ulm university universität

Klausur zu „Systemnahe Software I“ (20. Februar 2012)

WS 2011/2012

Dr. Andreas Borchert mit Markus Schnalke

Bearbeitungszeit: 120 Minuten

Nicht mit Bleistift oder Rotstift schreiben!

Name:	
Vorname:	
Matrikelnummer:	
Studiengang:	

Nr	Max	Bewertung	
1	**18**	xxxxx	
(a)	4		xxxxx
(b)	3		xxxxx
(c)	4		xxxxx
(d)	4		xxxxx
(e)	3		xxxxx
2	**18**	xxxxx	
(a)	4		xxxxx
(b)	4		xxxxx
(c)	3		xxxxx
(d)	3		xxxxx
(e)	4		xxxxx
3	**12**	xxxxx	
(a)	6		xxxxx
(b)	3		xxxxx
(c)	3		xxxxx

Nr	Max	Bewertung	
4	**12**	xxxxx	
(a)	8		xxxxx
(b)	4		xxxxx
5	**12**	xxxxx	
(a)	2		xxxxx
(b)	2		xxxxx
(c)	4		xxxxx
(d)	4		xxxxx
6	**14**	xxxxx	
7	**14**	xxxxx	
(a)	3		xxxxx
(b)	3		xxxxx
(c)	4		xxxxx
(d)	4		xxxxx
Summe	**100**		

Für Ihre Lösungen verwenden Sie bitte den freigelassenen Platz nach der Aufgabenstellung, die Rückseite der jeweiligen Aufgabe oder die angehängte leere Seite unter Angabe der Aufgabennummer. Prüfen Sie zu Beginn, ob Ihre Klausur aus 18 durchnummerierten Seiten besteht. Nennen Sie möglichst alle Annahmen, die Sie gegebenenfalls für die Lösung einer Aufgabe treffen!

Viel Erfolg!

Aufgabe 1 **(18 Punkte)** Programmier-Techniken

(a) 4 Punkte
Welchen Wert haben die folgenden arithmetischen Ausdrücke?

(a) `11 % 4 * 2`
(b) `7 >> 1`
(c) `17 / 3`
(d) `2 + 4 % 3`
(e) `1.0 / 4`
(f) `5 ^ 6`
(g) `0666 & ~022`
(h) `1 << 3 | 1 << 1`

Lösung:

(b) 3 Punkte
Gegeben sei die Deklaration

```
int x = 1, y = 4;
```

Geben Sie jeweils das Resultat der folgenden Ausdrücke an:

(a) `x <= y < 1`
(b) `x-y? x: y`
(c) `x = y = x + y`

Lösung:

(c) 4 Punkte
Jemand zeigt Ihnen die untenstehende Fassung der Funktion *init_complex* und beklagt sich darüber, dass sie keinen Effekt habe. Was ist falsch? Korrigieren Sie bitte die Funktion, wobei Sie bei Bedarf auch die Signatur (Parametertypen und Return-Wert) der Funktion verändern dürfen!

```
typedef struct {
   double real;
   double img;
} Complex;

void init_complex(Complex c) {
   c.real = 0;
   c.img = 0;
}
```

Lösung:

(d) 4 Punkte

Schreiben Sie eine Funktion *teiler_vielfachheit*, die zwei ganze Zahlen n und m des Typs `unsigned int` mit $m > 1$ erhält und zurückliefert, wie oft m als Teiler in n enthalten ist. Beispiele:

Aufruf	Ergebnis
`teiler_vielfachheit(7, 2)`	0
`teiler_vielfachheit(12, 2)`	2

Lösung:

(e) 3 Punkte

Bitte kreuzen Sie bei den folgenden Behauptungen zu den Datentypen *float* und *double* an, ob sie zutreffen oder inkorrekt sind:

Behauptung	trifft zu	ist falsch
0.125 ist präzise darstellbar	☐	☐
0.1 ist präzise darstellbar	☐	☐
Alle Werte des Typs *long* sind als *float* darstellbar	☐	☐
Alle ganzzahligen Werte des Typs *float* sind als *long* darstellbar	☐	☐
Es gibt genau eine Repräsentierung der 0	☐	☐
Bei der Berechnung von $\sum_1^n x_i$ sollten die x_i absteigend sortiert werden	☐	☐

Aufgabe 2 **(18 Punkte)** Funktionen und Strukturen

(a) 4 Punkte
Stellen Sie fest, ob C bei der Übergabe von Arrays als Parametern *call by value* oder *call by reference* unterstützt bzw. welche Repräsentierung diese Parameter haben. Zeigen Sie an einem kleinen Beispiel, wie eine Funktion mit einem als Parameter übergebenen *int*-Array mit 10 Elementen alle Elemente auf 0 initialisiert.

Lösung:

(b) 4 Punkte
Geben Sie bitte in den folgenden Fällen an, ob die Deklarationen zulässig sind und falls ja, welchen Typ *f* jeweils hat:

- `int (*f)(double);`
- `int *(*f)();`
- `int *(f*)();`
- `int (*f[])();`

Lösung:

(c) 3 Punkte
Spezifizieren Sie eine Datenstruktur *Student*, die ein Tupel repräsentiert, bestehend aus einer ganzzahligen Matrikelnummer ohne Vorzeichen, einem Namen beliebiger Länge und dem Geschlecht (männlich oder weiblich).

Lösung:

(d) 3 Punkte
Deklarieren Sie einen Typ `DoubleFunction` für einen Zeiger auf eine Funktion, die keine Parameter entgegennimmt und die `double` zurückliefert.

Lösung:

(e) 4 Punkte

Die Hofstadter-Conway-$10000-Folge $HC(n)$ für $n > 0$ ist folgendermaßen definiert:

$$\begin{aligned} HC(1) &= 1 \\ HC(2) &= 1 \\ HC(n) &= HC(HC(n-1)) + HC(n - HC(n-1)) \end{aligned}$$

Schreiben Sie in C eine rekursive Funktion hc, die $HC(n)$ für nicht-negative ganze Zahlen n mit $n > 0$ berechnet.

Lösung:

Aufgabe 3 **(12 Punkte)** Makros

(a) 6 Punkte

In C wird der zu übersetzende Programmtext durch den sogenannten Präprozessor gefiltert. Es seien die vier Dateien *main.c*, *b.h*, *c.h* und *d.h* gegeben (siehe unten). Geben Sie die Ausgabe des Präprozessors an, wie sie etwa durch das Kommando `gcc -E main.c` erzeugt wird. (Sie können dabei die Leerzeilen und durch den Präprozessor normalerweise erzeugten Zusatzzeilen mit Dateinamen und Zeilennummern weglassen.)

main.c

```
Anfang
#include "b.h"
FOO

#include "c.h"
BAR
#include "d.h"
FOO
```

b.h

```
#ifndef B_H
#define B_H

#define FOO BAR
Inhalt b

#endif
```

c.h

```
#include "b.h"
#ifdef FOO
#define BAR FOOBAR
Inhalt c
#endif
```

d.h

```
#define FOOBAR fb
Inhalt d
```

Lösung:

(b) 3 Punkte

Schreiben Sie ein Makro `DEGREES`, das ein Bogenmaß (Radiant) in ein Gradmaß konvertiert. Dabei darf die Definition des Makros *M_PI* vorausgesetzt werden, das π im Rahmen der zur Verfügung stehenden Genauigkeit näherungsweise darstellt. Entsprechend sollte `DEGREES(0)` den Wert 0 liefern und `DEGREES(M_PI)` den Wert 180. Schreiben Sie das Makro dabei so, dass ein Aufruf problemlos in einen größeren Ausdruck eingebettet werden kann.

Lösung:

(c) 3 Punkte

Gegeben sei folgender Programmtext:

```
#define MIN(a,b) (a < b? a: b)
int min(int a, int b) {
   return a < b? a: b;
}

void f() {
   int a = 1; int b = 2;
   printf("%d\n", MIN(a += 1, b));
}
void g() {
   int a = 1; int b = 2;
   printf("%d\n", min(a += 1, b));
}
```

Bitte kreuzen Sie bei den folgenden Behauptungen Makros an, ob sie zutreffen oder inkorrekt sind:

Behauptung	trifft zu	ist falsch
Die Klammern bei dem Makro *MIN* sind überflüssig.	☐	☐
Bei *MIN* sollten besser noch Klammern hinzugefügt werden.	☐	☐
Die Funktion *min* ist in ihrer Ausführung effizienter als das Makro *MIN* und daher vorzuziehen.	☐	☐
Das Makro *MIN* kann auch für Werte des Typs *double* verwendet werden.	☐	☐
f und *g* liefern die gleiche Ausgabe.	☐	☐
Rekursive Makrodefinitionen sind zulässig.	☐	☐

Aufgabe 4 **(12 Punkte)** Makefile

Es sei das folgende aus fünf Dateien bestehende C-Programm gegeben. Das Bild zeigt schemenhaft den Aufbau des Programms. Die Rechtecke stellen jeweils Dateien dar, deren Name rechts oberhalb des Rechtecks steht.

dirtree.h

```
#ifndef DIRTREE_H
#define DIRTREE_H

typedef struct DirNode {
   // ...
} DirNode;

DirNode* create_dirnode();
// ...

#endif
```

dirtree.c

```
#include "dirtree.h"

DirNode* create_dirnode() {
   // ...
}
// ...
```

scandir.h

```
#ifndef SCANDIR_H
#define SCANDIR_H

#include "dirtree.h"

DirNode*
scan_directory(char* dirpath);
#endif
```

scandir.c

```
#include "dirtree.h"
#include "scandir.h"

DirNode*
scan_directory(char* dirpath) {
   // ...
}
```

dircmp.c

```
#include "scandir.h"

// ...
int main(int argc, char** argv) {
   // ...
   DirNode* dir =
      scan_directory(argv[1]);
   // ...
}
```

(a) 8 Punkte
Schreiben Sie ein *Makefile*, das aus den gegebenen Dateien mit Hilfe des gcc-Compilers in ein ausführbares Programm mit dem Namen *dircmp* erzeugt. Das *Makefile* sollte sämtliche zu erkennenden Abhängigkeiten berücksichtigen. Entsprechend sollte bei einer Änderung einer der fünf Dateien ein anschließender Aufruf von *make* nur zur Neuübersetzung der Programmtexte führen, die zwingend neu übersetzt werden müssen.

Lösung:

(b) 4 Punkte

Was muss neu übersetzt werden, wenn die jeweils angegebene Datei verändert wird?

Veränderte Datei	Neu zu übersetzen
dirtree.c	
dirtree.h	
scandir.h	
dircmp.c	

Aufgabe 5 **(12 Punkte)** Dateisystem

(a) 2 Punkte
Kann ein Aufruf des Systemaufrufs *open* an der Existenz einer Datei scheitern, d.h. dass er erfolgreich gewesen wäre, wenn die Datei zuvor noch nicht existiert hätte? Erklären Sie kurz ihre Antwort.

Lösung:

(b) 2 Punkte
Sie haben eine Datei zum Lesen eröffnet und ein Aufruf von *read* für diese Datei liefert einen Return-Wert von 0. Was bedeutet dies?

Lösung:

(c) 4 Punkte

Sie eröffnen erfolgreich eine zuvor noch nicht existierende Datei mit

```
int fd = open("notizen.txt", O_WRONLY|O_CREAT|O_TRUNC, 0666);
```

und stellen anschließend fest, dass die Datei folgende Zugriffsrechte hat:

```
clonard$ ls -l notizen.txt
-rw-rw-r-- 1 borchert sai 0 Feb 17 10:57 notizen.txt
clonard$
```

- Stimmen diese Zugriffsrechte mit denen bei *open* angegebenen überein? Wenn nicht, woran liegt das?
- Welche Bedeutung hat *O_TRUNC*?

Lösung:

(d) 4 Punkte

Bitte kreuzen Sie bei den folgenden Behauptungen zu einem POSIX-Dateisystem, ob sie zutreffen oder inkorrekt sind:

Behauptung	trifft zu	ist falsch
Der Dateiname gehört zur Inode	☐	☐
Die Größe einer Datei gehört zur Inode	☐	☐
Die Zugriffsrechte gehören zur Inode	☐	☐
Eine Datei kann mehrere Besitzer haben	☐	☐
Dateien können verkleinert werden	☐	☐

Aufgabe 6 **(14 Punkte)** Ein- und Ausgabe

Schreiben Sie ein C-Programm namens *xit*, das den Inhalt einer Datei vollständig mit dem Zeichen „x“ überschreibt, wobei die Länge der Datei erhalten bleiben soll. Die zu verändernde Datei ist dabei als Kommandozeilenparameter zu übergeben.

Alle möglichen Fehler sind abzufangen. Im Falle eines Fehlers ist die Funktion *die()* aufzurufen, die Sie in Ihrer Lösung nicht mit implementieren müssen.

Sie dürfen die *stdio* hierbei nicht verwenden. Stattdessen sind nur Systemaufrufe zulässig. Und diese sollten hinreichend effizient benutzt werden, d.h. eine zeichenweise Ausgabe ist nicht gestattet. Grundsätzlich ist bei allen Einlese- und Schreiboperationen damit zu rechnen, dass weniger Bytes gelesen oder geschrieben werden als gewünscht. In diesem Falle darf *die()* nicht aufgerufen werden.

Beispiel:

```
clonard$ gcc -std=gnu99 -o xit xit.c
clonard$ echo Hallo >a
clonard$ ls -l a
-rw-rw-r-- 1 borchert sai 6 Feb 17 11:28 a
clonard$ cat a
Hallo
clonard$ xit a
clonard$ ls -l a
-rw-rw-r-- 1 borchert sai 6 Feb 17 11:28 a
clonard$ cat a; echo
xxxxxx
clonard$
```

Hinweise: Sie können auf die Angabe der `#include`-Anweisungen verzichten. Die Länge einer Datei können Sie dem Feld *st_size* eines Objekts des Typs *struct stat* entnehmen, das Sie mit einem geeigneten Systemaufruf zuvor befüllen müssen.

Lösung bitte auf nächster Seite!

Lösung:

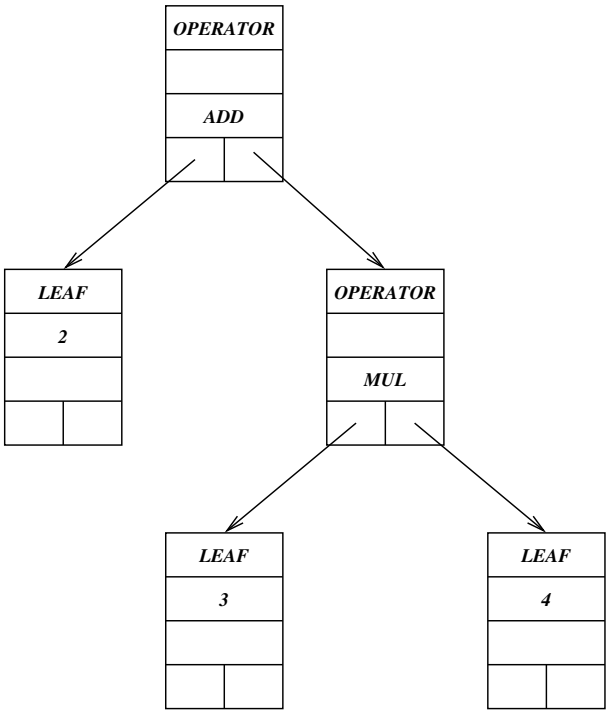
Aufgabe 7 **(14 Punkte)** Dynamische Datenstrukturen

Gegeben sei folgende Datenstruktur für binäre Ausdrucksbäume:

```
typedef enum {ADD = '+', SUB = '-', MUL = '*', DIV = '/'} Operator;
typedef struct expr {
   enum {LEAF, OPERATOR} kind;
   // if kind == LEAF
   double value;
   // if kind == OPERATOR
   Operator operator;
   struct expr* left;
   struct expr* right;
} Expression;
```

Das Feld *kind* legt fest, ob es sich um einen Blatt-Knoten mit einem Wert (*value*) handelt oder um einen Operator-Knoten mit dem binären Operator *operator* und den beiden Operanden *left* und *right*. Unterstützt werden nur die vier Grundrechenarten und die Operatoren werden durch den Aufzählungstyp *Operator* repräsentiert.

Der Ausdruck „2 + 3 * 4“ kann auf diese Weise repräsentiert werden:



Wenn die in den folgenden Teilaufgaben implementierten Funktionen zur Verfügung stehen, liefert

```
int main() {
   Expression* expr =
      create_expr(ADD,
         create_value(2),
         create_expr(MUL,
            create_value(3),
            create_value(4)));
   print_expr(expr);
   printf("\n%lg\n", eval_expr(expr));
}
```

die Ausgabe

```
(2 + (3 * 4))
14
```

(a) 3 Punkte
Schreiben Sie eine Funktion *create_value*, die einen Wert des Typs *double* erhält und einen entsprechenden Blatt-Knoten dynamisch erzeugt und einen Zeiger darauf zurückliefert. Wenn kein Speicher mehr zur Verfügung steht, soll 0 zurückgeliefert werden.

Lösung:

(b) 3 Punkte
Schreiben Sie eine Funktion *create_expr*, die die einen Operator erhält (Typ *Operator*) und zwei Zeiger auf die Operanden und einen entsprechenden Operator-Knoten dynamisch erzeugt und einen Zeiger darauf zurückliefert. Wenn kein Speicher mehr zur Verfügung steht, soll 0 zurückgeliefert werden.

Lösung:

(c) 4 Punkte
Schreiben Sie eine rekursive Funktion *eval_expr*, die einen Zeiger auf einen Ausdrucksbaum erhält und dessen Wert als *double* zurückliefert.

Lösung:

(d) 4 Punkte
Schreiben Sie eine rekursive Funktion *print_expr*, die einen Zeiger auf einen Ausdrucksbaum erhält und diesen auf der Standard-Ausgabe ausgibt. Sie sollten dabei die Ausgabe jeweils vollständig klammern, damit die Ausgabe unabhängig von den Vorrängen der Operatoren korrekt ist.

Lösung: