# C-Crashkurs

GLUG Herford
Frank W. Bergmann

# Geschichte

- 1972 entwickelt, um UNIX zu portieren
- Hardwarenah bis zum Setzen von Registerbits
- viele “Weiterentwicklungen”: C++, ObjectiveC, C#

# Hello World

```
#include <stdio.h>

int main()
{
    printf("hello, world\n");
    return 0;
}
```

# Integer

```
#include <stdio.h>
#include <stdint.h>
int main() {
  char c;
  int i=65;
  long l;
  uint64_t x;

  c = 'Z'; x = l = c; c = (char) i;
  printf(“c=%c l=%d\n”, c, (int) l);
  return 0;
}
```

# Char

```
#include <stdio.h>
int main(int argc, char **argv) {
  char a[] = “Alfred E. Neumann”;
  char nl[2];
  char *p;
  p = nl; *p++ = '\n'; *p = '\0';
  printf(“%c%c%c%s”, *a, a[1], *(a+2), nl);
  return 0;
}
```

# Variablen

- Variablentypen: int, char
- Auszeichner: short, long, unsigned
- portabel: z. B. uint32_t (stdint.h)
- Array name[6]
- Zeigerverarbeitung: * (“Inhalt von”), & (“Zeiger auf”)
- Strings: char[9] oder *char
- char c = 'A'; char s[] = “Hello”;
- struct node { int key; struct tnode *next; }

# Elemente der Sprache

Vergleiche:  ==  !=  <=  >=  <  >

Operationen:  !  &&  ||  &  |  +  -  *  /  ...

(|| = logisch OR, & = bitweises AND)

Zuweisungen:  =  +=  -=  ...

Funktionen:  int strcmp( const char *cs, const char *ct)

Bedingungen: if (...) { ... } else { ... }

Verzeigungen/Schleifen: while  do/while  for

# Schreibweisen

```
for (c = 1; c < 10; c++)
{
  printf(“%d\n”, c*c);
}


for (c = 1; c < 10;) printf(“%d\n”, c*c++);


c = 1;
for(;;) {
  if (c >= 10) break;
  printf(“%d\n”, c*c);
  ++c;
}
```

# Preprozessor/Compiler

Preprozessor:

#include <headerdatei.h>
enthält Prototypen:
int strcmp(const char *, const char *);

#define BEZEICHNUNG Ersatztext bla bla

Compiler:

sizeof(Object)

# Verschachtelung und goto

```
if (...) {
  if (...) {
    } else {
      goto error_out;
    }
  }
}
...
error_out:
```

# Compile

```
~ $ cat test-libcall.c
#include <stdio.h>
main() { printf("hello, world\n"); }
~ $ gcc -Os test-libcall.c;./sstrip a.out
~ $ ls -lGn a.out
-rwxrwxr-x. 1 500 1584 2010-01-17 21:11
a.out
~ $ file a.out
a.out: ELF 32-bit LSB executable, Intel 80386,
version 1 (GNU/Linux), dynamically linked
(uses shared libs), stripped
```

# static Compile

```
~ $ diet -Os gcc -Os test-libcall.c;./sstrip a.out
~ $ ls -lGn a.out
-rwxrwxr-x. 1 500 1163 2010-01-17 21:12
a.out
~ $ file a.out
a.out: ELF 32-bit LSB executable, Intel 80386,
version 1 (SYSV), statically linked, stripped
~ $
```

# Optimizations

```
~ $ cat test-syscall.c
#include <unistd.h>
#define HW "hello, world\n"
main() { write(1, HW, sizeof(HW)); }
~ $ diet -Os gcc -Os test-syscall.c;./sstrip
a.out
~ $ ls -lGn a.out
-rwxrwxr-x. 1 500 951 2010-01-17 21:22 a.out
~ $ file a.out
a.out: ELF 32-bit LSB executable, Intel 80386,
version 1 (SYSV), statically linked, stripped
```

Ende C-Crashkurs
Vorkurs für
“Systemadministration mit C”

Danke für's Dabeisein!

Noch Fragen?