Para ejecutar las prácticas, los distintos sistemas Unix compilan y ejecutan yacc y lex de distinta forma:

- ✓ **Lisisu02.usal.es (Silicon Graphics)**

```
yacc -d calculadora.yacc
lex calculadora.lex
gcc -o calculadora y.tab.c lex.yy.c -ly -ll
calculadora
```

- ✓ **Encina.usal.es (Sun / Solaris 8)**

```
yacc -d calculadora.yacc
lex calculadora.lex
cc -o calculadora y.tab.c lex.yy.c -ly -ll
calculadora
```

- ✓ **Tejo.usal.es (HPUX)**

```
yacc -d calculadora.yacc
lex calculadora.lex
cc -o calculadora y.tab.c lex.yy.c -ly -ll
calculadora
```

- ✓ **Lisipc (Linux RedHat 7.2)**

```
yacc -d calculadora.yacc
lex calculadora.lex
gcc -o calculadora y.tab.c lex.yy.c -ll
calculadora
```

El código lex, es el mismo en todos los casos:

```
%{
#include "y.tab.h"
extern int yylval;
%}

%%

[0-9]+      {yylval=atoi(yytext); return NUMERO;}
[ \t] ;
\n    return 0;
.     return yytext[0];

%%
```

En cuanto al código yacc, para encina y tejo, es el siguiente:

```
%token NOMBRE NUMERO

%%
instruccion: NOMBRE '=' expresion
        | expresion {printf("=%d\n",$1);}
        ;
expresion: expresion '+' NUMERO { $$ = $1 + $3;}
        | expresion '-' NUMERO { $$ = $1 - $3;}
        | NUMERO { $$ = $1;}
        ;

%%
```

En lisisu es necesario añadir el main con el <stdio.h> al principio. El resultado es el siguiente:

```
%{
      #include <stdio.h>
%}

%token NOMBRE NUMERO


%%
instruccion: NOMBRE '=' expresion
      | expresion {printf("=%d\n",$1);}
      ;
expresion: expresion '+' NUMERO { $$ = $1 + $3;}
      | expresion '-' NUMERO { $$ = $1 - $3;}
      | NUMERO { $$ = $1;}
      ;

%%

main(){ yyparse(); yylex(); }
```

Por último, en lisipc (linux) habría que añadir la función yyerror para que funcionara de modo correcto.

```
%{
      #include <stdio.h>
      yyerror(char *s);
%}

%token NOMBRE NUMERO


%%
instruccion: NOMBRE '=' expresion
      | expresion {printf("=%d\n",$1);}
      ;
expresion: expresion '+' NUMERO { $$ = $1 + $3;}
      | expresion '-' NUMERO { $$ = $1 - $3;}
      | NUMERO { $$ = $1;}
      ;

%%

main(){ yyparse(); yylex(); }

yyerror(char *s) { fprintf(stderr, "%s\n", s); }
```