

预备工作 2：定义编译器功能 & 汇编语言编程

高文畅 1812983

摘 要

本次实验分析了 GCC 编译器支持的 C 语言特性,在此基础上定义了 C 语言的子集,之后用上下文无关文法对其做了简单的描述.之后以一个简单的 C 语言程序为例,该程序用来求解较小范围的整数 n 的阶乘,编写了与之等价的 x86 汇编程序.

关键词: 编译系统, C 语言, 上下文无关文法, 汇编语言

目录

1	编译器支持的 C 语言特性及 C 语言子集	3
1.1	支持的特性	3
1.1.1	数据类型	3
1.1.2	变量, 常量的声明和赋值	3
1.1.3	定义函数	3
1.1.4	注释	3
1.1.5	分支结构	3
1.1.6	循环结构	3
1.2	用 CFG 描述 C 语言子集	3
1.2.1	标识符的命名	3
1.2.2	变量声明	3
1.2.3	变量赋值	4
1.3	数组声明	4
1.4	函数定义	4
1.5	循环语句	4
1.6	条件语句	4
2	汇编程序	5
2.1	原 C 语言程序	5
2.2	汇编语言程序	5
2.3	用 gcc 生成的汇编语言程序	6

1 编译器支持的 C 语言特性及 C 语言子集

1.1 支持的特性

1.1.1 数据类型

基本数据类型: short, int, long, char, float, double, bool

指针 (如 int* 等), 结构体 (struct), 枚举 enum, 联合 union

1.1.2 变量, 常量的声明和赋值

1.1.3 定义函数

1.1.4 注释

1.1.5 分支结构

分支结构包括关键字: if, else, switch, case, default

1.1.6 循环结构

循环结构包括关键字: for, while, do

1.2 用 CFG 描述 C 语言子集

1.2.1 标识符的命名

$$letter \rightarrow a \mid b \mid \dots \mid z \mid A \mid B \mid \dots \mid Z$$
$$digit \rightarrow 0 \mid 1 \mid \dots \mid 9$$
$$begin \rightarrow _ \mid \epsilon$$
$$body \rightarrow body \ letter \mid body \ digit \mid letter \mid digit \mid _ \mid \epsilon$$
$$id \rightarrow begin letter body$$

1.2.2 变量声明

$$type \rightarrow *type \mid short \mid int \mid long \mid char \mid float \mid double \mid bool$$
$$ids \rightarrow ids, \ id \mid id$$
$$var_decl \rightarrow type \ ids$$

1.2.3 变量赋值

$$\begin{aligned} integer &\rightarrow integer\ digit \mid digit \\ float &\rightarrow integer.integer \mid integer \\ boolean &\rightarrow true \mid false \\ value &\rightarrow integer \mid integer \mid float \mid boolean \\ var_assi &\rightarrow id = integer \mid float \mid boolean \end{aligned}$$

1.3 数组声明

$$\begin{aligned} decl &\rightarrow type\ arrays \\ arrays &\rightarrow arrays, id[integer] \mid id[integer] \end{aligned}$$

1.4 函数定义

$$\begin{aligned} ftype &\rightarrow type \mid void \\ fun_defn &\rightarrow ftype\ funname(paras)\{stmt\} \\ paras &\rightarrow paras, type\ id \mid type\ id \mid \epsilon \\ funname &\rightarrow id \end{aligned}$$

1.5 循环语句

$$\begin{aligned} stmt &\rightarrow while(expr)\ stmt \\ stmt &\rightarrow for(expr; expr; expr) stmt \\ stmt &\rightarrow do\ stmt while(expr) \end{aligned}$$

1.6 条件语句

$$\begin{aligned} stmt &\rightarrow if(stmt) else\ stmt \\ stmt &\rightarrow if(stmt)\ stmt \end{aligned}$$

2 汇编程序

2.1 原 C 语言程序

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#define INF 9999
int x =101;
int fact(int a)
{
    int i=1,ans=1;
    for(;i<=a;i++)
        ans*=i;
    return ans;
}
int main()
{
    int n;
    scanf("%d",&n);
    printf("%d\n",fact(n));
    printf("%d",x);
    return 0;
}
```

2.2 汇编语言程序

```
.text
.global max
.type fact, @function
max:
    movl    8(%ebp), %edx
    movl    $1, %eax
    cmpl    $1, %edx
    jle     L1

L0:
    imull   %edx, %eax
    subl    $1, %edx
    cmpl    $1, %edx
    jg      .L0

L1:        done:
x:
```

```

.zero 4
.align 4

.section      .rodata
STR0:
.string "%d"
STR1:
.string "%d\n"
STR2:
.string "%d"

.text
.global      main
.type main, @function
main:
pushl    $n
pushl    $STR0
call     scanf
addl     $12, esp
movl     n, %rbp+8
pushl    %eax
pushl    $STR1
call     printf
movl     x, %eax
pushl    %eax
pushl    $STR2
call     printf
addl     $8, %esp
movl     $0, %eax
ret

.section      .note.GNU-stack,"",@progbits

```

2.3 用 gcc 生成的汇编语言程序

```

.file "E2.c"
.text
.globl  x
.data
.align 4
x:

```

```

.long 100
.text
.globl max
.def max; .scl 2; .type 32; .endef
.seh_proc max
max:
pushq %rbp
.seh_pushreg %rbp
movq %rsp, %rbp
.seh_setframe %rbp, 0
subq $16, %rsp
.seh_stackalloc 16

.seh_endprologue
movq %rcx, 16(%rbp)
movl %edx, 24(%rbp)
movl $0, -4(%rbp)
movl $-9999, -8(%rbp)
jmp .L2
.L4:
movl -4(%rbp), %eax
cltq
leaq 0(,%rax,4), %rdx
movq 16(%rbp), %rax
addq %rdx, %rax
movl (%rax), %eax
cmpl %eax, -8(%rbp)
jge .L3
movl -4(%rbp), %eax
cltq
leaq 0(,%rax,4), %rdx
movq 16(%rbp), %rax
addq %rdx, %rax
movl (%rax), %eax
movl %eax, -8(%rbp)
.L3:
addl $1, -4(%rbp)
.L2:
movl -4(%rbp), %eax
cmpl 24(%rbp), %eax
jl .L4
movl -8(%rbp), %eax
addq $16, %rsp
popq %rbp

```

```

ret
.seh_endproc
.def __main; .scl 2; .type 32; .endef
.section .rdata,"dr"
.LC0:
.ascii "%d\0"
.text
.globl main
.def main; .scl 2; .type 32; .endef
.seh_proc main
main:
pushq %rbp
.seh_pushreg %rbp
movq %rsp, %rbp
.seh_setframe %rbp, 0
subq $96, %rsp
.seh_stackalloc 96
.seh_endprologue
call __main
movl $0, -4(%rbp)
leaq -52(%rbp), %rax
movq %rax, %rdx
leaq .LC0(%rip), %rcx
call scanf
jmp .L7
.L8:
movl -52(%rbp), %eax
imull -4(%rbp), %eax
movl %eax, %edx
movl -4(%rbp), %eax
cltq
movl %edx, -48(%rbp,%rax,4)
addl $1, -4(%rbp)
.L7:
cmpl $9, -4(%rbp)
jle .L8
leaq -48(%rbp), %rax
movl $10, %edx
movq %rax, %rcx
call max
movl %eax, -8(%rbp)
movl -8(%rbp), %eax
movl %eax, %edx
leaq .LC0(%rip), %rcx

```



```
call    printf
movl    $0, %eax
addq    $96, %rsp
popq    %rbp
ret
.seh_endproc
.ident   "GCC: (x86_64-posix-seh-rev0, Built by MinGW-W64 project) 7.3.0"
.def     scanf; .scl 2; .type 32; .endef
.def     printf; .scl 2; .type 32; .endef
+
```