New Language

Δευτέρα, 1 Δεκεμβρίου 2008
5:20 μμ

- Παραμετρικά πρότυπα
- Σχωρισμοί

synonyms

```
macro int sum(int x , int y)
{
	return x+y;
}
```

Αυτές οι συναρτήσεις θα παρεμβάλονται υποχρεοτικά και δεν θα δημιουργείτε αντίγραφό τους διαθέσιμο για κλήση. Δεν θα είναι δυνατή η λήψη της διεύθυνσής τους με το '&'. (κάτι πλησιέστερο στα macro της C)

```
inline int sum(int x , int y)
{
	return x+y;
}
```

Όπως το inline της C++. Δημιουργείτε ένα αντίγραφο για κλήση και επιτρέπεται η λήψη της διεύθυνσης με το '&'. Ο μεταγλωτιστής "παρακαλείτε" να παρεμβάλει την συνάρτηση στις θέσεις της κλήσης της αλλά δεν είναι υποχρεομένος να το κάνει.

```
int head(int x)
wraps body
{
	// head code here
	// body call here
}
```

Θα μπορούσε να είναι pointer σε συνάρτηση;

το σώμα της "head" ωφείλει να τελειώνει με μια κλήση στην "body" και οι επιστρεφόμενοι τύποι τους πρέπει να είναι οι ίδιοι. Στην ουσία δεν θα γίνεται κλήση της body αλλά άλμα χωρίς συνθήκη σε αυτή ή θα "πέφτουμε" στον κώδικά της μετά το τέλος του κώδικα της "head". (βλέπε κώδικα MIPS για sort/mergeSort και mege1/2 από εργασία)

Ως συνήθως ο μεταγλωτιστής ωφείλει κατα την βελτιστοποίηση να παρεμβάλει όσες συναρτήσεις έχουν (ωφέλιμο) μήκος (μετρούμενο σε bytes εντολών γλώσσας μηχανής) μικρότερο ή ίσο από το μήκος της κλήσης τους.
Στην πράξη, βέβαια δεν είμαι σίγουρος κατα πόσο γίνεται μια τέτοια σύγκριση, αφού η παρεμβολή συναρτήσεων γίνεται στα πρώτα στάδια της μεταγλώτισης και η παραγωγή γλώσσας μηχανής στα τελευταία. Θα χρειαζόταν να μεταγλωτίζωνται οι συναρτήσεις φύλλα πρίν από αυτές που τις καλούν και να καλείτε και κάποια συνάρτηση της επισθοφυλακής για να υπολογίσει το μήκος της κλήσης, αν υπάρχουν όλες οι πληροφορίες που χρειάζονται για να υπολογιστεί ακριβώς και δεν χρειάζεται να γίνουν πρώτα άλλες βελτιστοποιήσεις. χμ...

import global συναρτήσεις σαν μεθόδους
και export μεθόδους σαν global συναρτήσεις.

```
template<operator Op = {'+','-','*','/',','}>
double operator Op (double x , double y)
{
	//
}
```

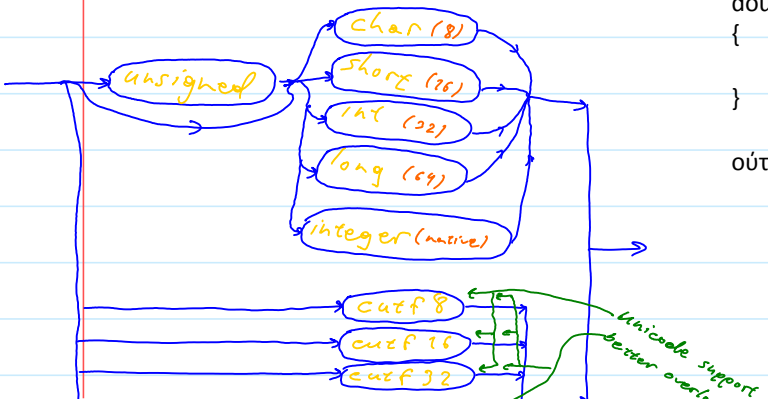
ή πιό απλά:

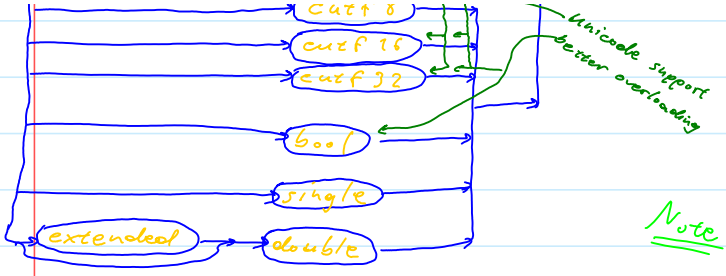
```
template<operator Op = {+,-,*,/,,}>
double operator Op (double x , double y)
{
	//
}
```

ούτος ή άλλως μετά από το διαχωριστικό περιμένει τελεστή.

Types

long long (128)?





Note:

Κάθε δεσμευμένη λέξη και κάθε κατηγορία όπως να αναγνωριστικά, οι τελεστές και τα διαχωριστικά, μπορούν να έχουν έναν διαφορετικό κωδικό, που θα οδηγεί σε μια ξεχωριστή συνάρτηση χειρισμού μέσω ενός πίνακα, ή μπορεί να αναπαρίσταται με ένα αντικείμενο διαφορετικής κλάσης και η σωστή συνάρτηση να επιλέγεται μέσω πολυμορφισμού.

extended long (64) ?

extended unsigned or unsigned extended ?

medium int ?

compile-time
run-time
} keywords

for example to choose if a template will be implemented with polymorfism or C++ templates

maybe a compile time function call operator as a dual to the runtime function call operator () ?

() {} [] < >

' "

Begin End

'( )' συνδυασμοί?

name ∈ set : specification

e.g. x ∈ integers :[32bit no overflow]

Arcadia Page 2