

# Flex

# Flex

## fast lexical analyzer generator

Vortrag von  
Thomas Weise  
28.05.2002

im Rahmen des  
Proseminar Softwarewerkzeuge

# Gliederung

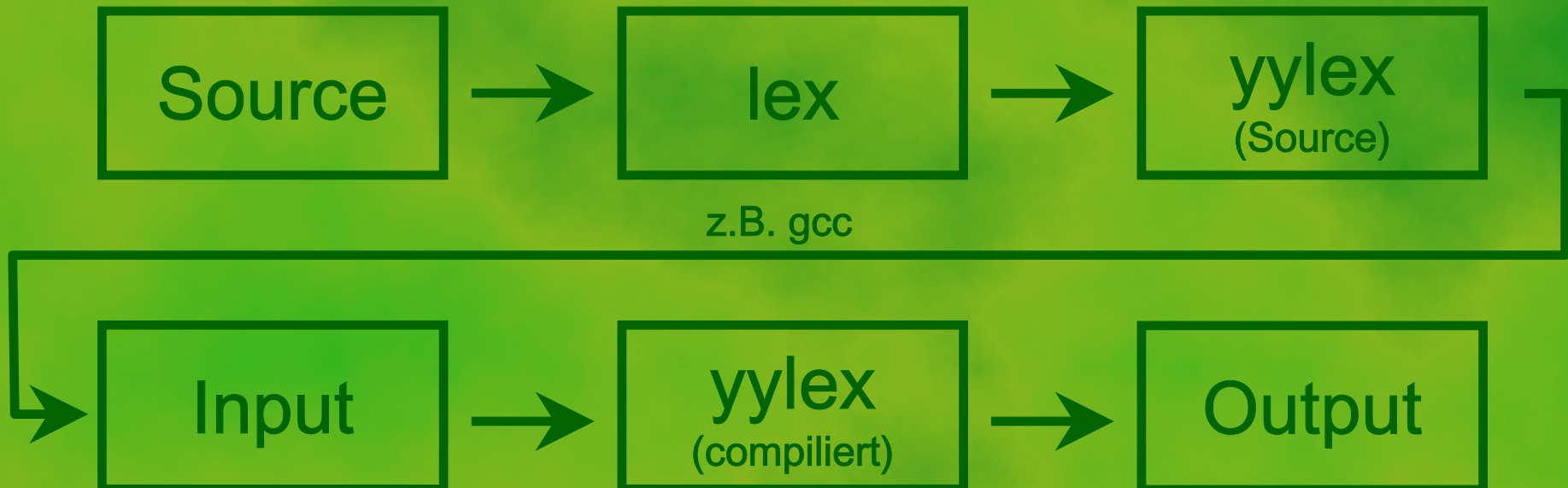
# Flex

- ★ Was ist Flex?
- ★ Eingabedaten?
  - ★ Definitionen
  - ★ Regeln
    - ★ reguläre Ausdrücke
    - ★ Aktionen
  - ★ benutzerdefinierter Code
- ★ kleines Beispiel
- ★ Quellen
- ★ Ausblick

# Was ist Flex?

# Flex

- Flex ≡ Tool um Scanner zu erzeugen
- Scanner ≡ erkennt Muster in Eingabedaten  
(und führt spezifische Aktionen aus)



# Eingabedaten?

Definitionen

%%

Regeln der Form

*{regulärer Ausdruck} {Aktion}*

%%

benutzerdefinierter Code

Beispiel 1

```
%%  
[ \t]+$ ;
```

löscht alle Leerzeichen/  
Tabstopps am Ende  
der Zeilen

## Definitionen

%%

Regeln der Form

*{regulärer Ausdruck} {Aktion}*

%%

benutzerdefinierter Code

## Beispiel 2

<i>D</i>	[0-9]
<i>E</i>	[Ee][-+]?{D}+
%%	
{D}+	printf("integer");
{D}+“.”{D}*({E})?	
{D}*“.”{D}+({E})?	
{D}+{E}	printf("real");

- ★ Sourcecode in Zielprogrammiersprache oder
- ★ Konstante-Wert Paare
- ★ Wert ist regulärer Ausdruck
- ★ Konstante wird zu Wert „expandiert“
- ★ Konstantenname bei Referenz in {}

erkennt ganze und Fließpunktzahlen.

Definitionen

%%

Regeln der Form

*{regulärer Ausdruck} {Aktion}*

%%

benutzerdefinierter Code

- ★ Regel  $\equiv$  regulärer Ausdruck Leerzeichen Aktion
- ★ Input wird nach Regeln gescannt
- ★ immer längster passender Text in Variable yyText gespeichert
- ★ dann zur Regel gehörende Aktion ausgeführt
- ★ Text, der zu keiner Regel passt wird in Output kopiert
- ★ ist keine Aktion zum Ausdruck definiert, wird yyText verworfen, nicht zum Output kopiert

## Definitionen

%%

## Regeln der Form

*{regulärer Ausdruck}* *{Aktion}*

%%

benutzerdefinierter Code

[aj-oZ] Klasse: a, j..o oder Z

r\* 0 oder mehr r

r? kein oder ein r

r{2,5} zwischen 2 und fünf r's

r{4} 4 r's

rs r gefolgt von s

^r r am Zeilenanfang

\x escaped Operator (z.B. \*)

<s>r r in Startkondition s

x Zeichenkette x

. jedes Zeichen außer „Neue Zeile“

[xyz] Klasse: x, y oder z

[^A-Z] negierte Klasse: alles außer A..Z

r+ ein oder mehr r

r{2,} zwei oder mehr r's

r|s r oder s

r/“xyz“ r, aber nur vor xyz

r\$ r am Zeilenende

\123 Octalwert eines Zeichens

(r) beliebige Klammerung

## Definitionen

%%

## Regeln der Form

*{regulärer Ausdruck}* **{Aktion}**

%%

benutzerdefinierter Code

beliebiger C-Code,  
Klammerungen {} werden  
zusammengehalten und richtig  
erkannt

	selbe Aktion wie bei nächster Regel
Begin(x)	in Startkondition x wechseln
ECHO	yyText an Output ausgeben
Reject	„zweitbeste“ Regel anwenden
input(c)	Zeichen aus InPut in c lesen
unput(c)	Zeichen c vorne in InPut schreiben
output(c)	Zeichen c in OutPut schreiben
yymore()	Text nächsten Tokens an aktuelles Token anhängen
yylless(x)	fügt alle bis auf die letzten n Zeichen des Tokens in Input ein.

Definitionen

%%

Regeln der Form

*{regulärer Ausdruck} {Aktion}*

%%

benutzerdefinierter Code

- ★ Sourcecode in Zielprogrammiersprache
- ★ ersetzt den standardmäßigen Hauptprogrammsource von Flex
- ★ sollte die Routine `yylex()` aufrufen, die das eigentliche Scannen erledigt

# kleines Beispiel

```
lex -obeispiel3.c beispiel3.l
```

```
gcc beispiel3.c -o beispiel3 -lflex
```

```
%{int charCount = 0, wordCount = 0, lineCount = 0; %}  
word [^ \t\n]+  
%%  
{word} {wordCount++; charCount += yyleng; }  
[\n] {charCount++; lineCount++; }  
. {charCount++; }  
%%  
main() {  
yylex();  
printf("Characters: %d Words: %d Lines %d\n",  
charCount, wordCount, lineCount); }
```

# Quellen

# Flex

- ★ Flex, version 2.5 - a scanner generator  
[tinfpc2.vub.ac.be/manuals/flex.html](http://tinfpc2.vub.ac.be/manuals/flex.html)
- ★ Flex - fast lexical analyzer generator  
[dinosaur.compilertools.net/flex/manpage.html](http://dinosaur.compilertools.net/flex/manpage.html)
- ★ A little manual for Lex and Yacc.htm  
[www.geocities.com/SiliconValley/Campus/3754/litl0.htm](http://www.geocities.com/SiliconValley/Campus/3754/litl0.htm)
- ★ C Programming Tools  
Hewlett-Packard

# Ausblick



- ★ Kommandozeilenparameter, Startkonditionen und interne Abläufe ausgespart
  - ★ wenig sinnvolle Einzelanwendungen
  - ★ richtige Vorzüge erst zusammen mit yacc o.ä.
- 
- ★ Beantwortung von Fragen
  - ★ [tweise@gmx.de](mailto:tweise@gmx.de)      [www.tu-chemnitz.de/~weist](http://www.tu-chemnitz.de/~weist)