
Subject: upp GTK compatibility for Ubuntu broken?
Posted by [kohait00](#) on Fri, 21 Oct 2011 06:30:45 GMT
[View Forum Message](#) <> [Reply to Message](#)

I congratulate you for this initiative.

I vote for:

- 1.- theide: Windows 64-bit debugging support. Remark: I think the debugger still contains some minor bugs;
- 2.- theide look&feel improvements (docking, new icons...);
- 3.- configurable menu bar: a gre

Subject: Re: upp GTK compatibility for Ubuntu broken?
Posted by [dolik.rce](#) on Fri, 21 Oct 2011 07:36:19 GMT
[View Forum Message](#) <> [Reply to Message](#)

at improvement, certainly;

- 4.- Multi threading for Painter: would it be possible to implement something similar fot GLCitr?.

Best wishes.

Thanks,

remark: somehow, my last post didn't work, that's why is empty.

Javier

Thinking about TCC and runtime compilation frameworks (also web templates) I have got an idea how represent the code structure in the tree of virtual objects (basically, via function pointers).

I have put together a litte experimental snippet:

```
#include <Core/Core.h>
```

```
using namespace Upp;
```

```
struct Oper {  
    virtual double Execute() = 0;  
    virtual ~Oper() {}  
};
```

```
struct BinOper : Oper {  
    One<Oper> a;
```

```

One<Oper> b;
};

struct Add : BinOper {
    virtual double Execute() { return a->Execute() + b->Execute(); }
};

struct Sub : BinOper {
    virtual double Execute() { return a->Execute() - b->Execute(); }
};

struct Mul : BinOper {
    virtual double Execute() { return a->Execute() * b->Execute(); }
};

struct Div : BinOper {

```

Subject: Re: upp GTK compatibility for Ubuntu broken?
 Posted by [kohait00](#) on Fri, 21 Oct 2011 08:01:36 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

}
};

struct Var : Oper {
    double *var;
    virtual double Execute() { return *var; }
};

struct Compiler {
    VectorMap<String, double *> var;
    One<Oper>

```

Subject: Re: upp GTK compatibility for Ubuntu broken?
 Posted by [mdelfede](#) on Fri, 21 Oct 2011 08:39:34 GMT
[View Forum Message](#) <> [Reply to Message](#)

```

> Term(CParser& p);
One<Oper> Exp(CParser& p);
One<Oper> Factor(CParser& p);
};

One<Oper> Compiler::Term(CParser& p)
{

```

```
One<Oper> result;  
if(p.IsId()) {  
    double *v = var.Get(p.ReadId(), NULL);  
    if(!v)  
        p.ThrowError("unknown variable");  
    result.Create<Var>().var = v  
}
```
