

任务











Вариант N 10

```
EX10: PROC OPTIONS ( MAIN );
DCL A DEC FIXED INIT ( 10 );
      DCL B BIT ( 16 );
      B = A;
END EX10;
```

将这个程序编译成汇编语言（IBM 370机器）。

工作目录及工作文件描述

工作目录：LinuxLabRab.tgz 解压出来的文件
工作目录介绍：

	Имя	Размер	Тип	Последнее изменение
1	 R_e_a_d_m_y	612 байт	Текст	апр. 6 2007
2	 Makefile	726 байт	Текст	февр. 13 2021
3	 komppl.c	70,5 кБ	Текст	янв. 11 2005
4	 kompassr.c	56,6 кБ	Текст	марта 7 2007
5	 examppl.pli	560 байт	Текст	янв. 11 2005
6	 absloadm.c	25,4 кБ	Текст	апр. 6 2007
7	 StartTestTask	689 байт	Программа	февр. 14 2021
8	 GenSysProg	206 байт	Программа	марта 9 2007
9	 ChangeCodTable	444 байта	Программа	февр. 13 2021
10	 spis.mod	11 байт	Звук	марта 7 2007

1. R_e_a_d_m_y

格式：txt
内容：
09.03.2007

1. Данный комплект, состоящий из трех элементов учебной системы программирования и тестового задания:

 - **компилятор** PL/1 --> Ассемблер (komppl.c),
 - **компилятор** Ассемблер --> объектный образ (kompassr.c),
 - **абсолютный загрузчик, эмулятор и отладчик** (absloadm.c),
 - **тестовое задание** (examppl.pli и spismod).

2. Для получения исполняемых модулей двух компиляторов и абсолютного загрузчика следует выполнить Bash-скрипт GenSysProg.

3. После получения исполняемого кода компиляторов можно запускать на выполнение тестовый пример с помощью Bash-скрипта StartTestTask.

作用：讲述了完成实验的顺序

2. Makefile

内容:

```
# Script for Programming system generation
all: komppl.exe kompassr.exe absloadm.exe
#komppl.o: komppl.c
# gcc -o komppl.o komppl.c
komppl.exe: komppl.c
    @echo "_____komppl.exe generation_____"
    gcc -o komppl.exe komppl.c
    @echo "_____"
kompassr.exe: kompassr.c
    @echo "_____kompassr.exe generation_____"
    gcc -o kompassr.exe kompassr.c
    @echo "_____"
absloadm.exe: absloadm.c
    @echo "_____absloadm.exe generation_____"
    gcc -o absloadm.exe absloadm.c -Incurses
    @echo "_____"
```

使用gcc编译器编译3个c程序: komppl.c kompassr.c absloadm.c

3. komppl.c

【重点文件】

这个c程序实现了PLI语言的简易编译器。该实验的任务就是完善这个文件，使编译器可以把源代码编译成正确的汇编代码。

4. kompassr.c

应该是暂时用不到

5. examppl.pli

【重要文件】

输入的源高级语言文件（PL/1语言），

6. absloadm.c

应该是暂时用不到

7. StartTestTask

脚本

8. GenSysProg

脚本，实际上执行的是make

9. ChangeCodTable

脚本，将文本的ROI-8编码转化成UTF-8编码

10. spis.mod

未知

komppl.c 变量列表和函数列表

常量列表

常量名称	常量描述
MAXNISXTXT	定义源代码最大行数，用于声明源代码存储数组的最大长度
NSINT	定义语法分析中用到的语法规则的最大数量
NCEL	定义目标栈的最大深度
NDST	定义达成目标的栈的最大深度
NVXOD	定义识别的不同的输入符号的最大数量
NSTROKA	定义存储压缩后源代码行的字符数组的最大长度
NNETRM	定义非终结符的最大数量
MAXLTXT	定义存储汇编语言输出文本行的最大数量
MAXFORMT	定义用于格式化源代码片段的数组的最大长度
NSYM	定义符号表中符号的最大数量

变量列表

变量名称	类型	描述
NISXTXT	int	数组长度
ISXTXT	char[MAXNISXTXT][80]	存储源代码的二维字符数组
I1, I2, I3, I4	int	循环计数器
PREDSYM	char	最后处理的字符
STROKA	char[NSTROKA]	存储压缩后的源代码
I, J, K, L	int	当前索引，分别用于 压缩文本、语法规则表、目标栈和目标栈
ASS_CARD	union	用于生成输出文件的记录模板
ASSTXT	char[MAXLTXT][80]	存储汇编语言输出文本的数组
IASSTXT	int	输出数组的索引
FORMT	char[MAXFORMT][9]	用于格式化源代码片段的数组
IFORMT	int	格式化数组的索引

函数列表及其功能描述

函数头	功能描述
void compress_ISXTXT()	压缩源代码文本，移除多余空格
void build_TPR()	根据邻接矩阵构建转移矩阵
void mcel(char* T1, int T2, int T3)	向目标栈添加新的元素
void mdst(char* T1, int T2, int T3, int T4, int T5)	向达成目标的栈添加新的元素
int numb(char* T1, int T2)	计算字符串在VXOD表中的索引编号
int sint_ANAL()	实现语法分析，构建语法分析树
long int VALUE(char* s)	将二进制常量字符串转换成long int类型
void FORM()	格式化指定的源代码片段
void ZKARD()	将生成的汇编语言指令添加到输出数组中
int AVI1()	第一遍语义分析中处理"AVI"（算术表达式）
int BUK1()	第一遍语义分析中处理字母（字符）
int CIF1()	第一遍语义分析中处理数字
int IDE1()	第一遍语义分析中处理标识符
int IPE1()	第一遍语义分析中处理变量名

int IPR1()	第一遍语义分析中处理程序名
int LIT1()	第一遍语义分析中处理字面量（常量）
int MAN1()	第一遍语义分析中处理浮点数的尾数部分
int ODC1()	第一遍语义分析中处理PL/1的DCL声明
int OEN1()	第一遍语义分析中处理PL/1的END结束声明
int OPA1()	第一
int OPR1()	第一遍语义分析中处理PL/1的PROC过程声明
int PRO1()	第一遍语义分析中处理整个PL/1程序
int RZR1()	第一遍语义分析中处理变量的精度声明
int TEL1()	第一遍语义分析中处理PL/1程序的主体部分
int ZNK1()	第一遍语义分析中处理算术操作符

实验步骤

- 1. 修改词法分析器的数据库
- 2. 修改语义分析器的数据库
 - 2.1 修改SINT [NSINT]
 - 2.2 修改VXOD [NVXOD]
 - 2.3 修改char TPR [NVXOD] [NNETRM]
 - 2.4 修改函数： : DCF1, DCF2, ODC1, AVI2, OEN2, OPA2, OPR2, RZR2
 - 2.4.1 DCF1
 - 2.4.2 ODC1
 - 2.4.3 AVI2
 - 2.4.4 OEN2
 - 2.4.5 OPA2
 - 2.4.6 OPR2
 - 2.4.7 RZR2
 - 2.4.8 DCF2
- 3. 运行程序，获取结果