Programovací jazyk Slang

Adam Suchý, 4.E

https://s.bain.cz/2024-slang

```
extern fn printf (&char, ...) -> i32;

stream main (i32, &&char) -> i32 {
  printf("Hello World!\n");
  0 as i32 | out;
}
```

```
extern fn printf (&char, ...) -> i32;

stream nums () -> i64 {
  let a = 0;
  while true { (a = a + 1) | out; }
}
stream fizzbuzz i64 -> (&char, i64) {
  let i = catch in;
  let c: &char;
  if i % 15 == 0 { c = "FizzBuzz\n"; }
  else if i % 3 == 0 { c = "Fizz\n"; }
  else if i % 5 == 0 { c = "Buzz\n"; }
  else { c = "%ld\n"; }
  (c, i) | out;
}
stream print (&char, i64) -> () {
  let i = catch in;
  printf(i.0, i.1);
}

stream main (i32, &&char) -> i32 {
  catch () | nums | fizzbuzz | print;
}
```

```
1
2
Fizz
4
Buzz
Fizz
7
8
Fizz
Buzz
11
Fizz
13
14
FizzBuzz
16
17
Fizz
19
Buzz
Fizz
22
...
```

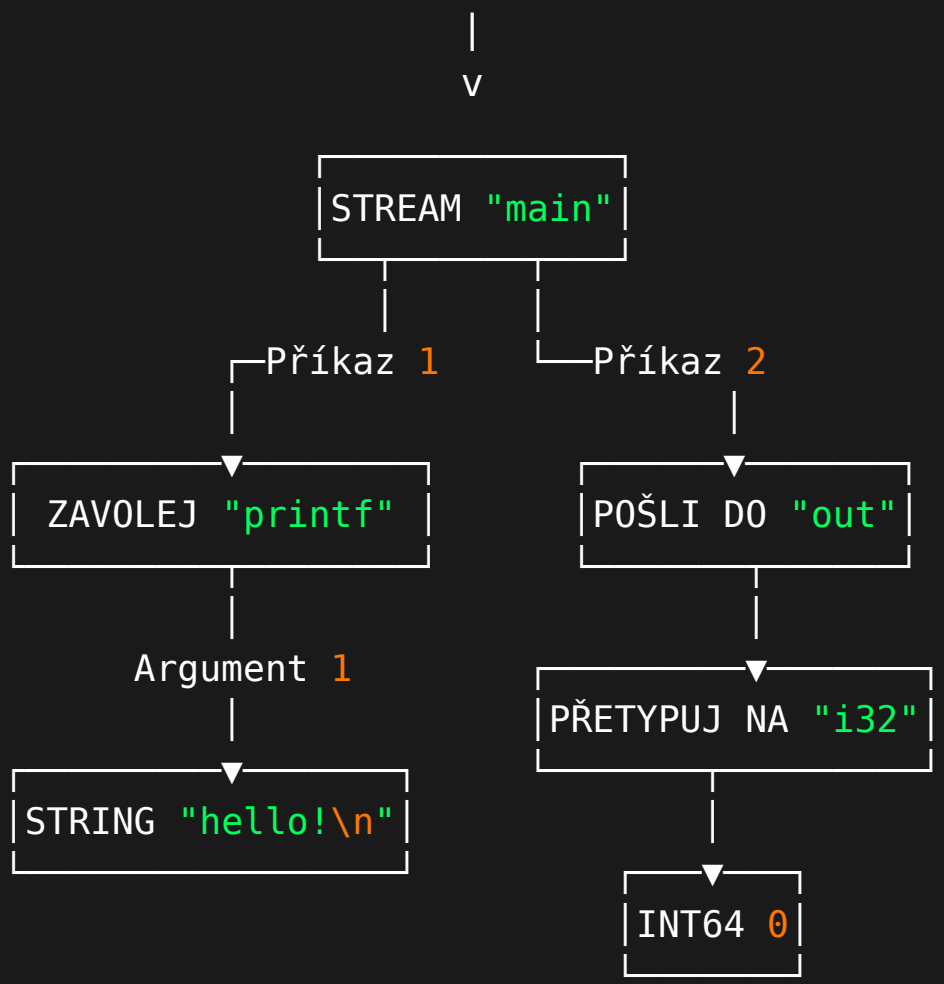
Překladač

LEXER -> PARSER -> ANALYZÉR -> CODEGEN

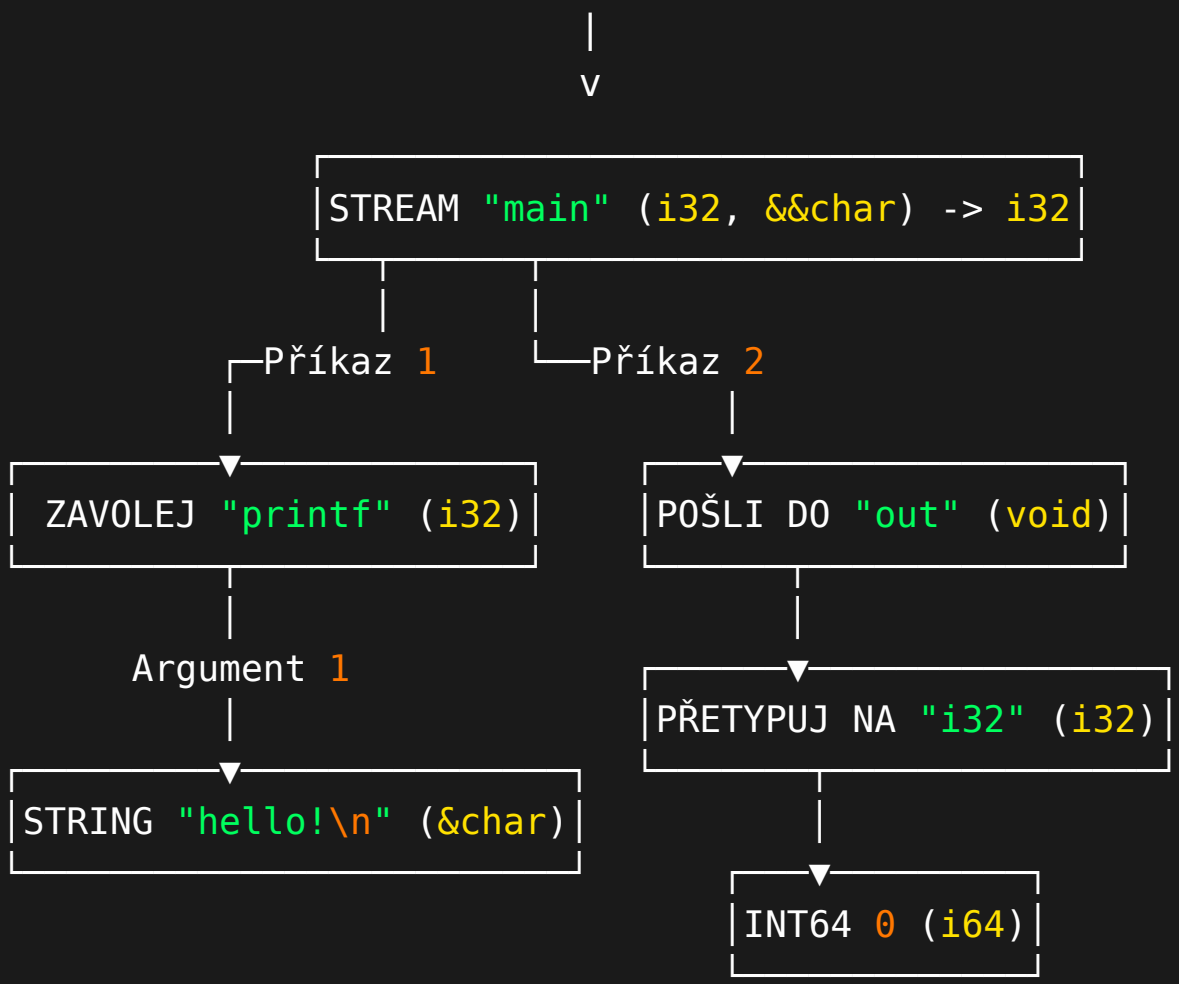
```
// vstupní bod
stream main (i32, &&char) -> i32 {
  printf("hello!\n");
  0 as i32 | out;
}
```

```
STREAM, (IDENTIFIER "main"), LPAREN,
(IDENTIFIER "i32"), COMMA, AMPERSAND,
AMPERSAND, (IDENTIFIER "char"), ARROW,
(IDENTIFIER "i32"), LBRACE, ..
```

LEXER -> **PARSER** -> ANALYZÉR -> CODEGEN



LEXER -> PARSER -> **ANALYZÉR** -> CODEGEN



LEXER -> PARSER -> ANALYZÉR -> **CODEGEN**

|
v

```
define i1 @stream_main(ptr %l, ptr %return_ptr, %clt* %c, i8** %block) noinline {
  %1 = load ptr, ptr %block
  indirectbr ptr %1, [ label %.block0, label %.block1, label %.blockblocked ]

.block0:
  %2 = call i32 (ptr, ...) @printf(ptr @str1)
  %3 = trunc i64 0 to i32
  store i32 %3, ptr %return_ptr
  store i8* blockaddress(@stream_main, %.block1), i8** %block
  ret i1 1

.block1:
  br label %.block0

.blockblocked:
  store i8* blockaddress(@stream_main, %.blockblocked), i8** %block
  ret i1 0
}
```

Ukázka