รายการอ้างอิง

ภาษาไทย

มงคล อัศวโกวิทกรณ์. ระบบปฏิบัติการ. พิมพ์ครั้งที่ 2. กรุงเทพมหานคร : สถาบันคอมพิวเตอร์ ไอ ไอ ซี, 2537.

ยรรยง เต็งอำนวย. ระบบปฏิบัติการ. กรุงเทพมหานคร:บริษัทซีเอ็ดยูเคชั่น. (ม.ป.ป.)

อิว ไอยราคาญจนกุล. การเขียนคอมไพเลอร์สำหรับ IBM PC. กรุงเทพมหานคร:บริษัทซีเอ็ดยูเคชั่น. (ม.ป.ป.)

ภาษาอังกฤษ

Aho,A.V., Sethi,R ,and Ullman,J.D. Compilers, principles,techniques, and tools. Massachusette: Addison Wesley , 1988.

Andrews, G.R., Concurrent programming : principles and practice. California : The Bejamin/Cummings Publishing Co. , 1991.

Bacon, J. Concurrent systems: An integrated approach to operating systems,database, and distributed systems. Massachusette: Addison-Weslay, 1993.

Gosling, J. , Mcgilton,H. Java language enviroment whitepaper. http://java.sun.com/doc/language-environment/ ,1996.

Hollen, I.A, Compiler design in C. New Jersey : Prentice-Hall International , 1990.

Mak,R. Writing compilers and interpreters an applied approach. New York:John Wiley & Sons inc. , 1991.

Pittman,T., Peters J. The art of compiler design : Theory and practice. New jersey : Prentice-Hall Inc , 1992.

Siberschatz,A., Galvin,P.B. Operating System Concepts Massachusette: Addision-Wesley , 1994.

Tribble,G. Java computing whitepapers. http://www.sun.com/javacomputing/whpaper/ , 1996

ภาคผนวก ก

ตารางแสดงรหัสกลาง

รหัสกลางที่ได้จากขั้นตอนคอมไพเลอร์จะเป็นรหัสไบต์และตัวถูกดำเนินการ (operand) เรียงกันตามลำดับ โดยรหัสไบต์แต่ละคำสั่งแทนรหัสกลางดังนี้

รหัสไบต์	รหัสกลาง	รหัสไบต์	รหัสกลาง
1	Literal	22	Equal
2	LvalueGV	23	Not
3	RvalueGV	24	And
4	LValueLV	25	Or
5	RValueLV	26	PID
6	FetchLV	27	FetchGV
7	JMP	28	Send
8	JZ	29	Receive
9	Call	30	Return0
10	Index	31	Return1
11	Set	32	Print
12	SetGV	33	PrintCh
13	Plus	34	Random
14	Minus	35	Halt
15	Multiply	36	Wait
16	Div	37	Signal
17	LT	38	Func
18	LE	39	Proc
19	GE	40	NOP
20	GT	41	EOF
21	NE		

ภาคผนวก ข

โปรแกรมปัญหาการขนส่ง

1.โปรแกรมต้นฉบับ

```
// Declaration Data
GLOBAL R[30],Link[100],IsReport,  RealTarget[3],RealSource[3];
GLOBAL Visit[30], located[3], RobotPID[3] ,Target[3];
GLOBAL Map[30], RobotStatus[3], CurrentPos[3];
GLOBAL sem;
Function PutR(Node,LinkNo,NodeLink) {
  R[Node*3+LinkNo-3] = NodeLink;        // Index 2 dimension
}

Function GetR(Node,LinkNo) {
  Return( R[Node*3+LinkNo-3] );
}

function Report()
{
 while (1) {
  if (IsReport)
  {
   wait(sem);;
   i = 1;
   while (i <> 4)
   {
    status = RobotStatus[i];
    print('      R',i,':',CurrentPos[i],'-> ');
    if (status>=2)
      Print(Target[I])
    Else
      Print('  ');

    IF     (status==0) printLn('No job')
    ELSE IF (status==1) printLn('Think ')
    ELSE IF (Status==2) printLn('Move  ')
    ELSE IF (status==3) printLn('Wait  ');
    i=i+1;
   }
   IsReport = 0;
```

```
    signal(sem);;
   }
  }
}

FUNCTION INIT() {
  PutR(1,1, 2); PutR(1,2, 4); PutR(1,3, 5);
  PutR(2,1, 1); PutR(2,2, 0); PutR(2,3, 0);
  PutR(3,1, 6); PutR(3,2, 7); PutR(3,3, 0);
  PutR(4,1, 1); PutR(4,2, 8); PutR(4,3, 0);
  PutR(5,1, 1); PutR(5,2, 6); PutR(5,3, 9);
  PutR(6,1, 3); PutR(6,2,5); PutR(6,3, 10);
  PutR(7,1, 3); PutR(7,2,10); PutR(7,3, 0);
  PutR(8,1, 4); PutR(8,2, 0); PutR(8,3, 0);
  PutR(9,1,10); PutR(9,2,5); PutR(9,3, 0);
  PutR(10,1,6); PutR(10,2,7); PutR(10,3,9);
  RobotPID[1] = GetPID(Robot1);
  RobotPID[2] = GetPID(Robot2);
  RobotPID[3] = GetPID(Robot3);
  I = 3;
  While (I) {
    CurrentPos[I] = 1; RobotStatus[i] = 0;
    I=I-1
  }
  I = 100;
  While (I) {
   Link[i] = 0; I=I-1 }
  JobCount = 0;
}

function GetLinkStatus(NodeA,NodeB) {
  IF (NodeA < NodeB)
   Return( Link[NodeB*10-10+NodeA] )
  ELSE
   Return( Link[NodeA*10-10+NodeB] );
}

function SetLinkStatus(NodeA, NodeB, Status) {
 IF (NodeA < NodeB)
  Link[NodeB*10-10+NodeA] = Status
 Else
  Link[NodeA*10-10+NodeB] = Status;
}

function Move(RobotNo, From, To)
{
 IsReport = 1;
 while(From <> To)
 {

  wait(sem);
   if ( GetLinkStatus(From,To) == 0 )
   {
    SetLinkStatus(From,To, RobotNo);
    PrintLn('Robot number ',RobotNo,' : ',From,' -> ',to,' start move');
    RobotStatus[RobotNo]=2;
```

```
    }
    else
    {
     PrintLn('Robot number ',RobotNo,' : ',From,' -> ',to,' wait');
     RobotStatus[RobotNo]=3;
    }
   signal(sem);;

   if ( GetLinkStatus(From,To) == RobotNo )
   {
    W=10;   while (W) W=W-1;
    wait(sem);;
     PrintLn('Robot number ',RobotNo,' : ',From,' -> ',to,' finish move');
     SetLinkStatus(From ,To , 0);
     RobotStatus[RobotNo]=1;
    signal(sem);;
    CurrentPos[RobotNo] = To;
    From = To;
   }
 }}

Function SearchAndMove(RobotNo, From , To)
{
 Visit[RobotNo*10-10 +To] = To;
 IF (CurrentPos[RobotNo]==RealTarget[RobotNo])
  RobotStatus[RobotNo] = 0;
 IF (From==To)  Located[RobotNo] = 1;
 I = 1;
 While (I <= 3)
 {
  NextNode = GetR(To,I);
  IF ((NextNode>0) & !Visit[RobotNo*10-10+NextNode] & !Located[RobotNo])
   SearchAndMove(RobotNo,From,NextNode);
  I=I+1;
 }
 Current = CurrentPos[RobotNo];
 IF (Located[RobotNo])
 {
  Target[RobotNo] = To;
  Move(RobotNo,Current,To);
 }

 IF (CurrentPos[RobotNo]==RealTarget[RobotNo])
  RobotStatus[RobotNo] = 0;
 Visit[RobotNo*10-10 +To] = 0;
}

Function RobotProgram(RobotNo)
{
 while (1) {
  To   = Receive(GetPid(GenJob));
  RealTarget[RobotNo] = To;
  RobotStatus[RobotNo] = 1;
  Located[RobotNo] =0; SearchAndMove(RobotNo,CurrentPos[RobotNo],To);
  RobotStatus[RobotNo] = 0;
```

```
   IsReport = 1;
   IF (CurrentPos[RobotNo] == From)
   {
        print('robot number ',robotno,' cannot go');

   }
  }

}

process Robot1() {
  RobotProgram( GetPid(Robot1) );
}

process Robot2() {
  RobotProgram( GetPid(Robot2) );
}

process Robot3() {
  RobotProgram( GetPid(Robot3) );
}

process GenJob(JobCount) {
  While (JobCount)
  {
    RobotNo = Random(3)+1;
    To   = random(10)+1;

   if ((!RobotStatus[RobotNo]) && (currentPos[RobotNo] <>To))
    {
     wait(sem);;
       PrintLn('Generate job for robot number ',RobotNo,' move to ',To);
     signal(sem);;
     send(RobotPID[RobotNo], To);
     JobCount = JobCount-1;
    }

  }
  Halt();
}

process Rr() {report();}
Run
{
  IsReport=0;
  init();
  sem = 1;
  GenJob(80);
  Robot1();
  Robot2();
  Robot3();
 // Rr();
}
```

~

2.รหัสกลางที่ได้

```
  1:call 2443
  4:func 3 0
  9:lvalueGV 1 rvalueLV 1 literal 3 multiply rvalueLV 2 plus literal 3 minus index rvalueLV 3
    setGV
 32:ret0
 33:func 2 0
 38:lvalueGV 1 rvalueLV 1 literal 3 multiply rvalueLV 2 plus literal 3 minus index fetchGV ret1
 59:ret0
 60:func 0 2
 65:literal 1 jz 410
 71:rvalueGV 133 jz 407
 77:wait 224
 80:lvalueLV 1 literal 1 set
 87:rvalueLV 1 literal 4 ne
 94:jz 394
 97:lvalueLV 2 lvalueGV 216 rvalueLV 1 index fetchGV set
109:literal 32 printCh
113:literal 32 printCh
117:literal 32 printCh
121:literal 32 printCh
125:literal 32 printCh
129:literal 32 printCh
133:literal 32 printCh
137:literal 82 printCh
141:rvalueLV 1 print
145:literal 58 printCh
149:lvalueGV 220 rvalueLV 1 index fetchGV print
158:literal 45 printCh
162:literal 62 printCh
166:literal 32 printCh
170:rvalueLV 2 literal 2 ge
177:jz 192
180:lvalueGV 181 rvalueLV 1 index fetchGV print
189:jmp 200
192:literal 32 printCh
196:literal 32 printCh
200:rvalueLV 2 literal 0 equal
207:jz 245
210:literal 78 printCh
214:literal 111 printCh
218:literal 32 printCh
222:literal 106 printCh
226:literal 111 printCh
230:literal 98 printCh
234:literal 13 printCh
238:literal 10 printCh
242:jmp 380
245:rvalueLV 2 literal 1 equal
252:jz 290
255:literal 84 printCh
259:literal 104 printCh
```

```
263:literal 105 printCh
267:literal 110 printCh
271:literal 107 printCh
275:literal 32 printCh
279:literal 13 printCh
283:literal 10 printCh
287:jmp 380
290:rvalueLV 2 literal 2 equal
297:jz 335
300:literal 77 printCh
304:literal 111 printCh
308:literal 118 printCh
312:literal 101 printCh
316:literal 32 printCh
320:literal 32 printCh
324:literal 13 printCh
328:literal 10 printCh
332:jmp 380
335:rvalueLV 2 literal 3 equal
342:jz 380
345:literal 87 printCh
349:literal 97 printCh
353:literal 105 printCh
357:literal 116 printCh
361:literal 32 printCh
365:literal 32 printCh
369:literal 13 printCh
373:literal 10 printCh
377:jmp 380
380:lvalueLV 1 rvalueLV 1 literal 1 plus set
391:jmp 87
394:lvalueGV 133 literal 0 setGV
401:signal 224
404:jmp 407
407:jmp 65
410:ret0
411:func 0 2
416:literal 1 literal 1 literal 2 call 4
428:literal 1 literal 2 literal 4 call 4
440:literal 1 literal 3 literal 5 call 4
452:literal 2 literal 1 literal 1 call 4
464:literal 2 literal 2 literal 0 call 4
476:literal 2 literal 3 literal 0 call 4
488:literal 3 literal 1 literal 6 call 4
500:literal 3 literal 2 literal 7 call 4
512:literal 3 literal 3 literal 0 call 4
524:literal 4 literal 1 literal 1 call 4
536:literal 4 literal 2 literal 8 call 4
548:literal 4 literal 3 literal 0 call 4
560:literal 5 literal 1 literal 1 call 4
572:literal 5 literal 2 literal 6 call 4
584:literal 5 literal 3 literal 9 call 4
596:literal 6 literal 1 literal 3 call 4
608:literal 6 literal 2 literal 5 call 4
620:literal 6 literal 3 literal 10 call 4
632:literal 7 literal 1 literal 3 call 4
```

```
644:literal 7 literal 2 literal 10 call 4
656:literal 7 literal 3 literal 0 call 4
668:literal 8 literal 1 literal 4 call 4
680:literal 8 literal 2 literal 0 call 4
692:literal 8 literal 3 literal 0 call 4
704:literal 9 literal 1 literal 10 call 4
716:literal 9 literal 2 literal 5 call 4
728:literal 9 literal 3 literal 0 call 4
740:literal 10 literal 1 literal 6 call 4
752:literal 10 literal 2 literal 7 call 4
764:literal 10 literal 3 literal 9 call 4
776:lvalueGV 177 literal 1 index literal 1 setGV
787:lvalueGV 177 literal 2 index literal 2 setGV
798:lvalueGV 177 literal 3 index literal 3 setGV
809:lvalueLV 1 literal 3 set
816:rvalueLV 1 jz 858
822:lvalueGV 220 rvalueLV 1 index literal 1 setGV
833:lvalueGV 216 rvalueLV 1 index literal 0 setGV
844:lvalueLV 1 rvalueLV 1 literal 1 minus set
855:jmp 816
858:lvalueLV 1 literal 100 set
865:rvalueLV 1 jz 896
871:lvalueGV 32 rvalueLV 1 index literal 0 setGV
882:lvalueLV 1 rvalueLV 1 literal 1 minus set
893:jmp 865
896:lvalueLV 2 literal 0 set
903:ret0
904:func 2 0
909:rvalueLV 1 rvalueLV 2 lt
916:jz 943
919:lvalueGV 32 rvalueLV 2 literal 10 multiply literal 10 minus rvalueLV 1 plus index fetchGV
    ret1
940:jmp 964
943:lvalueGV 32 rvalueLV 1 literal 10 multiply literal 10 minus rvalueLV 2 plus index fetchGV
    ret1
964:ret0
965:func 3 0
970:rvalueLV 1 rvalueLV 2 lt
977:jz 1006
980:lvalueGV 32 rvalueLV 2 literal 10 multiply literal 10 minus rvalueLV 1 plus index
    rvalueLV 3 setGV
1003:jmp 1029
1006:lvalueGV 32 rvalueLV 1 literal 10 multiply literal 10 minus rvalueLV 2 plus index
     rvalueLV 3 setGV
1029:ret0
1030:func 3 1
1035:lvalueGV 133 literal 1 setGV
1042:rvalueLV 2 rvalueLV 3 ne
1049:jz 1619
1052:wait 224
1055:rvalueLV 2 rvalueLV 3 call 904
1064:literal 0 equal
1068:jz 1241
1071:rvalueLV 2 rvalueLV 3 rvalueLV 1 call 965
1083:literal 82 printCh
1087:literal 111 printCh
```

```
1091:literal 98 printCh
1095:literal 111 printCh
1099:literal 116 printCh
1103:literal 32 printCh
1107:literal 110 printCh
1111:literal 117 printCh
1115:literal 109 printCh
1119:literal 98 printCh
1123:literal 101 printCh
1127:literal 114 printCh
1131:literal 32 printCh
1135:rvalueLV 1 print
1139:literal 32 printCh
1143:literal 58 printCh
1147:literal 32 printCh
1151:rvalueLV 2 print
1155:literal 32 printCh
1159:literal 45 printCh
1163:literal 62 printCh
1167:literal 32 printCh
1171:rvalueLV 3 print
1175:literal 32 printCh
1179:literal 115 printCh
1183:literal 116 printCh
1187:literal 97 printCh
1191:literal 114 printCh
1195:literal 116 printCh
1199:literal 32 printCh
1203:literal 109 printCh
1207:literal 111 printCh
1211:literal 118 printCh
1215:literal 101 printCh
1219:literal 13 printCh
1223:literal 10 printCh
1227:lvalueGV 216 rvalueLV 1 index literal 2 setGV
1238:jmp 1372
1241:literal 82 printCh
1245:literal 111 printCh
1249:literal 98 printCh
1253:literal 111 printCh
1257:literal 116 printCh
1261:literal 32 printCh
1265:literal 110 printCh
1269:literal 117 printCh
1273:literal 109 printCh
1277:literal 98 printCh
1281:literal 101 printCh
1285:literal 114 printCh
1289:literal 32 printCh
1293:rvalueLV 1 print
1297:literal 32 printCh
1301:literal 58 printCh
1305:literal 32 printCh
1309:rvalueLV 2 print
1313:literal 32 printCh
1317:literal 45 printCh
```

```
1321:literal 62 printCh
1325:literal 32 printCh
1329:rvalueLV 3 print
1333:literal 32 printCh
1337:literal 119 printCh
1341:literal 97 printCh
1345:literal 105 printCh
1349:literal 116 printCh
1353:literal 13 printCh
1357:literal 10 printCh
1361:lvalueGV 216 rvalueLV 1 index literal 3 setGV
1372:signal 224
1375:rvalueLV 2 rvalueLV 3 call 904
1384:rvalueLV 1 equal
1388:jz 1616
1391:lvalueLV 4 literal 10 set
1398:rvalueLV 4 jz 1418
1404:lvalueLV 4 rvalueLV 4 literal 1 minus set
1415:jmp 1398
1418:wait 224
1421:literal 82 printCh
1425:literal 111 printCh
1429:literal 98 printCh
1433:literal 111 printCh
1437:literal 116 printCh
1441:literal 32 printCh
1445:literal 110 printCh
1449:literal 117 printCh
1453:literal 109 printCh
1457:literal 98 printCh
1461:literal 101 printCh
1465:literal 114 printCh
1469:literal 32 printCh
1473:rvalueLV 1 print
1477:literal 32 printCh
1481:literal 58 printCh
1485:literal 32 printCh
1489:rvalueLV 2 print
1493:literal 32 printCh
1497:literal 45 printCh
1501:literal 62 printCh
1505:literal 32 printCh
1509:rvalueLV 3 print
1513:literal 32 printCh
1517:literal 102 printCh
1521:literal 105 printCh
1525:literal 110 printCh
1529:literal 105 printCh
1533:literal 115 printCh
1537:literal 104 printCh
1541:literal 32 printCh
1545:literal 109 printCh
1549:literal 111 printCh
1553:literal 118 printCh
1557:literal 101 printCh
1561:literal 13 printCh
```

```
1565:literal 10 printCh
1569:rvalueLV 2 rvalueLV 3 literal 0 call 965
1581:lvalueGV 216 rvalueLV 1 index literal 1 setGV
1592:signal 224
1595:lvalueGV 220 rvalueLV 1 index rvalueLV 3 setGV
1606:lvalueLV 2 rvalueLV 3 set
1613:jmp 1616
1616:jmp 1042
1619:ret0
1620:func 3 3
1625:lvalueGV 142 rvalueLV 1 literal 10 multiply literal 10 minus rvalueLV 3 plus index
      rvalueLV 3 setGV
1648:lvalueGV 220 rvalueLV 1 index fetchGV lvalueGV 134 rvalueLV 1 index fetchGV equal
1665:jz 1682
1668:lvalueGV 216 rvalueLV 1 index literal 0 setGV
1679:jmp 1682
1682:rvalueLV 2 rvalueLV 3 equal
1689:jz 1706
1692:lvalueGV 173 rvalueLV 1 index literal 1 setGV
1703:jmp 1706
1706:lvalueLV 4 literal 1 set
1713:rvalueLV 4 literal 3 le
1720:jz 1807
1723:lvalueLV 5 rvalueLV 3 rvalueLV 4 call 33
1735:set
1736:rvalueLV 5 literal 0 gt
1743:lvalueGV 142 rvalueLV 1 literal 10 multiply literal 10 minus rvalueLV 5 plus index fetchGV
      not and lvalueGV 173 rvalueLV 1 index fetchGV not and jz 1793
1778:rvalueLV 1 rvalueLV 2 rvalueLV 5 call 1620
1790:jmp 1793
1793:lvalueLV 4 rvalueLV 4 literal 1 plus set
1804:jmp 1713
1807:lvalueLV 6 lvalueGV 220 rvalueLV 1 index fetchGV set
1819:lvalueGV 173 rvalueLV 1 index fetchGV jz 1856
1830:lvalueGV 181 rvalueLV 1 index rvalueLV 3 setGV
1841:rvalueLV 1 rvalueLV 6 rvalueLV 3 call 1030
1853:jmp 1856
1856:lvalueGV 220 rvalueLV 1 index fetchGV lvalueGV 134 rvalueLV 1 index fetchGV equal
1873:jz 1890
1876:lvalueGV 216 rvalueLV 1 index literal 0 setGV
1887:jmp 1890
1890:lvalueGV 142 rvalueLV 1 literal 10 multiply literal 10 minus rvalueLV 3 plus index
      literal 0 setGV
1913:ret0
1914:func 1 2
1919:literal 1 jz 2118
1925:lvalueLV 2 literal 4 receive
1932:set
1933:lvalueGV 134 rvalueLV 1 index rvalueLV 2 setGV
1944:lvalueGV 216 rvalueLV 1 index literal 1 setGV
1955:lvalueGV 173 rvalueLV 1 index literal 0 setGV
1966:rvalueLV 1 lvalueGV 220 rvalueLV 1 index fetchGV rvalueLV 2 call 1620
1983:lvalueGV 216 rvalueLV 1 index literal 0 setGV
1994:lvalueGV 133 literal 1 setGV
2001:lvalueGV 220 rvalueLV 1 index fetchGV rvalueLV 3 equal
2013:jz 2115
```

```
2016:literal 114 printCh
2020:literal 111 printCh
2024:literal 98 printCh
2028:literal 111 printCh
2032:literal 116 printCh
2036:literal 32 printCh
2040:literal 110 printCh
2044:literal 117 printCh
2048:literal 109 printCh
2052:literal 98 printCh
2056:literal 101 printCh
2060:literal 114 printCh
2064:literal 32 printCh
2068:rvalueLV 1 print
2072:literal 32 printCh
2076:literal 99 printCh
2080:literal 97 printCh
2084:literal 110 printCh
2088:literal 110 printCh
2092:literal 111 printCh
2096:literal 116 printCh
2100:literal 32 printCh
2104:literal 103 printCh
2108:literal 111 printCh
2112:jmp 2115
2115:jmp 1919
2118:ret0
2119:proc 1 0 0
2126:literal 1 call 1914
2132:EOF proc 2 0 0
2140:literal 2 call 1914
2146:EOF proc 3 0 0
2154:literal 3 call 1914
2160:EOF proc 4 1 2
2168:rvalueLV 1 jz 2430
2174:lvalueLV 2 literal 3 random literal 1 plus set
2186:lvalueLV 3 literal 10 random literal 1 plus set
2198:lvalueGV 216 rvalueLV 2 index fetchGV not lvalueGV 220 rvalueLV 2 index fetchGV
      rvalueLV 3 ne
2219:and jz 2427
2223:wait 224
2226:literal 71 printCh
2230:literal 101 printCh
2234:literal 110 printCh
2238:literal 101 printCh
2242:literal 114 printCh
2246:literal 97 printCh
2250:literal 116 printCh
2254:literal 101 printCh
2258:literal 32 printCh
2262:literal 106 printCh
2266:literal 111 printCh
2270:literal 98 printCh
2274:literal 32 printCh
2278:literal 102 printCh
2282:literal 111 printCh
```

```
2286:literal 114 printCh
2290:literal 32 printCh
2294:literal 114 printCh
2298:literal 111 printCh
2302:literal 98 printCh
2306:literal 111 printCh
2310:literal 116 printCh
2314:literal 32 printCh
2318:literal 110 printCh
2322:literal 117 printCh
2326:literal 109 printCh
2330:literal 98 printCh
2334:literal 101 printCh
2338:literal 114 printCh
2342:literal 32 printCh
2346:rvalueLV 2 print
2350:literal 32 printCh
2354:literal 109 printCh
2358:literal 111 printCh
2362:literal 118 printCh
2366:literal 101 printCh
2370:literal 32 printCh
2374:literal 116 printCh
2378:literal 111 printCh
2382:literal 32 printCh
2386:rvalueLV 3 print
2390:literal 13 printCh
2394:literal 10 printCh
2398:signal 224
2401:lvalueGV 177 rvalueLV 2 index fetchGV rvalueLV 3 send
2413:lvalueLV 1 rvalueLV 1 literal 1 minus set
2424:jmp 2427
2427:jmp 2168
2430:Halt EOF proc 5 0 0
2439:call 60
2442:EOF func 0 0
2448:lvalueGV 220 literal 3 setGV
2455:lvalueGV 32 literal 100 setGV
2462:lvalueGV 173 literal 3 setGV
2469:lvalueGV 185 literal 30 setGV
2476:lvalueGV 1 literal 30 setGV
2483:lvalueGV 138 literal 3 setGV
2490:lvalueGV 134 literal 3 setGV
2497:lvalueGV 177 literal 3 setGV
2504:lvalueGV 216 literal 3 setGV
2511:lvalueGV 181 literal 3 setGV
2518:lvalueGV 142 literal 30 setGV
2525:lvalueGV 133 literal 0 setGV
2532:call 411
2535:lvalueGV 224 literal 1 setGV
2542:literal 80 call 2161
2548:call 2119
2551:call 2133
2554:call 2147
2557:EOF
```

3.ผลการรัน

3.1 เมื่อกำหนดให้ควอนตัมมีค่าเท่ากับ 1 (ผลการทดลองเพียงบางส่วน)

```
start at node 1
Generate job for robot number 3  move to 3
Generate job for robot number 1  move to 7
Generate job for robot number 2  move to 5
Robot number 3  : 1  -> 5  start move
Robot number 2  : 1  -> 5  wait
Robot number 1  : 1  -> 5  wait
Robot number 3  : 1  -> 5  finish move
Robot number 2  : 1  -> 5  start move
Robot number 1  : 1  -> 5  wait
Robot number 3  : 5  -> 6  start move
Robot number 2  : 1  -> 5  finish move
Robot number 1  : 1  -> 5  start move
Robot number 3  : 5  -> 6  finish move
Robot number 1  : 1  -> 5  finish move
Generate job for robot number 2  move to 9
Robot number 3  : 6  -> 3  start move
Robot number 1  : 5  -> 6  start move
Robot number 3  : 6  -> 3  finish move
Robot number 1  : 5  -> 6  finish move
Generate job for robot number 3  move to 2
Robot number 1  : 6  -> 3  start move
Robot number 1  : 6  -> 3  finish move
Robot number 2  : 5  -> 6  start move
Robot number 1  : 3  -> 7  start move
Robot number 2  : 5  -> 6  finish move
Robot number 1  : 3  -> 7  finish move
Robot number 2  : 6  -> 10  start move
Generate job for robot number 1  move to 2
Robot number 2  : 6  -> 10  finish move
Robot number 3  : 3  -> 6  start move
Robot number 2  : 10  -> 9  start move
Robot number 3  : 3  -> 6  finish move
Robot number 2  : 10  -> 9  finish move
Robot number 3  : 6  -> 5  start move
Generate job for robot number 2  move to 1
Robot number 3  : 6  -> 5  finish move
Robot number 3  : 5  -> 1  start move
Robot number 3  : 5  -> 1  finish move
Robot number 1  : 7  -> 3  start move
Robot number 3  : 1  -> 2  start move
Robot number 1  : 7  -> 3  finish move
Robot number 3  : 1  -> 2  finish move
Generate job for robot number 3  move to 5
Robot number 1  : 3  -> 6  start move
Robot number 1  : 3  -> 6  finish move
Robot number 1  : 6  -> 5  start move
```

```
Robot number 1  : 3  -> 7  finish move
Robot number 2  : 10  -> 7  start move
Generate job for robot number 1  move to 3
Robot number 2  : 10  -> 7  finish move
Robot number 3  : 6  -> 5  start move
Robot number 2  : 7  -> 3  start move
Robot number 3  : 6  -> 5  finish move
Robot number 2  : 7  -> 3  finish move
Robot number 3  : 5  -> 1  start move
Robot number 2  : 3  -> 6  start move
Robot number 3  : 5  -> 1  finish move
Robot number 2  : 3  -> 6  finish move
Generate job for robot number 2  move to 5
Robot number 3  : 1  -> 2  start move
Robot number 3  : 1  -> 2  finish move
Generate job for robot number 3  move to 7
Robot number 1  : 7  -> 10  start move
Robot number 1  : 7  -> 10  finish move
Robot number 1  : 10  -> 9  start move
Robot number 1  : 10  -> 9  finish move
Robot number 3  : 2  -> 1  start move
Robot number 2  : 6  -> 5  start move
Robot number 1  : 9  -> 5  start move
Robot number 3  : 2  -> 1  finish move
Robot number 2  : 6  -> 5  finish move
Robot number 1  : 9  -> 5  finish move
Robot number 3  : 1  -> 5  start move
Generate job for robot number 2  move to 8
Robot number 1  : 5  -> 6  start move
Robot number 3  : 1  -> 5  finish move
Robot number 1  : 5  -> 6  finish move
Robot number 3  : 5  -> 6  start move
Robot number 1  : 6  -> 3  start move
Robot number 3  : 5  -> 6  finish move
Robot number 1  : 6  -> 3  finish move
Robot number 2  : 5  -> 1  start move
Generate job for robot number 1  move to 10
Robot number 3  : 6  -> 3  start move
Robot number 2  : 5  -> 1  finish move
Robot number 3  : 6  -> 3  finish move
Robot number 2  : 1  -> 4  start move
Robot number 3  : 3  -> 7  start move
Robot number 2  : 1  -> 4  finish move
Robot number 1  : 3  -> 6  start move
Robot number 3  : 3  -> 7  finish move
Robot number 2  : 4  -> 8  start move
Robot number 1  : 3  -> 6  finish move
Generate job for robot number 3  move to 8
Robot number 2  : 4  -> 8  finish move
Robot number 1  : 6  -> 10  start move
Generate job for robot number 2  move to 5
Robot number 1  : 6  -> 10  finish move
Generate job for robot number 1  move to 3
Robot number 2  : 8  -> 4  start move
Robot number 3  : 7  -> 3  start move
Robot number 2  : 8  -> 4  finish move
```

```
Robot number 3  : 7  -> 3  finish move
Robot number 2  : 4  -> 1  start move
Robot number 3  : 3  -> 6  start move
Robot number 2  : 4  -> 1  finish move
Robot number 3  : 3  -> 6  finish move
Robot number 2  : 1  -> 5  start move
Robot number 1  : 10  -> 9  start move
Robot number 3  : 6  -> 5  start move
Robot number 2  : 1  -> 5  finish move
Robot number 1  : 10  -> 9  finish move
Generate job for robot number 2  move to 9
Robot number 3  : 6  -> 5  finish move
Robot number 1  : 9  -> 5  start move
Robot number 3  : 5  -> 1  start move
Robot number 1  : 9  -> 5  finish move
Robot number 3  : 5  -> 1  finish move
Robot number 1  : 5  -> 6  start move
Robot number 3  : 1  -> 4  start move
Robot number 1  : 5  -> 6  finish move
Robot number 3  : 1  -> 4  finish move
Robot number 2  : 5  -> 6  start move
Robot number 1  : 6  -> 3  start move
Robot number 3  : 4  -> 8  start move
Robot number 2  : 5  -> 6  finish move
Robot number 1  : 6  -> 3  finish move
Robot number 3  : 4  -> 8  finish move
Robot number 2  : 6  -> 10  start move
Generate job for robot number 1  move to 8
Robot number 2  : 6  -> 10  finish move
Generate job for robot number 3  move to 5
Robot number 2  : 10  -> 9  start move
Robot number 2  : 10  -> 9  finish move
Generate job for robot number 2  move to 8
Robot number 3  : 8  -> 4  start move
Robot number 1  : 3  -> 6  start move
Robot number 3  : 8  -> 4  finish move
Robot number 1  : 3  -> 6  finish move
Robot number 3  : 4  -> 1  start move
Robot number 1  : 6  -> 5  start move
Robot number 3  : 4  -> 1  finish move
Robot number 1  : 6  -> 5  finish move
Robot number 3  : 1  -> 5  start move
Robot number 1  : 5  -> 1  wait
Robot number 3  : 1  -> 5  finish move
Robot number 2  : 9  -> 10  start move
Robot number 1  : 5  -> 1  start move
Generate job for robot number 3  move to 4
Robot number 2  : 9  -> 10  finish move

Quantum                 : 1
No of instruction       : 309006
Block Count             : 360165
Total execute time      : 14.99
Instruction per sec     : 20614.14
```

3.2 เมื่อกำหนดให้ควอนตัมมีค่าเท่ากับ 1000

```
start at node 1
Generate job for robot number 3  move to 3
Generate job for robot number 1  move to 7
Robot number 3  : 1  -> 5  start move
Robot number 3  : 1  -> 5  finish move
Robot number 3  : 5  -> 6  start move
Robot number 3  : 5  -> 6  finish move
Robot number 3  : 6  -> 3  start move
Robot number 3  : 6  -> 3  finish move
Robot number 1  : 1  -> 5  start move
Robot number 1  : 1  -> 5  finish move
Robot number 1  : 5  -> 6  start move
Robot number 1  : 5  -> 6  finish move
Robot number 1  : 6  -> 3  start move
Robot number 1  : 6  -> 3  finish move
Robot number 1  : 3  -> 7  start move
Robot number 1  : 3  -> 7  finish move
Generate job for robot number 1  move to 2
Generate job for robot number 2  move to 5
Generate job for robot number 3  move to 4
Robot number 2  : 1  -> 5  start move
Robot number 2  : 1  -> 5  finish move
Robot number 1  : 7  -> 3  start move
Robot number 1  : 7  -> 3  finish move
Robot number 1  : 3  -> 6  start move
Generate job for robot number 3  move to 4
Robot number 1  : 3  -> 6  finish move
Robot number 1  : 6  -> 5  start move
Robot number 1  : 6  -> 5  finish move
Robot number 1  : 5  -> 1  start move
Robot number 1  : 5  -> 1  finish move
Robot number 1  : 1  -> 2  start move
Robot number 1  : 1  -> 2  finish move
Generate job for robot number 2  move to 3
Robot number 3  : 3  -> 6  start move
Robot number 3  : 3  -> 6  finish move
Robot number 3  : 6  -> 5  start move
Robot number 3  : 6  -> 5  finish move
Robot number 3  : 5  -> 1  start move
Robot number 3  : 5  -> 1  finish move
Robot number 3  : 1  -> 4  start move
Robot number 3  : 1  -> 4  finish move
Generate job for robot number 3  move to 3
Robot number 2  : 5  -> 6  start move
Robot number 2  : 5  -> 6  finish move
Generate job for robot number 3  move to 3
Robot number 3  : 4  -> 1  start move
Robot number 3  : 4  -> 1  finish move
Robot number 3  : 1  -> 5  start move
Robot number 3  : 1  -> 5  finish move
Robot number 3  : 5  -> 6  start move
Robot number 3  : 5  -> 6  finish move
Robot number 3  : 6  -> 3  start move
```

```
Robot number 1  : 4  -> 1  start move
Robot number 1  : 4  -> 1  finish move
Robot number 1  : 1  -> 2  start move
Robot number 1  : 1  -> 2  finish move
Generate job for robot number 2  move to 10
Generate job for robot number 2  move to 5
Robot number 3  : 1  -> 4  start move
Robot number 3  : 1  -> 4  finish move
Robot number 3  : 4  -> 8  start move
Robot number 3  : 4  -> 8  finish move
Generate job for robot number 3  move to 1
Robot number 2  : 9  -> 5  start move
Robot number 2  : 9  -> 5  finish move
Robot number 2  : 5  -> 6  start move
Robot number 2  : 5  -> 6  finish move
Robot number 2  : 6  -> 10  start move
Robot number 2  : 6  -> 10  finish move
Generate job for robot number 1  move to 1
Robot number 3  : 8  -> 4  start move
Generate job for robot number 1  move to 8
Robot number 3  : 8  -> 4  finish move
Robot number 3  : 4  -> 1  start move
Robot number 3  : 4  -> 1  finish move
Robot number 2  : 10  -> 7  start move
Robot number 2  : 10  -> 7  finish move
Robot number 2  : 7  -> 3  start move
Robot number 2  : 7  -> 3  finish move
Robot number 2  : 3  -> 6  start move
Robot number 2  : 3  -> 6  finish move
Robot number 2  : 6  -> 5  start move
Robot number 2  : 6  -> 5  finish move
Robot number 1  : 2  -> 1  start move
Robot number 1  : 2  -> 1  finish move
Robot number 1  : 1  -> 4  start move
Robot number 1  : 1  -> 4  finish move
Robot number 1  : 4  -> 8  start move
Robot number 1  : 4  -> 8  finish move
Generate job for robot number 3  move to 4
Generate job for robot number 3  move to 5
Robot number 3  : 1  -> 4  start move
Robot number 3  : 1  -> 4  finish move
Generate job for robot number 2  move to 9
Robot number 3  : 4  -> 1  start move
Robot number 3  : 4  -> 1  finish move
Robot number 3  : 1  -> 5  start move
Robot number 3  : 1  -> 5  finish move
Robot number 3  : 1  -> 5  start move
Robot number 3  : 1  -> 5  finish move
Generate job for robot number 1  move to 1
Generate job for robot number 1  move to 2
Robot number 2  : 5  -> 6  start move
Robot number 2  : 5  -> 6  finish move
Robot number 2  : 6  -> 10  start move
Robot number 2  : 6  -> 10  finish move
Robot number 1  : 8  -> 4  start move
Generate job for robot number 3  move to 3
```

```
Robot number 2  : 10  -> 9  start move
Robot number 2  : 10  -> 9  finish move
Robot number 1  : 8  -> 4  finish move
Robot number 1  : 4  -> 1  start move
Robot number 1  : 4  -> 1  finish move
Generate job for robot number 2  move to 3
Robot number 3  : 5  -> 6  start move
Robot number 3  : 5  -> 6  finish move
Robot number 1  : 1  -> 2  start move
Robot number 1  : 1  -> 2  finish move
Generate job for robot number 1  move to 4
Robot number 3  : 6  -> 3  start move
Robot number 3  : 6  -> 3  finish move
Generate job for robot number 3  move to 4
Robot number 2  : 9  -> 5  start move
Robot number 2  : 9  -> 5  finish move
Robot number 1  : 2  -> 1  start move
Robot number 1  : 2  -> 1  finish move
Generate job for robot number 3  move to 8
Robot number 2  : 5  -> 6  start move
Robot number 2  : 5  -> 6  finish move
Robot number 2  : 6  -> 3  start move
Robot number 2  : 6  -> 3  finish move
Robot number 1  : 1  -> 4  start move
Robot number 1  : 1  -> 4  finish move
Generate job for robot number 2  move to 8
Robot number 3  : 3  -> 6  start move
Robot number 3  : 3  -> 6  finish move
Robot number 3  : 6  -> 5  start move
Robot number 3  : 6  -> 5  finish move
Robot number 3  : 5  -> 1  start move
Robot number 3  : 5  -> 1  finish move
Robot number 3  : 1  -> 4  start move
Robot number 3  : 1  -> 4  finish move
Generate job for robot number 2  move to 6
Robot number 3  : 4  -> 8  start move
Robot number 3  : 4  -> 8  finish move
Generate job for robot number 3  move to 2
Robot number 2  : 3  -> 6  start move
Robot number 2  : 3  -> 6  finish move
Robot number 2  : 6  -> 5  start move
Robot number 2  : 6  -> 5  finish move
Robot number 2  : 5  -> 1  start move
Robot number 2  : 5  -> 1  finish move
Robot number 2  : 1  -> 4  start move
Robot number 2  : 1  -> 4  finish move
Robot number 2  : 4  -> 8  start move
Robot number 2  : 4  -> 8  finish move

Quantum                 : 1000
No of instruction       : 206518
Block Count             : 507
Total execute time      : 8.62
Instruction per sec     : 23958.00
```

4.ตารางเปรียบเทียบผลการรันเมื่อเปลี่ยนแปลงควอนตัม

ควอนตัม	จำนวนคำสั่งที่ประมวลผล	จำนวนครั้งที่บล็อก	เวลาในการประมวลผล (sec)	ความเร็ว (intr./sec)
1	1,066,032	1,291,336	44.82	23,784.74
2	1,054,954	625,896	39.00	27,050.10
4	1,019,083	296,037	35.26	28,901.96
8	1,002,091	152,527	33.94	29,525.37
16	1,063,275	75,802	35.15	30,249.64
32	1,011,235	43,420	33.51	30,177.11
64	993,837	23,541	32.79	30,309.15
128	957,765	11,513	31.48	30,424.56
256	1,053,808	6,845	34.49	30,554.02
512	1,028,699	3,525	33.67	30,552.39
1,024	971,450	1,947	31.85	30,500.78
2,048	957,191	1,146	31.42	30,464.39
4,096	860,873	822	28.28	30,441.05
8,192	806,719	794	26.48	30,465.22
16,384	804,933	794	26.47	30,409.26
32,768	804,933	794	26.42	30,466.81
65,536	804,933	794	26.36	30,536.15

ภาคผนวก ค

โปรแกรมอนุกรม Fibonacci

กำหนดให้ Fib(0) เท่ากับศูนย์และ Fib(1) เท่ากับหนึ่ง อนุกรม Fibonacci เป็นอนุกรมของตัวเลขที่ได้จากผลบวกของของเลขสองตัวที่อยู่ลำดับก่อนหน้า

Fib(0) = 0

Fib(1) = 1ที่ได้จากผลบวกของ

Fib(n) = $F_{n-1} + F_{n-2}$ เมื่อ $n \geq 1$

1.โปรแกรมต้นฉบับ

```
function fib(n)
{
  if (n <= 2)
    return(1)
   else
     return( fib(n-1) + fib(n-2) ) ;
}

run
{
 n = 1;
 while (N<=23)
 {
   println('Fibonaci of ',n,' is ',fib(n));
   n=n+1;
 }
}
```

2.รหัสกลางที่ได้

```
 1:call 48
 4:func 1 0
 9:rvalueLV 1 literal 2 le
16:jz 26
19:literal 1 ret1
23:jmp 48
26:rvalueLV 1 literal 1 minus call 4
36:rvalueLV 1 literal 2 minus call 4
46:plus ret1
48:func 0 1
53:lvalueLV 1 literal 1 set
60:rvalueLV 1 literal 23 le
67:jz 167
70:literal 70 printCh
74:literal 105 printCh
78:literal 98 printCh
82:literal 111 printCh
86:literal 110 printCh
90:literal 97 printCh
94:literal 99 printCh
98:literal 105 printCh
102:literal 32 printCh
106:literal 111 printCh
110:literal 102 printCh
114:literal 32 printCh
118:rvalueLV 1 print
122:literal 32 printCh
126:literal 105 printCh
130:literal 115 printCh
134:literal 32 printCh
138:rvalueLV 1 call 4
144:print
145:literal 13 printCh
149:literal 10 printCh
153:lvalueLV 1 rvalueLV 1 literal 1 plus set
164:jmp 60
167:EOF
```

3.ผลการรัน

```
Fibonaci of 1  is 1
Fibonaci of 2  is 1
Fibonaci of 3  is 2
Fibonaci of 4  is 3
Fibonaci of 5  is 5
Fibonaci of 6  is 8
Fibonaci of 7  is 13
Fibonaci of 8  is 21
Fibonaci of 9  is 34
Fibonaci of 10  is 55
Fibonaci of 11  is 89
Fibonaci of 12  is 144
Fibonaci of 13  is 233
Fibonaci of 14  is 377
Fibonaci of 15  is 610
Fibonaci of 16  is 987
Fibonaci of 17  is 1597
Fibonaci of 18  is 2584
Fibonaci of 19  is 4181
Fibonaci of 20  is 6765
Fibonaci of 21  is 10946
Fibonaci of 22  is 17711
Fibonaci of 23  is 28657
program terminate normally.

Quantum                 :        1
Size of mail buffer     :        1
No of instruction       :  1651366
Block Count             :  1651366
Total execute time      :    61.74
instruction per sec.    : 26747.10
```

ภาคผนวก ง

โปรแกรมหอคอยแห่งเมืองฮานอย

1.โปรแกรมต้นฉบับ

```
global p[30], n[3];
function push(Diskno,poleNo)
{
  n[PoleNo]    = n[poleno] + 1;
  p[PoleNo*10-10+n[poleNo]] = diskNo;
}

function pop(poleNo)
{
  n[PoleNo]    = n[poleno] - 1;
  return( p[PoleNo*10-10+n[poleNo]+1]);
}

function TopOfDisk(diskNo,poleNo)
{
  return( p[PoleNo*10-10+n[poleNo] ] == diskNo);
}

function Level(diskNo,poleNo)
{
  L = 10;
  while  (DiskNo <> p[poleNo*10-10+L] )  { L=L-1; }
  return(L);
}

function MoveDisk(DiskNo,From,to)
{
  temp = 6-From-to;
  if  ( TopOfDisk(DiskNo,From) )
  {
    println('move disk no ',DiskNo,'  from ',from,' to ',to);
    Pop(From);
    push(DiskNo,to);
  }
  else
  {
    OnTop = p[ from*10-10 + level(diskno,from)+1];
```

```
   MoveDisk( OnTop, from, temp);
   MoveDisk( DiskNo, from, to);
   MoveDisk( ontop ,  temp, to);
  }
}

Run
{
  NumOfDisk = 5;
  i=NumOfDisk * 3;     // set all pole to zero
  while(i<>0)
  {
    p[i]=0;
    i=i-1;
  }
  n[1] = 0;
  n[2] = 0;
  n[3] = 0;

  i=NumOfDisk;
  while(i<>0)
  {
    push(i,1);
    i=i-1;
  }
  println('Tower of Hanoi problem');
  println('======================');

  println('                                  ');
  println('                |          |          |  ');
  println('                1|1         |          |  ');
  println('               22|22        |          |  ');
  println('              333|333       |          |  ');
  println('             4444|4444      |          |  ');
  println('            55555|55555     |          |  ');
  println('              Pole 1         Pole 2     Pole 3 ');
  println('                                  ');
  MoveDisk(NumOfDisk,1,2);
}
```

~

2.ผลการรัน

```
Tower of Hanoi problem
======================

                |           |           |
              1|1           |           |
             22|22          |           |
            333|333         |           |
           4444|4444        |           |
          55555|55555       |           |
            Pole 1          Pole 2      Pole 3

move disk no 1   from 1  to 2
move disk no 2   from 1  to 3
move disk no 1   from 2  to 3
move disk no 3   from 1  to 2
move disk no 1   from 3  to 1
move disk no 2   from 3  to 2
move disk no 1   from 1  to 2
move disk no 4   from 1  to 3
move disk no 1   from 2  to 3
move disk no 2   from 2  to 1
move disk no 1   from 3  to 1
move disk no 3   from 2  to 3
move disk no 1   from 1  to 2
move disk no 2   from 1  to 3
move disk no 1   from 2  to 3
move disk no 5   from 1  to 2
move disk no 1   from 3  to 1
move disk no 2   from 3  to 2
move disk no 1   from 1  to 2
move disk no 3   from 3  to 1
move disk no 1   from 2  to 3
move disk no 2   from 2  to 1
move disk no 1   from 3  to 1
move disk no 4   from 3  to 2
move disk no 1   from 1  to 2
move disk no 2   from 1  to 3
move disk no 1   from 2  to 3
move disk no 3   from 1  to 2
move disk no 1   from 3  to 1
move disk no 2   from 3  to 2
move disk no 1   from 1  to 2
program terminate normally.

Quantum                 :        1
Size of mail buffer     :        1
No of instruction       :     9403
Block Count             :     9403
Total execute time      :     0.50
instruction per sec.    : 18806.00
```

~

ภาคผนวก จ

การแปลรหัสกลางไปเป็นภาษาเป้าหมาย

เนื่องจากงานวิจัยนี้เป็นอินเตอร์พรีเตอร์จึงมีความเร็วต่ำ การปรับปรุงเพื่อเพิ่มประสิทธิภาพอาจทำได้โดยการแปลรหัสกลางที่ได้ไปเป็นภาษาเป้าหมายโดยตรง ในบทนี้จะแสดงวิธีการแปลรหัสกลางที่ได้จากขั้นตอนคอมไพเลอร์ไปเป็นภาษาเป้าหมายคือภาษาปาลกาล โดยเลือกเอาตัวอย่างจากรหัสกลางของโปรแกรมอนุกรม Fibonacci ที่ปรากฏในภาคผนวก ค.

การทำงานของอินเตอร์พรีเตอร์จะมีโปรแกรมย่อยที่อ่านรหัสกลางที่ได้มาประมวลผล โดยเมื่อได้รหัสกลางหนึ่งๆจะนำไปตรวจสอบว่าเป็นคำสั่งใดและจะเรียกโปรแกรมย่อยของคำสั่งนั้นๆ มาประมวลผลดังนี้

```
Procedure Execute1Instruc;
begin
        ...
        ...
 case Icode of
   icLiteral      : ExecLiteral(Operand);
   icJmp          : ExecJmp(operand);
   icJZ           : ExecJZ(operand);
   icCall         : ExecCall(operand);
   icSet          : ExecSet;
        ...
        ...
   icEOF          : ExecEOF;
 end; {case}
end;
```

ถ้าเราต้องการแปลรหัสกลางที่ได้ไปเป็นภาษาเป้าหมาย (ปาลกาล) จะทำได้โดยตัดส่วนของโปรแกรมย่อย Exec1Instruc ออกและแปลรหัสกลางต่างๆไปเป็นคำสั่งในการเรียกโปรแกรมย่อยได้โดยตรง โดยมีขั้นตอนดังนี้

1.ขั้นตอนการแปลรหัสกลางไปเป็นภาษาเป้าหมาย

1.แอดเดรสของตัวชี้คำสั่ง (instruction pointer) ของรหัสกลาง จะเปลี่ยนเป็น label ในภาษาปาลกาล และการอ้างอิงตำแหน่งของตัวชี้คำสั่งจะอ้างอิงจาก label เพื่อที่จะย้ายตำแหน่งในการประมวลผลไปสู่แอดเดรสที่ต้องการได้

2.รหัสกลางที่ไม่เกี่ยวข้องกับการเปลี่ยนแปลงค่าตัวชี้คำสั่ง สามารถเปลี่ยนเป็นการเรียกโปรแกรมย่อย (procedure call) ของภาษาปาลกาลได้โดยตรงเช่นรหัสกลาง lilteral 3 จะแปลเป็น literal(3); หรือ rvalueLV 1 จะแปลเป็น rvalueLV(1); ได้โดยตรง

3.รหัสกลางที่เปลี่ยนแปลงค่าของตัวชี้คำสั่ง จะมีผลทำให้โปรแกรมย้ายตำแหน่งของคำสั่งในการประมวลผล โดยในอินเตอร์พรีเตอร์จะย้ายตำแหน่งในการประมวลผลไปที่ตำแหน่งอื่นโดยการเปลี่ยนแปลงค่าตัวชี้คำสั่ง แต่ในปาลคาลจำเป็นต้องย้ายโดยคำสั่ง goto label ในตัวอย่างนี้ (Fibonacci series) จะมีคำสั่งที่ทำให้ตัวชี้คำสั่งเปลี่ยนแปลงคือ

- คำสั่ง call [addr] จะถูกแปลเป็น goto [addr]
- คำสั่ง Jump [addr] จะถูกแปลเป็น goto [addr]
- คำสั่ง JZ [addr]จะถูกแปลเป็น if pop then goto [Addr]
- คำสั่ง Ret1

คำสั่ง ret จะทำการคืนค่า machine status และจะไปประมวลผลคำสั่งถัดไป ปัญหาที่ตามมาคือตำแหน่งก่อนการเรียกโปรแกรมย่อยมีหลายตำแหน่งและในปาลกาลไม่สามารถสั่งให้ goto variable ได้ ดังนั้นถ้าต้องการให้กลับไปประมวลผลคำสั่งที่ถูกต้องตาม IP จะต้องนำเอา IP ก่อนการเรียก (call) ทุกตำแหน่งมาเพื่อตรวจสอบดังนี้

```
case CRTE.IP of
    36:    goto 36
    46:    goto 46
    144:   goto 144
end; {case}
```

2.โปรแกรมที่ได้จากการแปลรหัสกลางเป็นภาษาปาสกาล

```
{ Direct translator from icode of fib.x to pascal program }


uses crt,dos;
label 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,
label 33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,
label 63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,
label 93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,
label 116,117118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,
label 137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,
label 158,159,160,161,162,163,164,165,166,167;

type
  StackOfByte    = Array [1..4096] of byte;
  StackOfInteger = Array [1..1024] of integer;
  ProcEnvPtr     = ^RuntimeEnvRec;
  RuntimeEnvRec  = record
                     PID       : integer;
                     FP,SP,IP  : integer;
                     SS        : StackOfinteger;
                     Previos,
                     Next      : ProcEnvPtr;
                   end;

var
  CRTE     : RuntimeEnvRec;
  DS       : Array [1..4096] of integer;
  CS       : StackOfByte;
  SizeOfCS : integer;
  MaxPID   : word;
  Summary  : record
               StartTime,
               EndTime    : Real;
               ExecTime   : Real;
               NoOfInstr  : longInt;
               BlockCount : Longint;
             end;
function Time : real;
var
  h, m, s, hund : Word;
begin
  GetTime(h,m,s,hund);
  Time := h*3600.0 + m*60 + s + hund/100
End;

Procedure Push(operand : integer);
begin
  with CRTE do
  begin
    inc(SP);
    SS[SP] := operand;
  end;
end;
```

```
function Pop : integer;
begin
  with CRTE do
  begin
    pop := SS[SP];
    ss[sp] := 0;   {replace unused with 0}
    dec(SP);
  End;
end;


procedure init;
var i : integer;
begin
  clrscr;
  CRTE.FP := 1;
  CRTE.IP := 1;
  CRTE.SP := 0;
  push(-1);      {frame pointer}
  push(-1);      {Instr pointer}
  push(-1);      {stack pointer}
end;

Procedure PrintCh;
begin
  write(chr(pop));
  inc(CRTE.IP,1);
end;

Procedure Print;
begin
  write(pop,' ');
  inc(CRTE.IP,1);
end;

Procedure Jmp(operand : integer);
begin
  CRTE.IP := operand;
end;


Procedure LvalueLV(operand : integer);
begin
  with CRTE do  push( FP + operand + 3 - 1 );
  inc(CRTE.IP,3);
end;

Procedure RValueLV(operand : integer);
begin
  with  CRTE do  push( SS[FP + operand + 3 - 1 ] );
  inc(CRTE.IP,3);
end;


Procedure Literal(operand : integer);
```

```
begin
  Push(operand);
  inc(CRTE.IP,3);
end;

Procedure Func(NumParam,NumLv : integer);
var
  oldSp : integer;
begin
  with CRTE do
  begin
    if sp+50 > sizeof(stackofinteger)/2 then writeln('Stack overflow');
    OldSp := SP - 2;   {now sp pointer to old IP. skip back to recover IP and FP}
    push(oldSp-NumParam);

    move(SS[ OldSP-NumParam+1],SS[OldSP+4], NumParam * 2 ); {copy passing parameter}
    inc(sp,numLV+numParam);
    FP := OldSP + 1;  {new sp}
  end;
  inc(CRTE.IP,5);
end;

Procedure Call(CallAddr: integer);
begin
  with crte do
  begin
    push(FP);
    push(IP);
    IP := CallAddr;
  end;
end;

Procedure LE;
begin
  if pop <= pop then
    push(1)
  else
    push(0);
  inc(CRTE.IP,1);
end;

Procedure Ret0;
begin
  with CRTE do
  begin
    SP := SS[FP+2];
    IP := SS[FP+1]+3;
    FP := SS[FP];
  end;
end;

Procedure Ret1;
var
  returnVal : integer;
begin
  returnval := pop;
```

```
 with CRTE do
 begin
   SP := SS[FP+2];
   IP := SS[FP+1]+3;
   FP := SS[FP];
 end;
 push(returnVal);
end;

Procedure Minus;
begin
 push(pop-pop);
 inc(CRTE.IP,1);
end;

Procedure Plus;
begin
 push(pop+pop);
 inc(CRTE.IP,1);
end;

Procedure ExecSet;              {Exec set in Stack segmant (local val)}
begin
   CRTE.SS[pop] := pop;
 inc(CRTE.IP,1);
end;

begin
   summary.starttime := time;
   init;

   1:   call (48);   goto 48;
   4:   func (1, 0);
   9:   rvalueLV (1); literal(2); le;
   16:   {jz (26);}                                  inc(crte.ip,3);
                                                     if pop=0 then
                                                     begin
                                                           CRTE.IP :=26;
                                                           goto 26;
                                                     end;
   19:   literal(1); ret1;                           {jump routine}
                                                     case CRTE.IP of
                                                           36 : goto 36;
                                                           46 : goto 46;
                                                           144: goto 144;
                                                     end;
   23:   jmp (48);                                   goto 48;
   26:   rvalueLV (1); literal(1); minus; call (4);  goto 4;
   36:   rvalueLV (1); literal(2); minus; call (4);  goto 4;
   46:   plus; ret1;                                 {jump}
                                                     case CRTE.IP of
                                                           36 : goto 36;
                                                           46 : goto 46;
                                                           144: goto 144;
                                                     end;
```

```
 48:   func (0, 1);
 53:   lvalueLV (1); literal(1);  ExecSet;
 60:   rvalueLV (1);
       literal(23);
       le;
 67:   {jz (167);}                                  inc(crte.ip,3);
                                                    if pop=0 then
                                                    begin
                                                          CRTE.IP :=167;
                                                          goto 167;
                                                    end;

 70:   literal(70);  printCh;
 74:   literal(105); printCh;
 78:   literal(98);  printCh;
 82:   literal(111); printCh;
 86:   literal(110); printCh;
 90:   literal(97);  printCh;
 94:   literal(99);  printCh;
 98:   literal(105); printCh;
102:   literal(32);  printCh;
106:   literal(111); printCh;
110:   literal(102); printCh;
114:   literal(32);  printCh;
118:   rvalueLV(1);  print;
122:   literal(32);  printCh;
126:   literal(105); printCh;
130:   literal(115); printCh;
134:   literal(32);  printCh;
138:   rvalueLV (1); call (4);                     goto 4;
144:   print;
145:   literal(13); printCh;
149:   literal(10); printCh;
153:   lvalueLV (1); rvalueLV (1); literal(1); plus; ExecSet;
164:   jmp(60);                                    goto 60;
167:   {EOF};
summary.Endtime := time;
with summary do
begin
  writeln('Total execution time = ',EndTime - startTime:10:2,' second');
  writeln('Performance          = ',1651366 /(EndTime - startTime ):10:2 ,' instruction/sec');
end;
end.
```

3.เปรียบเทียบผลการทำงานกับอินเตอร์พรีเตอร์

เมื่อประมวลผลโปรแกรมที่ได้ทั้งสองพบว่า

ชนิด	จำนวนคำสั่งที่ประมวลผล (คำสั่ง)	เวลาในการประมวลผล (วินาที)	ความเร็ว (คำสั่ง/วินาที)
Interpreter	1,651,366	61.74	26747.10
Direct compiler	1,651,366	20.33	81228.04

โปรแกรมปาลกาลที่ได้จะมีความเร็วในการประมวผลสูงกว่าอินเตอร์พรีเตอร์ 3 เท่า

ประวัติผู้เขียน

นายสมศักดิ์ รวมมหทรัพย์ เกิดเมื่อวันที่ 9 มีนาคม พุทธศักราช 2514 ที่ตำบลพรานนก อำเภอบางกอกน้อย จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิศวกรรมศาสตร์บัณฑิต (อุตสาหการ) ภาควิชาอุตสาหการ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น ในปีการศึกษา 2535 และเข้าศึกษาต่อในหลักสูตรวิทยาศาสตร์มหาบัณฑิต ณ.ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย เมื่อปีพ.ศ.2536