

# Pretty-Output

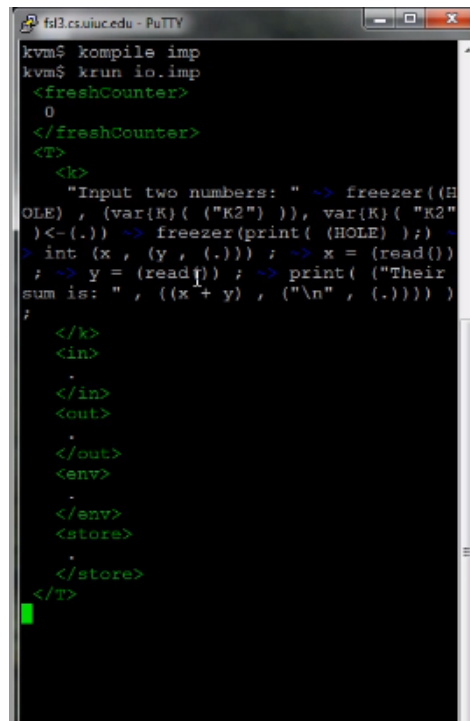
Palade Valentin, Grupa B4

## 1 Ce vrem sa facem?

Obiectivul acestui proiect il constituie realizarea unei unelte pentru framework-ul K, numita Pretty-Output, ce va permite infrumusetarea modului in care arata output-urile programelor realizate in K.

Aceasta unealta va prelua output-ul oricarui program rulat cu ajutorul framework-ului K si il va preda analizatorului lexical, numit Lex. Prin intermediul acestui ultim program output-ul va fi modificat astfel incat aspectul textului sau sa devina mai atractiv.

Un exemplu de output atractiv poate fi observat in urmatoarea imagine:



```
fs13.cs.uiuc.edu - PuTTY
kvm$ kcompile imp
kvm$ krun io.imp
<freshCounter>
0
</freshCounter>
<T>
  <k>
    "Input two numbers: " ~> freezer((HOLE) , (var{K}("K2") ), var{K}("K2") )<-(.) ~> freezer(print( (HOLE) );) ~>
    int (x , (y , (.) ) ) ; ~> x = (read())
    ; ~> y = (read()) ; ~> print( ("Their sum is: " , ((x + y) , ("\n" , (.) ) ) )
  ;
  </k>
  <in>
    .
  </in>
  <out>
    .
  </out>
  <env>
    .
  </env>
  <store>
    .
  </store>
</T>
```

## 2 Cum vrem sa facem?

Va fi folosita consola sistemului de operare Linux pentru a trimite output-ul, programelor realizate in K, direct catre analizatorului lexical, Lex.

Prin comenzile specifice ale acestui analizator va fi modificat aspectului textului (output-ului) primit din programele k.

Unele dintre aceste modificari vor fi colorarea in mod unic a fiecarei celule si repositionarea acestora si a continutului lor.

Un exemplu de cod scris in Lex:

```
3  %}  
4  cifra [0-9]  
5  nr {cifra}+  
6  litera [a-zA-Z]  
7  cuvant {litera}+  
8  celula {nr}*{cuvant}*  
9  %%  
10 \<{celula}\>|\<\/{celula}\> printf("%s\n",yytext);  
11 .|\n ;
```

Tot prin intermediul consolei din Linux, output-ul final, modificat in Lex, va fi afisat.

Structura implementarii uneltei Pretty-Output poate fi observata in imaginea urmatoare:

