The CKPMcc Project

Compiler Construction Course

Summer Term 2006

Clemens Krainer 9020112

Content

- Goals
- Input Language
- Output Language
- Virtual Machine
- Code Example
- Project Status
- Planned Tasks
- Questions & Answers

Goals

- Compiler should accept a subset of C
- Separate Pre-Processor module
- Separate Compiler module
- Separate Link Editor module
- Executable should run on a VM
- Self Compilation

Input Language

- A subset of ANSI-C, no new keywords
- Typing
  - Strong
  - Basic: char, int, bool
  - Composite: arrays, structs
  - Pointers and type casts
- Functions: local hiding
- Control: if, while, break, continue
- Nested Structures

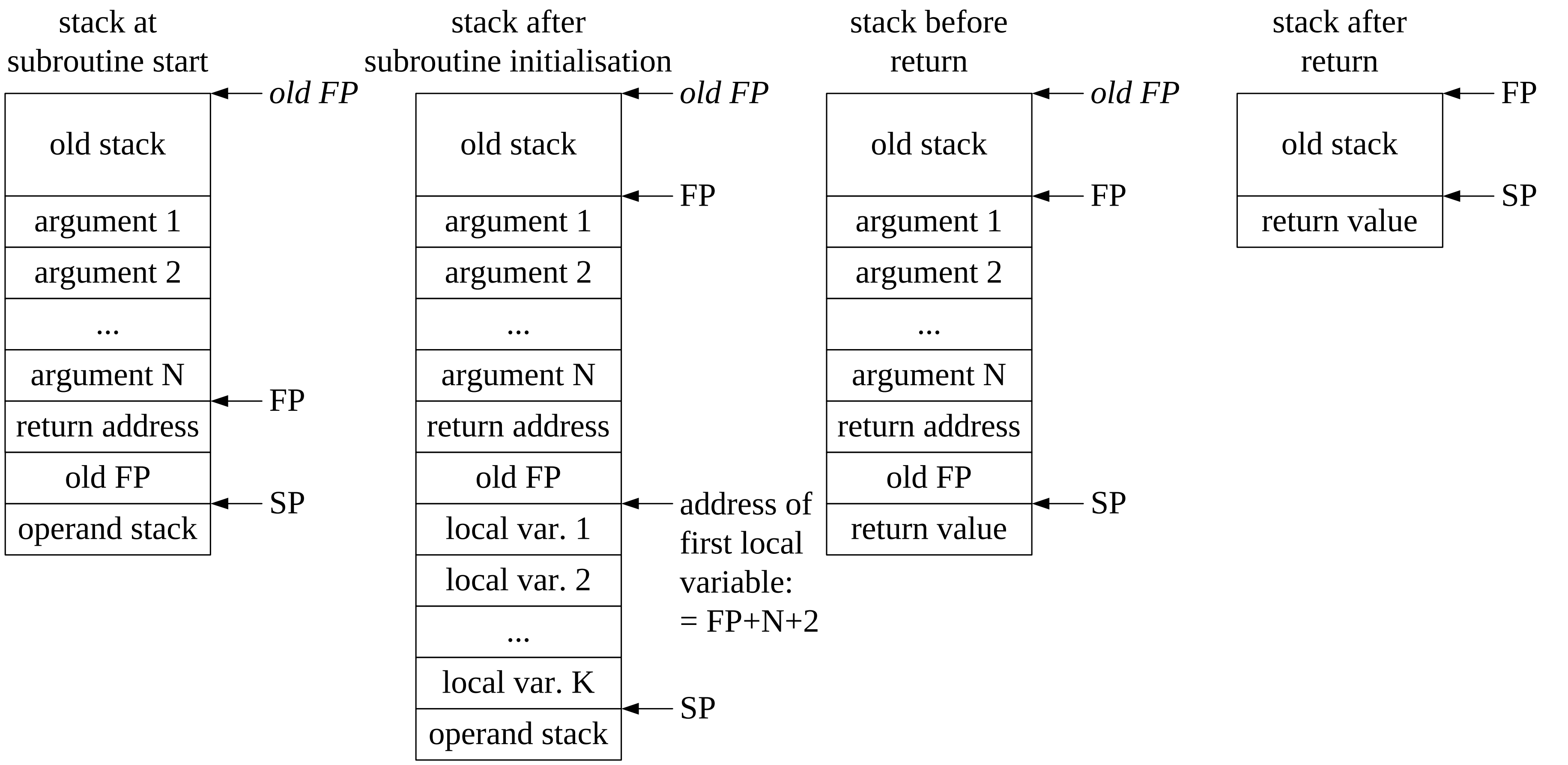
Output Language

- Self defined object file format
- Code Segment
  - machine code as byte sequence
- Data Segment
  - String literals
- Symbol Table
  - External and internal references
- String Table
  - Symbol Names

Virtual Machine

- Stack based
- Implemented in C
- Registers
  - PC: Program Counter
  - FP: Frame Pointer
  - SP: Stack Pointer
- Non standard instructions:
  - File I/O: open, close, read, write
  - Memory: malloc, free

VM Calling Conventions



Code Example

```
/* function prologue */
0x00000000:   dec_fp    #0x0008         int main (int argc, char** argv) {

0x00000003:   inc_sp    #0x0008             char *s;   int r;

/* function code */
0x00000006:   push_i    #0x00000000         s = "Hello World.\n";
0x0000000B:   st_i      #0x0010,fp

0x0000000E:   push_b    #0x01               r = write (1, (void*)s, 13);
0x00000010:   ld_i      #0x0010,fp
0x00000013:   push_b    #0x0d
0x00000015:   write
0x00000016:   st_i      #0x0014,fp

0x00000019:   push_b    #0x0f               return 15;
0x0000001B:   ldc_i_0
0x0000001C:   beq       #0x0000

/* function epilogue */
0x0000001F:   push_sp
0x00000020:   push_b    #0x08
0x00000022:   sub_i
0x00000023:   swap
0x00000024:   st_i_sp
0x00000025:   dec_sp    #0x0004
0x00000028:   ret_i                     }

/* data segment */
0x00000000:   48 65 6c 6c 6f 20 57 6f   72 6c 64 2e 0a 00        Hello World...
```

Code Example Execution

```
clem@nanook:~/Studium/Compiler-Systeme/work/trunk/vm/src> ./CKPMvm -v -v hello.o
Verbose mode, level is 2. VM version is 1
File name is 'hello.o'
Try to allocate 4 MB virtual machine memory.
argv[0]='hello.o'
Loading file 'hello.o'
Start address is 0x00000200.
0x00000200:   jsr      #0x00000000        /* runtime environment */
0x00000000:   dec_fp   #0x0008            int main (int argc, char** argv) {   /* prologue */
0x00000003:   inc_sp   #0x0008                char *s;   int r;
0x00000006:   push_i   #0x00000100            s = "Hello World.\n";
0x0000000B:   st_i     #0x0010,fp
0x0000000E:   push_b   #0x01                  r = write (1, (void*)s, 13);
0x00000010:   ld_i     #0x0010,fp
0x00000013:   push_b   #0x0d
0x00000015:   write
Hello World.
write string fd=1, written=13, length=13 string='Hello World.'
0x00000016:   st_i     #0x0014,fp
0x00000019:   push_b   #0x0f                  return 15;
0x0000001B:   ldc_i_0
0x0000001C:   beq      #0x0000
0x0000001F:   push_sp                         /* epilogue */
0x00000020:   push_b   #0x08
0x00000022:   sub_i
0x00000023:   swap
0x00000024:   st_i_sp
0x00000025:   dec_sp   #0x0004
0x00000028:   ret_i                       }
0x00000205:   halt                        /* runtime environment */
Execution terminated.
Virtual Machine Shutdown.
Virtual Machine Memory freed.
clem@nanook:~/Studium/Compiler-Systeme/work/trunk/vm/src>
```

Project Status

- Pre-Processor, Scanner & Parser:
  - Finished
- Code Generator:
  - Finished: if, while, break, continue, return, assignments, functions, type casts
  - Open: pointer operators, i.e.: *, &, ++, --
- Link Editor:
  - Open
- Virtual Machine:
  - Finished

Planned Tasks

- Operators: *, &, ++, --
- Separate compilation (Link Editor)
- Self compilation

Q & A