

NAME

p236 - generate code for IBM System/360 programs

SYNOPSIS

p236 -Abss catg# ck dcatg# dm# do# dseq# err +list m# o# p s x#A (file)

FUNCTION

p236 is the code generating pass of the C compiler. It accepts a sequential file of flow graphs and parse trees from p1 and outputs a sequential file of assembly language statements suitable for input to the IBM System/360 assembler.

As much as possible, the compiler generates free-standing code, but, for those operations which cannot be done compactly, it generates inline calls to a set of machine-dependent runtime library routines. The names of the routines in the Machine Interface Library are listed in Section IV of the Interface Manual for your machine.

p236 accepts the following flags:

- bss inhibit generating code by a DS (define space) directive. By default, data initialized to zero is defined by a DS directive.
- catg# give the resulting object module category number # by including a HEADERS CATG=# statement in the output assembler file.
- ck enable stack overflow checking.
- dcatg# give the resulting data object module category number # by including a HEADERS CATG=# statement in the data output assembler file. This flag is used only with the -s flag.
- dm# set data object output module name to # when generating the assembler code by a START assembly directive. This flag is used only with the -s flag. The default module name is "DS(seqn)" where (seqn) is a hexadecimal sequence number generated from the sequence file (see -dseq flag).
- do# write the data output to the file #. This flag is used only with the -s flag. The default data file name is "dat(seqn).asm" where (seqn) is a hexadecimal sequence number generated from the sequence file (see -dseq flag).
- dseq# use the sequence file # to create unique file and label names for data files. The sequence file is created and initialized to zero if it does not exist. The file contains a short (16 bits) value that is incremented by one for each data file created. This flag is used only with the -s flag.
- err pass error diagnostics from input file to output file, add error diagnostics generated to output file, and report success. Only fatal errors (such as not being able to write an output file due to lack of space) will be reported and cause the error status to change to failure.

- +list** generate line number directives interspersed with assembly code to be used by the **lm** utility to provide listing file. The default is to turn listings off.
- o*** write the output to the file ***** and write error messages to STDOUT. Default is STDOUT for output and STDERR for error messages.
- m*** set object output module name to ***** when generating the assembler code by a **START** assembly directive. The default module name is the name of the first non local function. If there is no functions in the source file, and the **-s** flag is not used, the module will get the name of the first data variable which must not be local.
- p** omit profiler calls on entry to each function. Currently this can be used only with the IDRIS profiler.
- s** create a separate assembler file containing all data variables.
- x#** map the three virtual sections, for Functions (04), Constant Data (02), and Variables (01), to the two physical sections, Code (bit is a one) and Data (bit is a zero). Thus, **-x4** is for separate text and data segments, which is the default. This has no effect when used with the IDM assembler as it does not maintain separate text and data sections, except when the **-s** flag is given.

If **<file>** is present, it is used as the input file instead of the default STDIN. On many systems (other than IDRIS/UNIX), specifying a **<file>** is mandatory, because STDIN is interpreted as a text file, and hence becomes corrupted.

Files output from **p1** for use with the IBM System/36 code generator must be generated with the following options: no **-l** flag, since pointers are short; use of **-b0** for compact data, **-r0** since there are no register variables and no **-a** flag, since registers are interchangeable. **p1** should also be run with the flags **-cn6** to check externals properly.

Wherever possible, labels in the emitted code each contain a comment which gives the source line to which the code immediately following pertains, along with a running count of the number of bytes of code produced for a given function body.

RETURNS

p236 returns success if it produces no diagnostics, or if the **-err** flag is specified.

EXAMPLE

p236 usually follows **pp** and **p1**, as follows:

```
pp -o temp1 -x file.c
p1 -o temp2 -cmu -b0 -n6 -r0 temp1
p236 -o file.s temp2
```

or if separate text and data files are to be generated

```
pp -o temp1 -x file.s
pi -o temp2 -cmu -b0 -n6 -r0 temp1
p236 -catg20 dcatg0 -s -o file.s temp2
```

SEE ALSO

pi

NOTES

Stack overflow checking is only approximate, since a calculation of the exact stack high water mark is not attempted.