

Techniki Kompilacji – dokumentacja wstępna

Konrad Kaproń

Treść zadania:**Opis struktur binarnych**

Język deklaracyjny opisujący struktury binarne (z możliwą dokładnością do pojedynczych bitów). Projekt powinien umożliwiać zakodowanie i zdekodowanie dowolnej opisanej struktury i prezentację jej w wybranym formacie. Powinno być możliwe zdefiniowanie pól zależnych - np. pole Length i Contents.

Składnia:

letter	= 'a..z' 'A..Z'
digit	= '0..9'
seq	= { letter digit } { seq }
ident	= letter { seq }
intconst	= digit { intconst }
comment	= # ... \n
data	= (structid unionid declaration) { data }
intvalue	= intconst (ident { arrayid })
arrayid	= '[' intvalue ']'
rangeid	= '(' intvalue ')'
type	= 'int' { rangeid } ident
declaration	= type { arrayid } { ident } ';'
structid	= 'struct' ident '{' data '}' ';'
unionid	= 'union' rangeid ident '{' caseblock { 'default' '{' data '}' } '}' ';'
caseblock	= 'case' rangeid '{' data '}' { caseblock }

Sposób uruchomienia:

Possible options:

-h [--help]	Prints this message
-d [--datafile] arg	Path to file with data declaration
-b [--binfile] arg	Path to file with binary data
-t [--test] [=arg(=1)] (=0)	Produce output more suitable for tests

Zasadniczo, program dostaje z linii poleceń ścieżki do pliku z deklaracją danych, oraz do pliku binarnego, który stara się odczytać jako zawierający te dane. Jeśli w trakcie parsowania nie napotka błędu, wypisze odczytane dane wraz z wartościami z pliku binarnego. W przeciwnym przypadku, przerwie parsowanie i poda komunikat błędu.

Przykładowe poprawne wyjście programu:

Kappa:

int (1)[1] = {1}

Keepo:

bool:

int (1)[1] = {0}

int (16)[1] val = {3}

Przykładowy komunikat błędu:

Error at line 2: asdf was not declared, got: 'asdf'

Projekt realizacji:

- Klasa czytająca plik z danymi, rozpoznająca tokeny i zwracająca je jeden po drugim;
- Klasa czytająca plik binarny, dbająca o poprawne odczytanie danych o nieustandaryzowanych rozmiarach;
- Klasa sprawdzająca kolejność wczytywanych tokenów, budująca struktury danych i wypełniająca je wartościami;

Przykłady testowe:

1:

```
struct Kappa {  
    int(1);  
}; #This is a comment
```

```
struct Keepo {  
    Kappa bool;  
    int(16) val;  
};
```

2:

```
int; #propably the simplest one you can make
```

3:

```
struct Kappa {  
    int value;  
    union(value) Keepo {  
        case(1){  
            int(16);  
        }  
        case(2){  
            int(32);  
        }  
        default{  
            int(64);  
        }  
    };  
};
```

4:

```
struct Kappa {  
    int(4) value;  
    int(16)[value] array;  
};
```