PicoC-Compiler

Übersetzung einer Untermenge von C in den Befehlssatz der RETI-CPU

Kolloquiumspräsentation

Präsentator: Jürgen Mattheis

Gutachter: Prof. Dr. Scholl Betreuung: M.Sc. Seufert

28. September 2022

Universität Freiburg, Lehrstuhl für Betriebssysteme

Code Generierung

Codebeispiel

- ▶ im Appendix ab Folie ?? sind Tokens, Ableitungsbaum, Vereinfachter Ableitungsbaum, Abstrakter Syntaxbaum und die modifzierten Abstrakten Synxtaxbäume der verschiedenen Passes an einem Codebeispiel erklärt.
- make test TESTNAME=example_presentation VERBOSE=-v

2 / 22

Code Generierung

Zugriff auf Zusammengesetzte Datentypen

```
// in:1
// expected:42
// datasegment:36
struct stt {int attr1: int attr2[2]:}:
struct stt ar_of_sts[3][2];
int fun(struct stt (*param)[3][2]){
 (*(*param+2))[1].attr2[input()] = 42;
 return 1:
void main() {
 struct stt (*pntr_on_ar_of_sts)[3][2] = &ar_of_sts;
 int res = fun(pntr_on_ar_of_sts);
 if (res) {
   print((*(*pntr_on_ar_of_sts+2))[1].attr2[1]);
```

Datentyp der Variable

```
// ...

struct stt

→ (*pntr_on_ar_of_sts)[3][2] =

→ &ar_of_sts;

// ...
```

- ein "Zeiger auf ein Feld der Mächtigkeit 3 von Feldern der Mächtigkeit 2 von Verbunden des Typs stt".
- ► Clockwise/Spiral Rule

Code Generierung

Codebeispiel

```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1]
                                                       Start
                                                             nichts momentan relevantes auf dem Stack
           Variable vom Stackframe
# Exp(Stackframe(Num('addr')))
                                                                       # Exp(Global(Num('addr')))
SUBI SP 1:
                                                                       SUBI SP 1:
LOADIN BAF ACC -(2 + addr):
                                                                       LOADIN DS ACC addr:
STOREIN SP ACC 1:
                                                                       STOREIN SP ACC 1:
                                                             Startadresse der Variable auf dem Stack
```

Jürgen Mattheis PicoC-Compiler Universität Freiburg 4 / 22

```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1]
```

```
struct stt (*pntr_on_ar_of_sts)[3][2]...
```

Mittelteil

Feldindexzugriff auf Feld

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2')),

→ Stack(Num('1')))
LOADIN SP IN1 2;
LOADIN SP IN2 1;
MULTI IN2 (∏<sub>j-1</sub>, din<sub>j</sub>)-size(datatype);
ADD IN1 IN2;
ADD IN1 IN2;
STOREIN SP IN1 1;
```

Feldindexzugriff auf Zeiger

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2'))),

→ Stack(Num('1'))))
LOADIN SP IN2 2;
LOADIN IN2 IN1 0;
LOADIN SP IN2 1;
MULTI IN2 (∏<sub>int,i</sub> din<sub>j</sub>). size(datatype);
ADD IN1 IN2;
ADDI SP 1;
STOREIN SP IN1 1;
```

Verbundsattribut-zugriff auf Verbund

5 / 22

```
# Ref(Attr(Stack(Num('1')), \hookrightarrow Name('attr')))
LOADIN SP IN1 1;
ADDI IN1 \sum_{t=1}^{Ld_{x}-1} stze(datatype_{1,x});
STOREIN SP IN1 1;
```

*1: Startadresse eines Zeigerelementes, Feldelementes





```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1]
```

```
struct stt (*pntr_on_ar_of_sts)[3][2]...
```

Mittelteil

Feldindexzugriff auf Feld

```
# z.B. Exp(Num('idx'))

SUBI SP 1;

LOADI ACC idx;

STOREIN SP ACC 1;

# Ref(Subscr(Stack(Num('2')),

→ Stack(Num('1'))))

LOADIN SP IN1 2;

LOADIN SP IN2 1;

MULTI IN2 (∏<sub>-++1</sub> dim<sub>1</sub>)-size(datatype);

ADDI SP 1;

STOREIN SP IN1 1;
```

Feldindexzugriff auf Zeiger

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2'))),

Stack(Num('1'))))
LOADIN SP IN2 2;
LOADIN IN2 IN1 0;
LOADIN SP IN2 1;
MULTI IN2 (II)—r, dinj) - size(datatype);
ADD IN1 IN2;
ADD IN1 IN2;
ADD ISP 1;
STOREIN SP IN1 1;
```

Verbundsattribut-zugriff auf Verbund

```
# Ref(Attr(Stack(Num('1')), \hookrightarrow Name('attr')))
LOADIN SP IN1 ;
ADDI IN1 \sum_{k=1}^{id_{x}-1} slxe(datatype_{1,x});
STOREIN SP IN1 1;
```











```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1]
```

```
struct stt (*pntr_on_ar_of_sts)[3][2]...
```

Mittelteil

Feldindexzugriff auf Feld

Feldindexzugriff auf Zeiger

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2'))),

Stack(Num('1'))))
LOADIN SP IN2 2;
LOADIN IN2 IN1 0;
LOADIN SP IN2 1;
MULTI IN2 (II)—r, dinj) - size(datatype);
ADD IN1 IN2;
ADD IN1 IN2;
ADD ISP 1;
STOREIN SP IN1 1;
```

Verbundsattribut-zugriff auf Verbund

```
# Ref(Attr(Stack(Num('1')), \hookrightarrow Name('attr')))
LOADIN SP IN1 1;
ADDI IN1 \sum_{k=1}^{n(d_k-1)} size(datatype_{1,k});
STOREIN SP IN1 1;
```







```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1]
```

```
struct stt (*pntr_on_ar_of_sts)[3][2]...
```

Mittelteil

Feldindexzugriff auf Feld

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2')),

→ Stack(Num('1')))
LOADIN SP IN1 2;
LOADIN SP IN2 1;
MULTI IN2 (∏<sub>j-++</sub>, din<sub>j</sub>)-size(datatypa);
ADD IN1 IN2;
ADDI SP 1;
STOREIN SP IN1 1;
```

Feldindexzugriff auf Zeiger

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2'))),

←> Stack(Num('1'))))
LOADIN SP IN2 2;
LOADIN IN2 IN1 0;
LOADIN SP IN2 1;
MULTI IN2 (∏—++, din<sub>j</sub>)-size(datatype);
ADD IN1 IN2;
ADDI SP 1;
STOREIN SP IN1 1;
```

Verbundsattribut-zugriff auf Verbund

```
\label{eq:continuous_section} \begin{split} & \text{\# Ref(Attr(Stack(Num('1'))),} \\ & \hookrightarrow \text{Name('attr')))} \\ & \text{LOADIN SP IM1 1;} \\ & \text{ADDI IN1 } \sum_{k=1}^{kd_{k}-1} slaze(\text{datatype}_{0,k}); \\ & \text{STOREIN SP IN1 1;} \end{split}
```

8 / 22

*1: Startadresse eines Zeigerelementes, Feldelementes





```
(*(*pntr_on_ar_of_sts+2))[1].attr2[1], struct stt {int attr1; int attr2[2];};

Startadresse der Variable auf dem Stack

Mittelteil
```

Feldindexzugriff auf Feld

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2')),

← Stack(Num('1'))))
LOADIN SP IN1 2;
LOADIN SP IN2 1;
MULTI IN2 (∏,, dim,) - size(datatype);
ADD IN1 IN2;
ADDI SP 1;
STOREIN SP IN1 1;
```

Feldindexzugriff auf Zeiger

```
# z.B. Exp(Num('idx'))
SUBI SP 1;
LOADI ACC idx;
STOREIN SP ACC 1;
# Ref(Subscr(Stack(Num('2')),
→ Stack(Num('1'))))
LOADIN SP IN2 2;
LOADIN IN2 IN1 0;
LOADIN SP IN2 1;
MULTI IN2 (∏<sub>intr</sub> din<sub>j</sub>) size(datatype);
ADD IN1 IN2;
ADDI SP 1;
STOREIN SP IN1 1;
```

Verbundsattribut-zugriff auf Verbund

```
# Ref(Attr(Stack(Num('1')), \hookrightarrow Name('attr')))
LOADIN SP IN1 1;
ADDI IN1 \sum_{k=1}^{id_{k-1}} size(datatype_{k,k});
STOREIN SP IN1 1;
```



9 / 22







```
struct stt {int attr1; int attr2[2];};
 (*(*pntr_on_ar_of_sts+2))[1].attr2[1],
                                                        Startadresse eines Zeigerelementes, Feldelementes
                                                                 oder Verbundsattributes auf dem Stack
                                                    Schlussteil
        Letzter Datentyp ist Verbund,
                                                                                  Letzter Datentyp ist Feld
          Zeiger oder Basisdatentyp
                                                                       # not included Exp(Stack(Num('1')))
# Exp(Stack(Num('1')))
LOADIN SP IN1 1:
LOADIN IN1 ACC 0;
STOREIN SP ACC 1:
                                                       Inhalt der Speicherzelle an der berechneten
                                                             Adresse oder die berechnete Adresse selbst
                                                       Ende
```

Fehlermeldungen

Kategorien

- ▶ UnexpectedCharacter
- ▶ UnexpectedToken
- ▶ UnexpectedEOF
- DivisionByZero
- ▶ UnknownIdentifier
- ▶ UnknownAttribute
- ► ReDeclarationOrDefinition
- ► TooLargeLiteral

- ► NoMainFunction
- ConstAssign
- DatatypeMismatch
- PrototypeMismatch
- ArgumentMismatch
- ▶ WrongNumberArguments
- ▶ WrongReturnType
- ▶ im Appendix ab Folie 3 mit Erklärung.

Literatur



20 / 22

Vorlesungen



21 / 22

Online



Clockwise/Spiral Rule. URL:

https://c-faq.com/decl/spiral.anderson.html (besucht am 29.07.2022).