

definition





## FLEX实例

下面通过一个实例来具体展示flex的使用方式，主要功能是扫描并匹配文件中的字符串，并回显其类型和内容，代码如下：

%{

#include "stdio.h"

#include "stdlib.h"

%}

INT\_DEX [1-9][0-9]\*|[0]

INT\_HEX [0][Xx]([1-9][0-9]\*|[0])

INT\_OCT [0][0-7]

FLOAT [0-9]\*[.][0-9]+([eE][+-]?[0-9]\*|[0])?f?

SEMI [;]

COMMA [,]

ASSIGNOP [=]

RELOP [>]|[<]|[>][=]|[<][=]|[=][=]|[!][=](^[=])

PLUS [+]

MINUS [-]

STAR [\*]

DIV [/]

AND [&][&]

OR [|][|]

DOT [.]

NOT [!]

TYPE int|float

LP \(

RP \)

LB \[

RB \]

LC \{

RC \}

STRUCT struct

RETURN return

IF if

ELSE else

WHILE while

SPACE [ \n\t]

ID [a-zA-Z\_][a-zA-Z\_0-9]\*

/\*end of definition\*/

%%

{SEMI} {

printf("get semmi : %s\n", yytext);

}

{COMMA} {

printf("get comma : %s\n", yytext);

}

{ASSIGNOP} {

printf("get assignop : %s\n", yytext);

}

{INT\_DEX} |

{INT\_HEX} |

{INT\_OCT} {

printf("get an integer: %s\n", yytext);

}

{FLOAT} {

printf("get a float: %s\n", yytext);

}

{PLUS} |

{MINUS} |

{DIV} |

{STAR} {

printf("get an operator: %s\n", yytext);

}

{RELOP} {

printf("get a relop: %s\n", yytext);

}

{AND} |

{OR} |

{NOT} {

printf("get a logic operator: %s\n", yytext);

}

{DOT} {

printf("get a dot: %s\n", yytext);

}

{STRUCT} |

{RETURN} |

{IF} |

{ELSE} |

{WHILE} {

printf("get keyword: %s\n", yytext);

}

{TYPE} {

printf("get type: %s\n", yytext);

}

{LP} |

{RP} |

{LB} |

{RB} |

{LC} |

{RC} {

printf("get brackets : %s\n", yytext);

}

{SPACE} |

. {

/\*ABANDON THESE CHARACTORS\*/

}

{ID} {

printf("get an ID: %s\n", yytext);

}

%%

int yywrap() {

return 1;

}

int main(int argc, char\*\* argv) {

if (argc > 1) {

if (!(yyin = fopen(argv[1], "r"))) {

perror(argv[1]);

return 1;

}

}

while (yylex());

return 0;