



SystemC Notion & QT

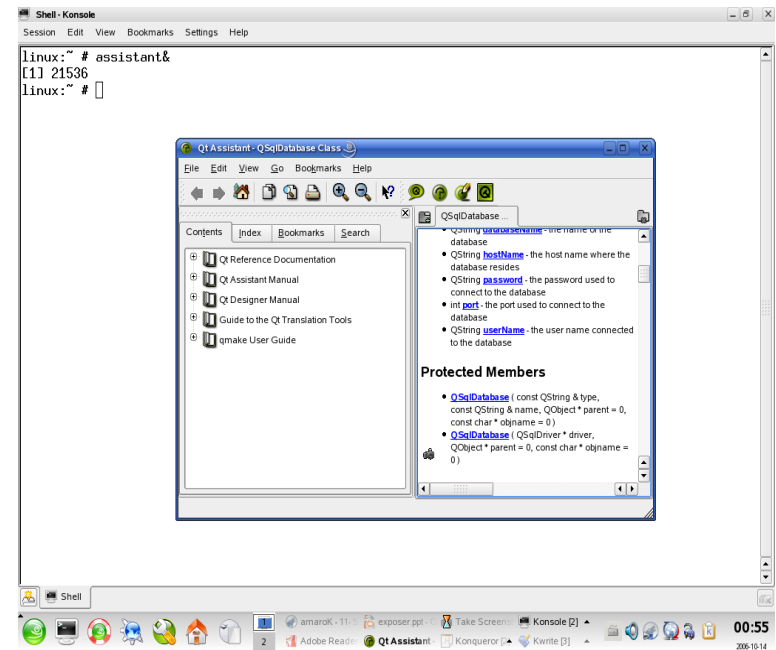
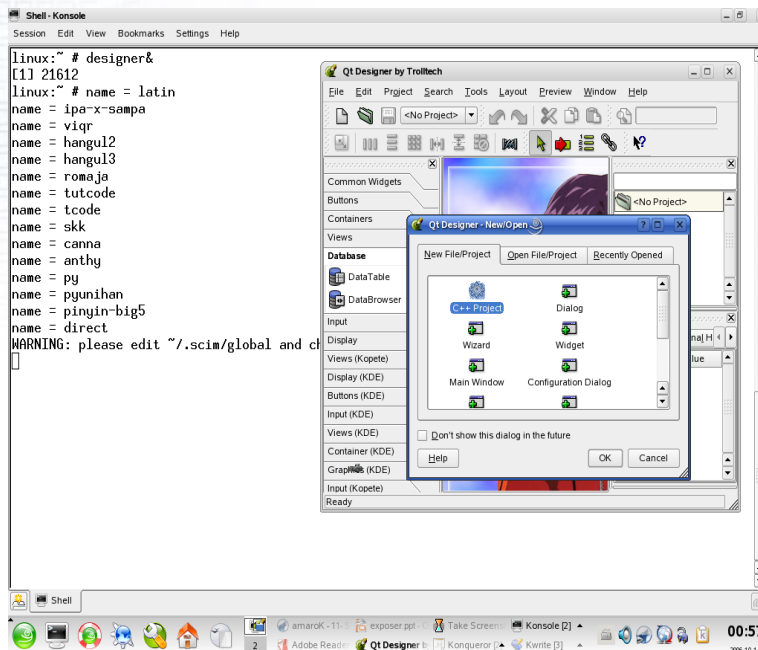


Exemples en pratiques

- Exemple -1- execution d'une commande shell avec QT
- Exemple -2- création d'une base de données MYSQL
- Exemple -3- interfaçage de la base de données avec QT
- Exemple -4- SystemC
- Exemple -5- interfacier le programme de SystemC avec QT

Exemple -1- execution d'une commande shell avec QT(1)

- lancez QTDESIGNER :
 - > designer (n'oubliez pas le caractere & pour qu'il soit en arriere plant)
- lancez l'ASSITANT de QT :
 - > assistant&





Exemple -1- execution d'une commande shell avec QT(2)

- créez un nouveau projet
- dans ce projet créer une nouvelle fenêtre de type dialogue
- insérez un bouton, un LineEdit et un textEdit
 - le bouton va vous permettre d'exécuter la commande
 - le LineEdit pour le saisie de la commande
 - le textEdit pour afficher le résultat de l'exécution
- choisissez le menu : Window->views->Object Explorer
 - accédez a l'onglet **Members** et ajoutez
 - **QProcess *p;**(dans les **Variables : public**)
 - **qprocess,h** (dans les **includes in declaration**)



Exemple -1- execution d'une commande shell avec QT(3)

- Cliquez avec le bouton droite de la souris
- Sur la boite de dialogue ajoutez un slot (gestionnaire d'événements)
 - connectez le slot créé avec le signal onclick() de la souris à l'aide du menu Tool->connect signal/slot;
 - implémentez le gestionnaire comme suit:

```
proc = new QProcess( this );  
proc->addArgument( "ls" );  
proc->addArgument( "-l" );  
proc->addArgument( "/root/exec/" );  
proc->addArgument( "/root/tmp2" );
```

```
connect( proc, SIGNAL(readyReadStdout()), this, SLOT(stdoutoutputread()) );  
if ( !proc->start() ) {  
// error handling }
```



Exemple -1- execution d'une commande shell avec QT(4)

- Ajoutez un slot `stdoutputread()`;
- Implémentez le slot `stdoutputread()`

```
void Form1::stdoutputread()  
{  
    textEdit2->setText(QString(proc->readStdout()));  
}
```
- Lancez la commande `qmake` pour créez le Makefile
- Lancez la commande `make` pour compiler et générer l'exécutable,



Exemple -2- création d'une base de données MYSQL

- activez le service de mysql
 - pour suse il suffit de lancer le YaSt
 - système--> éditeur du niveau d'exécution
 - lancez mysql
- lancez la console et tapez : `mysql -u root;`
 - `mysql> create database « test1 »;`
 - `mysql> connect test1;`
 - `mysql> create table tab1(code numeric(8,0), nom varchar(20), valeur varchar(20), primary key(code));`
 - `mysql> commit;`



Exemple -3- interfaçage de la base de données avec QT

- créez un nouveau projet
- créez une nouvelle boite de dialogue
- insérez la composante data table;
 - connectez la base déjà créée;
 - choisissez le tableau tab1;
 - choisissez les données à afficher
 - ,,
 - confirmer
 - testez la connections à l'aide du menu **Preview**
- Lancez la commande qmake puis make
- Modifiez la connection dans le fichier du répertoire ,ui/form1 ,cpp et relancez make



Exemple -4- SystemC

- préparation de la plate forme
 - installation de **green socs (compilateur version 3,4,2)**
 - installation de SystemC a l'aide de green socs
 - installation de gtkwave (si c'est possible)
- teste de la plate forme :
 - lancement des exemples
- édition d'un exemple
 - compteur
 - porte logique AND




Exemple -4- SystemC (1)

- préparation de la plate forme
 - installation de **green socs**:
 - décompresser **gs-systemc-1.0.0**
 - lancer **gs-setup**
 - installation de SystemC
 - décompresser systemc-2.1.v1.tar
 - >cd systemc-2.1.v1
 - >mkdir objdir
 - >cd objdir
 - >csh
 - >setenv CXX /[Myrep]/**gs-systemc-1.0.0**/bin/g++
 - >.,/configure
 - >make install



Exemple -4- SystemC (2)

- teste de la plate forme :
 -  lancement des exemples :
 - >cd /[Myrep]/systemc-2.1.v1/objdir/example/sysc/pipe
 - pipe>make check
 - >,/pipe



Exemple -4- SystemC (3)

- Édition d'un exemple (compteur)

- créer un répertoire dans `/exemple/sysc/Myproject` qui contient les `.CPP` et les `.H` Makefile,in Makefile,am(copier les à partir d'un exemple)
- un autre dans `/objdir/exemple/sysc/Myproject` qui contient Makefile copier le à partir d'un exemple)

-  `<=>`VHDL

-  `--interface`

-  `library ieee;`

- `use ieee.std_logic_1164.all;`

- `use ieee.std_logic_unsigned.all;`

- `entity counter is`

- `port(clock : in std_logic;`

- `load : in std_logic;`

- `clear : in std_logic;`

- `din : in std_logic_vector(7 downto 0);`

- `dout : inout std_logic_vector(7 downto 0));`

- `end counter;`



Exemple -4- SystemC (4)

- `<==>` SystemC

```
#include "systemc.h"
SC_MODULE(counter)
{
    sc_in<bool> clock;
    sc_in<bool> load;
    sc_in<bool> clear;
    sc_in<sc_int<8> > din;
    sc_out<sc_int<8> > dout;
    int countval;
    void onetwothree();
    SC_CTOR(counter)
    {
        SC_METHOD(onetwothree);
        sensitive_pos (clock);
    }
};
```



Exemple -4- SystemC (5)

- Architecture VHDL:

architecture rtl of counter is

```
signal countval : std_logic_vector(7 downto 0);
```

```
begin
```

```
    process(load, clear, din, dout)
```

```
    begin
```

```
        if clear = '1' then
```

```
            countval <= "00000000";
```

```
        elsif load = '1' then
```

```
            countval <= din;
```

```
        else
```

```
            countval <= dout + "00000001";
```

```
        end if;
```

```
    end process;
```

```
    process
```

```
    begin
```

```
        wait until
```

```
            clock'event
```

```
            and clock =
```

```
            '1';
```

```
        dout <=
```

```
            countval;
```

```
    end process;
```

```
end rtl;
```



Exemple -4- SystemC (6)

- SystemC:

```
#include "counter.h"
```

```
void counter::onetwothree()
```

```
{
```

```
    if (clear) {
```

```
        countval = 0;
```

```
    } else if (load) {
```

```
        countval = din.read();
```

```
    } else {
```

```
        countval++;
```

```
    }
```

```
        dout = countval;
```

```
}
```



Exemple -4- SystemC (7)

```
#include "systemc.h"
#include "counter.h"
int sc_main(int ac, char* av[]){
    //sc_signal<bool> clock;
    sc_signal<bool> load;
    sc_signal<bool> clear;
    sc_signal<sc_int<8> > din;
    sc_signal<sc_int<8> > dout;
    sc_clock clock1("CLOCK1", 75, 0.5, 0.0);
    counter cter;
    cter.clock(clock1);
    cter.clear(clear);
    cter.din(din);
    cter.dout(dout);
    sc_start(clock,-1);
    return 0;
}
```




Exemple -4- SystemC (8)



```
FILE *f_res,  
f_res = fopen("out_val","w");  
''  
fprintf(f_res,"a=0 et b=X ==> s= %e \n",s.read());  
''  
fclose(f_res);
```



Exemple -4- SystemC (9)

- Makefile

-  recommandations

-  copiez un Makefile existant dans votre répertoire (celui de l'exemple de **systemC** que vous avez créé et celui de **objdir/,,/**) et effectuez les modifications suivantes :
 - changez les noms des fichiers avec celui de votre projet
 - changez automake1,6 avec celui de votre version (>sh; man automake)
-  dans le répertoire **objdir/example/sysc/myproject/** lancez la commande **make check**