



Inst. für Angew. Informationsverarbeitung

Prof. Dr. Franz Schweiggert  
Michaela Weiss  
Wolfgang Kaifler

15.02.2011  
Lösung 14

# Systemnahe Software I (WS 2010/2011)

**Augabetermin: 15.02.2011**

## 1 Deklarationen

Beschreiben Sie folgende Deklarationen:

- `int i;`  
Eine Variable des Typs Integer mit Namen `i`
- `double p[9];`  
Ein Array namens `p` bestehend aus 9 Elementen des Typs `double`
- `char (*g)();` Ein Zeiger auf eine Funktion Namens `g` die einen `char` zurückliefert

## 2 Fakultät

Schreiben Sie eine Funktion `fak(int i)` die einen Integer als Parameter übergeben bekommt und dessen Fakultätswert zurückliefert!

```
#include <stdio.h>
```

```
int fak(int i) {  
    if (i==0) return 0;  
    int fak = 1;  
    for(int j=1; j<= i; j++) {  
        fak *= j;  
    }  
    return fak;  
}
```

```
int main() {  
    int i = 5;  
    int j = 365;
```

```

printf("Die Fakultaeet aus %d betraegt %d\n", i, fak(i));
printf("Die Fakultaeet von %d ist angeblich %d\n", j, fak(j));
return 0;
}

```

### 3 Makro

Formulieren Sie Folgende Funktion als Makro:  $Betrag(n) = \begin{cases} n, & \text{falls } n > 0 \\ 0, & \text{falls } n = 0 \\ -n, & \text{falls } n < 0 \end{cases}$

```

#define BETRAG(n) (((n) > 0)? (n) : ((n) < 0)? -1*(n) : 0) oder
#define BETRAG(n) (((n) > 0)? (n) : ((n) < 0)? -(n) : 0)

```

### 4 Inode

Nennen Sie die Inhalte der Inode!

- Eigentümer
- Gruppe
- Dateityp
- Rechte
- letzter Lesezugriff auf den Dateiinhalt
- letzte Änderung am Dateiinhalt
- letzte Änderung an der Inode
- Linkcount
- Größe
- Blockadressen

### 5 Ein- und Ausgabe

```

#include <unistd.h>
#include <stdlib.h>
#include <fcntl.h>
#include <string.h>
#include <sys/types.h>

void die() {
    exit(1);
}

```

```

int main(int argc, char** argv) {
    //Parameteranzahl ueberpruefen
    if (argc < 4) {
        die();
    }

    //Datei ueberpruefen
    char* filename = argv[1];
    int fd = open(filename, O_RDWR);
    if (fd < 0) {
        die();
    }

    //Startposition ueberpruefen
    off_t pos = atoi(argv[2]);
    if (pos < 0) {
        die();
    }

    //Zeichenkette ueberpruefen
    char* s = argv[3];
    ssize_t len = strlen(s);
    if (len == 0) {
        exit(0);
    }

    //In Datei an gewuenschte Position springen
    if (lseek(fd, pos, SEEK_SET) < 0) {
        die();
    }

    //Gesamte Zeichenkette in Datei schreiben
    ssize_t written = 0;
    while (written < len) {
        ssize_t nbytes = write(fd, s + written, len - written);
        //Fehler beim write
        if (nbytes < 0) {
            die();
        }
        written += nbytes;
    }
    return 0;
}

```