rechts verschoben) und danach ausgegeben werden. Alle anderen Zeichen sollen unverändert ausgegeben werden. Verwenden Sie für den Schlüssel eine Konstante, die Sie am Anfang der Moduldatei definieren. (10 Punkte)

(b) Schreiben Sie analog zu (a) ein Modul `Decrypt`, das die Standardeingabe entschlüsselt. Dazu müssen Sie nur eine zyklische Verschiebung in der entgegengesetzten Richtung durchführen. (2 Punkte)

(c) Sie bekommen die Datei *text.txt*, von der sie wissen, dass es sich um einen Text handelt, der mit dem hier vorgestellten Algorithmus verschlüsselt wurde. Finden Sie durch Probieren den verwendeten Schlüssel und den Originaltext heraus. (3 Punkte)
(Hinweis: Ändern Sie jeweils die Konstante in `Encrypt.om` ab und übersetzen Sie das Modul dann neu.)

Lösung:

```
DEFINITION Encrypt;

END Encrypt.
```

```
MODULE Encrypt;

  IMPORT Read, Write, Streams;

  CONST key = 3;

  VAR ch : CHAR;
      z  : INTEGER;

BEGIN
  Read.Char(ch);
  WHILE ~ Streams.stdin.eof DO    (* Ende, wenn kein Zeichen gelesen *)
    IF (ch >= "a") & (ch <= "z") THEN
      z  := ORD(ch) - ORD("a");   (* Buchstabe in Zahl aus 0 .. 25 umwandeln *)
      z  := (z + key) MOD 26;     (* zyklisch nach rechts verschieben *)
      ch := CHR(ORD("a") + z);    (* wieder in Buchstabe konvertieren *)
    ELSIF (ch >= "A") & (ch <= "Z") THEN
      z  := ORD(ch) - ORD("A");   (* Buchstabe in Zahl aus 0 .. 25 umwandeln *)
      z  := (z + key) MOD 26;     (* zyklisch nach rechts verschieben *)
      ch := CHR(ORD("A") + z);    (* wieder in Buchstabe konvertieren *)
    END;
    Write.Char(ch);
    Read.Char(ch);
  END;
END Encrypt.
```

```
DEFINITION Decrypt;

END Decrypt.
```

```
MODULE Decrypt;

  IMPORT Read, Write, Streams;

  CONST key = 3;

  VAR ch : CHAR;
      z  : INTEGER;
```

```
BEGIN
  Read.Char(ch);
  WHILE ~ Streams.stdin.eof DO    (* Ende, wenn kein Zeichen gelesen *)
    IF (ch >= "a") & (ch <= "z") THEN
      z  := ORD(ch) - ORD("a");   (* Buchstabe in Zahl aus 0 .. 25 umwandeln *)
      z  := (z - key) MOD 26;     (* zyklisch nach links verschieben *)
      ch := CHR(ORD("a") + z);    (* wieder in Buchstabe konvertieren *)
    ELSIF (ch >= "A") & (ch <= "Z") THEN
      z  := ORD(ch) - ORD("A");   (* Buchstabe in Zahl aus 0 .. 25 umwandeln *)
      z  := (z - key) MOD 26;     (* zyklisch nach links verschieben *)
      ch := CHR(ORD("A") + z);    (* wieder in Buchstabe konvertieren *)
    END;
    Write.Char(ch);
    Read.Char(ch);
  END;
END Decrypt.
```

Bei dem Text in Teilaufgabe (c) handelt es sich um die Bürgschaft von Schiller, die mit `Encrypt` unter Verwendung des Schlüssels 3 verschlüsselt wurde. Dies kann man wie folgt verifizieren:

```
turing$ Decrypt < text.txt | diff - schiller.ok
```

Die Datei `schiller.ok` ist bei Blatt 2 zu finden. Natürlich müssen Sie vor dem Übersetzen von `Decrypt.om` die Schlüsselkonstante auf 3 setzen. (Mit `diff` können Sie zwei Dateien byteweise vergleichen. Der besondere Dateiname `-` steht dabei für die Standardeingabe; hier wird also die Standardeingabe mit der Datei `schiller.ok` verglichen.)