

# ETAPA7 - Trabalho Prático de Compiladores - Giovani Da Silva

## a.Implementação dos tratamentos de erros:

Todos os tratamentos de erros foram chamados por meio da função `syntax_error`, que pega a linha do erro e uma mensagem informando onde está o erro.

```
void syntax_error(char* error_message){  
    fprintf(stderr, "Syntax error at line %d: %s\n", getLineNumber(), error_message);  
    return;  
}
```

### Alguns exemplos de tratamentos de erros

```
| identifier '=' error      { $$ = 0; syntax_error("missing right operand expression"); }
```

```
data{  
    int:i=0;  
}  
char:main(){  
    i=  
    until(i==10){  
        i=i+2;  
    };  
}
```

```
giovani@DESKTOP-1AR1513: /mnt/c/Users/Cliente/Documents/compiladores/compilers/et  
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compilado  
Syntax error at line 6: missing right operand expression  
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compilado
```

```
data{  
    int:i=0;  
}  
char:main(){  
    print;  
    until(i==10){  
        i=i+2;  
    };  
}
```

```
giovani@DESKTOP-1AR1513: /mnt/c/Users/Cliente/Documents/compiladores/compilers/et..  
apa7$ ./etapa7 in.txt out.txtmake
```

```
Syntax error at line 5: missing expression after print
```

```
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compilador  
apa7$ giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/comp  
ers/etapa7giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/  
mpilers/etangiovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Document
```

```

data{
    int:i=0;
}
char:main(){
    if({});
    until(i==10){
        i=i+2;
    };
}

```

```

-o etapa7
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compiladores,
etapa7$ ./etapa7 in.txt out.txtmake

```

Syntax error at line 5: missing logical expression

```

giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compiladores,
etapa7$ _

```

```

data{
    int:i=0;
}
char:main(){

    until(i==10){
        i=i+2
    };
}

```

```

giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents...
adores/compilers/etapa7$ ./etapa7 in.txt out.tx

```

Syntax error at line 8: missing ;

```

giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Dc
adores/compilers/etapa7$

```

```

data{
    int:i=0;
}
char:main(){

    until(i==10){
        i=i+2;
    };
}

```

```

adores/compilers/etapa7$ ./etapa7 in.txt out.t

```

Syntax error at line 8: missing }

```

giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/D
adores/compilers/etapa7$ _

```

## b.Otimização

A técnica escolhida para otimização foi a LoopUnrolling. Foi feito sobre o loop do nosso compilador, o until, serve apenas para caso em que se tem um until[<,>,==...] x e dentro do until temos x = x+ y;

## Implementação

A função can\_optimize\_until faz a verificação, se esse Until pode ser otimizado ou não

```
int can_optimize_until(AST *node)
{
    if (node->son[1])
    {
        if (node->son[1]->son[0])
        {
            if (node->son[1]->son[0]->son[0])
            {
                if (node->son[1]->son[0]->son[0]->son[1])
                {
                    if (node->son[1]->son[0]->son[0]->son[1]->type == AST_ADD)
                        return 1;
                    else
                        return 0;
                }
            }
        }
    }
    return 0;
}
```

Exemplos:

Exemplo 1 : Antes do loop unroll

```
data{
    int:i=0;
}
char:main(){
    until(i==10){
        i=i+1;
    };
}

adores/compilers/etapa7$ ./etapa7 in.txt out.txtmak
TAC(TAC_DECL_VAR, i, 0, 0)
TAC(TAC_BEGINFUN, main, 0, 0)
TAC(TAC_LABEL, mYLabule_0, 0, 0)
TAC(TAC_EQ, mYWeeirT_emp0, i, 10)
TAC(TAC_IF, mYLabule_1, mYWeeirT_emp0, 0)
TAC(TAC_ADD, mYWeeirT_emp1, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp1, 0)
TAC(TAC_JUMP, mYLabule_0, 0, 0)
TAC(TAC_LABEL, mYLabule_1, 0, 0)
TAC(TAC_ENDFUN, main, 0, 0)
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Docume
adores/compilers/etapa7$ _
```

### Exemplo 1: Depois do loop unroll

```
data{
    int:i=0;
}
char:main(){
    until(i==10){
        i=i+1;
    };
}

TAC(TAC_DECL_VAR, i, 0, 0)
TAC(TAC_BEGINFUN, main, 0, 0)
TAC(TAC_ADD, mYWeeirT_emp12, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp12, 0)
TAC(TAC_ADD, mYWeeirT_emp11, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp11, 0)
TAC(TAC_ADD, mYWeeirT_emp10, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp10, 0)
TAC(TAC_ADD, mYWeeirT_emp9, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp9, 0)
TAC(TAC_ADD, mYWeeirT_emp8, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp8, 0)
TAC(TAC_ADD, mYWeeirT_emp7, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp7, 0)
TAC(TAC_ADD, mYWeeirT_emp6, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp6, 0)
TAC(TAC_ADD, mYWeeirT_emp5, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp5, 0)
TAC(TAC_ADD, mYWeeirT_emp4, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp4, 0)
TAC(TAC_ADD, mYWeeirT_emp3, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp3, 0)
TAC(TAC_ADD, mYWeeirT_emp2, i, 1)
TAC(TAC_MOVE, i, mYWeeirT_emp2, 0)
TAC(TAC_ENDEFUN, main, 0, 0)
```

Outro exemplo:

Antes

```
data{
    int:i=0;
}
char:main(){
    until(i<50){
        i=i+12;
    };
}

gcc main.o lex.yy.o y.tab.o hash.o ast.o decompiler.o semant
ic.o tacs.o asm.o -o etapa7
giovani@DESKTOP-1AR1513:/mnt/c/Users/Cliente/Documents/compil
adores/compilers/etapa7$ ./etapa7 in.txt out.txtmake
TAC(TAC_DECL_VAR, i, 0, 0)
TAC(TAC_BEGINFUN, main, 0, 0)
TAC(TAC_LABEL, mYLabule_0, 0, 0)
TAC(TAC_LES, mYWeeirT_emp0, i, 50)
TAC(TAC_IF, mYLabule_1, mYWeeirT_emp0, 0)
TAC(TAC_ADD, mYWeeirT_emp1, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp1, 0)
TAC(TAC_JUMP, mYLabule_0, 0, 0)
TAC(TAC_LABEL, mYLabule_1, 0, 0)
```

Depois

```
data{
    int:i=0;
}
char:main(){
    until(i<50){
        i=i+12;
    };
}
TAC(TAC_DECL_VAR, i, 0, 0)
TAC(TAC_BEGINFUN, main, 0, 0)
TAC(TAC_ADD, mYWeeirT_emp6, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp6, 0)
TAC(TAC_ADD, mYWeeirT_emp5, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp5, 0)
TAC(TAC_ADD, mYWeeirT_emp4, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp4, 0)
TAC(TAC_ADD, mYWeeirT_emp3, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp3, 0)
TAC(TAC_ADD, mYWeeirT_emp2, i, 12)
TAC(TAC_MOVE, i, mYWeeirT_emp2, 0)
TAC(TAC_ENDFUN, main, 0, 0)
giovani@DESKTOP-1AR1513:/mnt/c/Users/Clier
adores/compilers/etapa7$
```