

EA872 Laboratório de Programação de Software Básico

Atividade 2

8. Atividades Práticas

Para entender melhor as ferramentas `lex` (ou `flex`) e `yacc` (ou `bison`), estude os programas abaixo, fazendo referência sempre que necessário às páginas de manual online do Unix.

Programando com LEX

Os códigos a seguir estão disponíveis no arquivo zip fornecido junto com o roteiro e devem ser compilados com o comando `flex p_*.l`, seguido do comando `gcc lex.yy.c -lfl -o p_*` (* deve ser a, b, c, ...).

Atividades durante a aula

As atividades (a) e (b) devem ser feitas e entregues até o final da aula.

(a) (1 pt.) p_a.l (exemplo de execução: `./p_a < p_a.l`)

```
%%
[a-z]  printf("%c", yytext[0]-( 'a'-'A' ));
.      ECHO;
```

(a.1) Explique por que é necessário usar o “ponto e barra `./`” antes do nome do executável produzido e porque o mesmo não é necessário quando se executa o `gcc`, o `flex` ou o `bison`.

(a.2) Mostre uma maneira de se configurar seu shell que permita a execução dos executáveis compilados sem a digitação de `“./”`.

(a.3) Teste o programa de análise léxica executável gerado a partir de `p_a.l` com diferentes arquivos de entrada e explique como ele faz para produzir a saída visualizada.

(b) (1 pt.) p_b.l (exemplo de execução: `date | ./p_b`)

```
%{
    int h;
}%
AM    [ ] (00|01|02|03|04|05|06|07|08|09|10|11) [:]
PM    [ ] (12|13|14|15|16|17|18|19|20|21|22|23) [:]
%%
{PM}  {sscanf(yytext, " %d", &h);
      if (h == 12) printf(" PM 12:");
      else printf(" PM %02d:", h-12);}

{AM}  {sscanf(yytext, " %d", &h);
      if (h==0) printf(" AM 12:");
      else printf(" AM %02d:", h);}

.      ECHO;
```

(b.1) Explique como o programa gerado por `p_b.l` consegue produzir os resultados visualizados. Documente alguns casos.

(b.2) Explique o funcionamento do comando pipe (`|`) usado entre `date` e `p_b`.

(b.3) Proponha, teste e documente uma outra forma de repassar o resultado do comando `date` para `p_b`.