

Project: Process labels. Find integer values for all labels. Labels that are referenced in jump-instructions and **cup**-instructions are symbols for instruction addresses. Labels that are referenced in **ent**-instructions are integer values that reserve storage on the stack for local variables and computations. Make a label table.

Print instructions in the trace file having the suffix “.atrc”

Print the label table together with constants and instructions in the listing file, “.alst”

Program Files:	File	Description
	makepasm	File makepasm contains instructions for program pasm . Instructions are written for the <i>Unix</i> utility make . Program pasm is contained in file pasm .
	mkpasm	File mkpasm is a Unix script file that removes old file created in the last creation of executable file pas and invokes file makepasm to create a new executable file pasm . File makepasm is given below. rm *.o rm pasmlex.cpp rm pasm make -f makepasm

Command Line: Project **p04.5** can be invoked with zero or one program parameters. The first program parameter is the input file name. Sample command lines together with corresponding actions by program **pasm** are shown below. Boldfaced type indicates data entered at the keyboard by the user.

\$ **pasm**

Enter the input file name: **p00.pasm**

\$ **pasm p00.pasm**

Input File: The input file contains a P-Code Assembler program. The input file name must have the suffix “.pcd” Please refer to figure 1 for an example of the format of an input file.

Output Files: Two output files are produced. Both output files have the same prefix as the input **source.pcd** and the suffix **.pcd** is replaced by the suffix **.atrc** in the trace file and **.alst** in the assembler listing file. For example, if the input file was named **hello.pcd**, output files would be named **hello.atrc** and **hello.alst**.

\$ **pasm p00.pcd**

File **p00.atrc** is produced as shown in Figure 2.

File **p00.alst** is produced as shown in Figure 3.

L00001	ent	sp	L00002
	ent	ep	L00003

```

rtn    p
#define L00002 4
#define L00003 4
mst    0
cup    0    L00001
stp

```

Figure 1. Example input file p00.pcd.

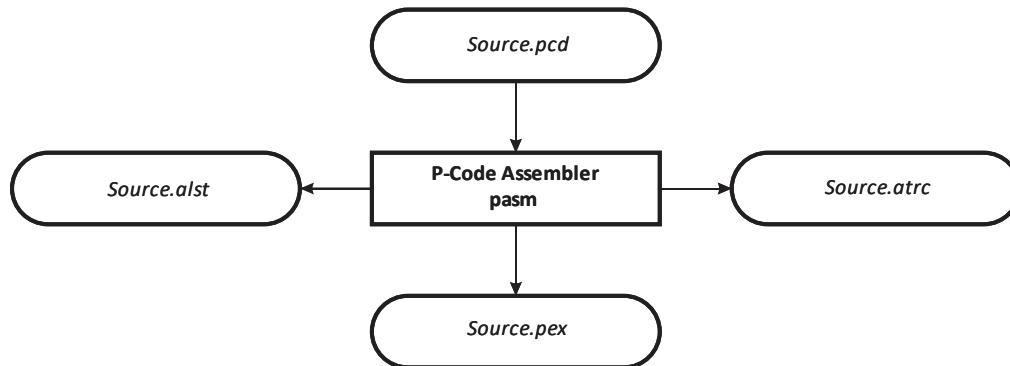


Figure 2. P-Code Assembler Block Diagram

```

Token=(258,LABEL,L00001)
#009 label-list->LABEL
Instruction Counter = 0
Token=(268,ENT_0,ent)
Token=(338,SP_R,sp)
#082 register->SP
Token=(258,LABEL,L00002)
060 class2-operation->ent register INTLITent(02) sp(00) 0(0000)
014 operation->class3-operation
#007 statement->label-list operation
Instruction Counter = 1
#004 item->statement
#002 list->item
Token=(268,ENT_0,ent)
Token=(339,EP_R,ep)
#083 register->EP
Token=(258,LABEL,L00003)
060 class2-operation->ent register INTLITent(02) ep(01) 0(0000)
014 operation->class3-operation
#008 statement->operation
Instruction Counter = 2
#004 item->statement
#003 list->list item

```

Figure 3. File p00.atrc

```
Token=(270,RTN_0,rtn)
Token=(335,P_T,p)
#073 type->P
042 class1-operation->rtn type
rtn(04)  p(07)      0(0000)
012 operation->class1-operation
#008 statement->operation
Instruction Counter = 3
#004 item->statement
#003 list->list item

Token=(259,DEFINE,#define)
Token=(258,LABEL,L00002)
Token=(260,INTLIT,4)
#006 definition->DEFINE LABEL INTLIT
Label="L00002"
INTLIT= 4
#005 item->definition
#003 list->list item

Token=(259,DEFINE,#define)
Token=(258,LABEL,L00003)
Token=(260,INTLIT,4)
#006 definition->DEFINE LABEL INTLIT
Label="L00003"
INTLIT= 4
#005 item->definition
#003 list->list item

Token=(269,MST_0,mst)
Token=(260,INTLIT,0)
041 class1-operation->mst intlrit
mst(03)  0(00)      0(0000)
012 operation->class1-operation
#008 statement->operation
Instruction Counter = 4
#004 item->statement
#003 list->list item

Token=(266,CUP_0,cup)
Token=(260,INTLIT,0)
Token=(258,LABEL,L00001)
059 class2-operation->cup INTLIT LABEL
014 operation->class3-operation
#008 statement->operation
Instruction Counter = 5
#004 item->statement
#003 list->list item
```

Figure 3. File p00.atrc (continued)

```
Token=(314,STP_0,stp)
040 class0-operation->stp
stp(30) 0(00) 0(0000)
011 operation->class0-operation
#008 statement->operation
Instruction Counter = 6
#004 item->statement
#003 list->list item

#001 program->list
```

Figure 3. File p00.atrc (continued)

String Constants
Index Constant

Set Constants
Index Constant

Integer Constants
Index Constant

Real Constants
Index Constant

P-Code Instruction Array

PC	OP	R1	R2
0	ent(02)	sp(00)	4(0004)
1	ent(02)	ep(01)	4(0004)
2	rtn(04)	p(07)	0(0000)
3	mst(03)	0(00)	0(0000)
4	cup(00)	0(00)	0(0000)
5	stp(30)	0(00)	0(0000)

Label Table

Label	Address	Resolved	Reference List
L00001	0	true	4
L00002	4	true	0
L00003	4	true	1

Figure 4. File p00.alst