

Programming Language Concepts & Issues II: Homework questions

Design a persistent variable extension to C. Use this to rewrite your favorite C file manipulation program.

This isn't as easy as it seems. There are a number of issues

- How do you associate the internal variable with an external file name? You could do it as part of the declaration:

```
persistent "/u/usr/local/dict" char dictionary[] [] ;
```

The disadvantage of this is that you have to know the file name at compile-time. It would be better to associate file names with variables at runtime say for instance by a command

```
char dictionary[] [] ;  
char filename[] = "/u/usr/local/dict",  
persistent(filename,dictionary) ;
```

- How to handle random access files? Definitely use arrays.
- How to handle different file access capabilities? May need to use a const keyword to indicate that the file and data are read-only, or else do run-time checking.
- How to ensure that the type is correct? Probably keep type information associated with a file so that this is checked when the file is opened. This would also be useful for programs in other languages accessing the file.
- How to handle pointers in data? This is a real problem. Probably they should be disallowed which means that the standard approach to defining recursive data structures in C will not be persistent. However since you want to allow persistent arrays this would mean that arrays were no longer considered identical to pointers.
- When to update the external file if the internal data is changed? Perhaps only do it when the program terminates normally. Basically it behaves like pass-by-copy/value parameter passing. This could be inefficient. And standard output should behave specially.
- How to handle more than one program accessing the same file at the same time? This helped by the above semantics. You could also associate locks with files.

```
char[] in;
char[] out;

main()
{
    persistent("stdin",in);
    persistent("stdout",out);
    strcpy(in,out);
}
```

Consider an extension to C which would allow nesting of functions. Do you see any problems?

No this seems relatively straightforward. You would want to ensure that a function is only visible inside the block it is declared in.

What sort of block structure does C++ have?

The same as C: variable blocks can be nested but functions cannot be nested. However I think gcc allows you to nest functions

Does C++ use dynamic or static binding?

Static binding.

Programming Language Concepts & Issues III : Homework questions

C++ provides context-independent overloading. In the lecture we stated that context-independent overloading allows the function to be called to be uniquely identified by the types of the actual parameters. Then why does C++ sometimes complain of ambiguity when you use overloaded operators?

Because C++ also performs (automatic) type coercion and this can lead to ambiguity.

How would you implement type classes?

Method A:

One way would be to have different version of the function for each possible class. Thus `square` would give rise to

```
fun square_int x = x*x : int;  
fun square_real x = x*x : real;
```

And the compiler would call the appropriate version. This is how templates work in C++.

Method B:

Another way would be to pass in a dictionary of operations for with each variable. This is similar to how most object oriented languages support objects.

```
fun square NumDict x = NumDict.* (x,x);
```

Which method do you prefer and why? Method A has the advantage that the generated code is more efficient than Method B since there is no need to pass and look-up a dictionary of methods.

The main advantage of Method B over A is that Method B is more general and better supports compile time type checking since with Method A it is only when files are linked together that one can know which types the function must be generated for. This means that one can't do all type checking at compile-time. And with very polymorphic code it may be impossible to use Method A since we can't determine at link-time what will be the monotype of a variable. I.e we need dynamic method dispatch. Also Method A generates more code than Method B.

Programming Language Concepts & Issues IV : Homework questions

Give an example of a procedure which behaves differently if pass-by-reference and pass-by-value-result are used.

```
void dummy(int x, int y, int z) {
    x := 1;
    y := 2;
    z := x+y;}

main () {
    int i := 0;
    int j;
    dummy(i,i,j);
    writeln(j);
}
```

With pass-by-reference this writes 4 but with pass-by-value-result it writes 3.

Do you think verbatim replacement of the formal parameter by the actual parameter is the way to formalize call-by-name parameter passing? Consider what happens if a local variable has the same name as a variable in the actual parameter.

No you need to be careful to ensure that local variables are renamed away from the actual parameter names. For instance

```
void swap(int x, int y)
{
    int t;
    t := x; x := y; y := t;
}

main ()
{
    int s := 1;
    int t := 2;

    swap(t,s);
}
```

will not behave as expected if the local τ is not given a different name.

What are the key differences between object classes and abstract types?

- Methods in the abstract types require the abstract type arguments to be explicit while object classes have an implicit first argument.
- Object classes have notion of a hidden internal updatable state: thus they only work with imperative programming while abstract types also work with logic and functional programming.

What do you think are the differences between object-based and object-oriented programming languages?

See the next lecture.

Programming Language Concepts & Issues V: Homework questions

Write a Prolog query to find out which objects orbit mars.

```
| ?- orbits(S,mars).
```