More Next Blog» avishekroy3july@gmail.com Dashboard Sign Out

3rd year cse lab programs

As per the anna university regulations - 2004, cs 1356 compilers lab and cs 1355 graphics and multimedia lab programs will be available here... u can also request for prog to this mail id cse.achievers@gmail.com...will be published soon...

Contributors

kannan - admin

cselab

Blog Archive

▼ 2010 (15)
► February (3)
▼ January (12)
projection of 3d image
CODE GENERATION
cohen sutherland line clipping
bresenhams line drawing algorithm
intermediate code generation
DDA LINE Drawing Algorithm
two dimensional transformation
midpoint circle algorithm
midpoint ellipse algorithm
shift reduce parser
recursive descent parser in c
lexical analyser in c

WEDNESDAY, JANUARY 13, 2010

recursive descent parser in c

Download this file : parse.c

program:

```
#include"stdio.h"
#include"conio.h"
#include"string.h"
#include"stdlib.h"
#include"ctype.h"

char ip_sym[15],ip_ptr=0,op[50],tmp[50];
void e_prime();
void e();
void t_prime();
void t();
void f();
void advance();
int n=0;
void e()
{
 strcpy(op,"TE'");
 printf("E=%-25s",op);
 printf("E->TE'\n");
 t();
 e_prime();
}

void e_prime()
{
int i,n=0,l;
for(i=0;i<=strlen(op);i++)
   if(op[i]!='e')
 tmp[n++]=op[i];
strcpy(op,tmp);
l=strlen(op);
for(n=0;n < l && op[n]!='E';n++);
if(ip_sym[ip_ptr]=='+')
 {
    i=n+2;
 do
 {
 op[i+2]=op[i];
 i++;
 }while(i<=l);
  op[n++]='+';
  op[n++]='T';
  op[n++]='E';
  op[n++]=39;
  printf("E=%-25s",op);
```

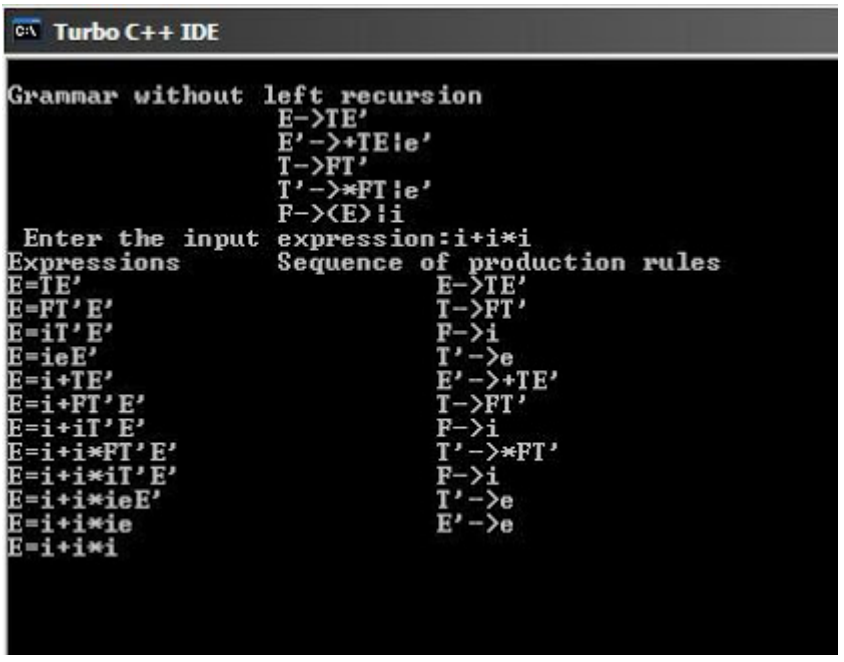
```
 printf("E'->+TE'\n");
 advance();
 t();
 e_prime();
}
else
{
   op[n]='e';
 for(i=n+1;i<=strlen(op);i++)
op[i]=op[i+1];
printf("E=%-25s",op);
printf("E'->e");
}
}
void t()
{
 int i,n=0,l;
 for(i=0;i<=strlen(op);i++)
  if(op[i]!='e')
   tmp[n++]=op[i];
 strcpy(op,tmp);
 l=strlen(op);
for(n=0;n < l && op[n]!='T';n++);

 i=n+1;
 do
 {
  op[i+2]=op[i];
  i++;
 }while(i < l);
 op[n++]='F';
 op[n++]='T';
 op[n++]=39;
 printf("E=%-25s",op);
 printf("T->FT'\n");
 f();
 t_prime();
}

void t_prime()
{
int i,n=0,l;
for(i=0;i<=strlen(op);i++)
   if(op[i]!='e')
 tmp[n++]=op[i];
strcpy(op,tmp);
l=strlen(op);
for(n=0;n < l && op[n]!='T';n++);
if(ip_sym[ip_ptr]=='*')
{
   i=n+2;
 do
 {
 op[i+2]=op[i];
 i++;
 }while(i < l);
  op[n++]='*';
  op[n++]='F';
  op[n++]='T';
  op[n++]=39;
  printf("E=%-25s",op);
  printf("T'->*FT'\n");
  advance();
  f();
  t_prime();
 }
 else
 {
   op[n]='e';
```

```
 for(i=n+1;i<=strlen(op);i++)
op[i]=op[i+1];
printf("E=%-25s",op);
printf("T'->e\n");
}
}

void f()
{
int i,n=0,l;
for(i=0;i<=strlen(op);i++)
   if(op[i]!='e')
tmp[n++]=op[i];
strcpy(op,tmp);
l=strlen(op);
for(n=0;n < l && op[n]!='F';n++);
if((ip_sym[ip_ptr]=='i')||(ip_sym[ip_ptr]=='I'))
{
op[n]='i';
printf("E=%-25s",op);
printf("F->i\n");
advance();
}
else
{
 if(ip_sym[ip_ptr]=='(')
 {
  advance();
  e();
  if(ip_sym[ip_ptr]==')')
  {
   advance();
    i=n+2;
do
{
op[i+2]=op[i];
i++;
}while(i<=l);
 op[n++]='(';
 op[n++]='E';
 op[n++]=')';
 printf("E=%-25s",op);
 printf("F->(E)\n");
  }
 }
 else
 {
  printf("\n\t syntax error");
  getch();
  exit(1);
 }
}
}

void advance()
{
ip_ptr++;
}

void main()
{
 int i;
 clrscr();
 printf("\nGrammar without left recursion");
 printf("\n\t\t E->TE' \n\t\t E'->+TE'|e \n\t\t T->FT' ");
 printf("\n\t\t T'->*FT'|e \n\t\t F->(E)|i");
 printf("\n Enter the input expression:");
 gets(ip_sym);
 printf("Expressions");
```

```
 printf("\t Sequence of production rules\n");
  e();
  for(i=0;i < strlen(ip_sym);i++)
 {
  if(ip_sym[i]!='+'&&ip_sym[i]!='*'&&ip_sym[i]!='('&&
     ip_sym[i]!=')'&&ip_sym[i]!='i'&&ip_sym[i]!='I')
  {
   printf("\nSyntax error");
   break;
  }
  for(i=0;i<=strlen(op);i++)
    if(op[i]!='e')
 tmp[n++]=op[i];
    strcpy(op,tmp);
    printf("\nE=%-25s",op);
 }
 getch();
}
```

Download this file : parse.c

Output:



Download this file : parse.c

Posted by cselab at 5:29 PM
Labels: compiler design lab , cs 1356 , recursive descent parser , recursive descent parser in c

Newer Post Home Older Post

Subscribe to: Post Comments (Atom)