# VNC



# 库修改的地方移植时需注意

Pdir 改了占用率获取,unix df -h和linux不一样,同时好像中英文也不一样注意

common\_getAllFilesAndPathsRecursion改的内部判断是否文件的类型

common\_getAllFilesRecursion改的内部判断是否文件的类型

pmap错误: 将map改为std::map

找不到g\_all等,将interface.cpp加入make文件

获取线程id:用pthread\_self()

# C语言开发

Fopen改为fopen64

## unix struct dirent' has no member named 'd\_type'

unix下没有,采用其他方式解决

## Bzero找不到错误

#define bzero(a, b) memset(a, 0, b)

## expected template-name before '<' token

一般用的时候加上std::

## Ld output错误

LIBS+=-lpthread

LIBS+=-lxml2

LIBS+= -lnsl

LIBS+= -lrt

LIBS+= -lsocket

## 如果直接打开linux工程错误,则直接新建工程添加.h.c等再复制代码

## 常见错误

### 定义变量超限(一般超过几千字节)

使用动态分配.

### Unsigned Long类似的类型在linux 和unix字节长度不一样

改成usigned long long

## 获取线程id

pthread\_self()直接用这个

Linux 3.2.0使用无符号长整型表示pthread\_t数据类型

Solaris 10把pthread\_t数据类型表示为无符号整型（不是长整型）

FreeBSD 8.0和Mac OS X 10.6.8用一个指向pthread结构的指针来表示pthread\_t数据类型（注意：事实上，Mac OS X与FreeBSD不用，Mac OS X主线程ID和使用pthread\_create创建的新线程的ID不在相同的地址范围内）

# Solaris11

# Solaris10

## 安装系统不要太小,100G起,不然扩容特别难

要修改分区,/目录弄90G,默认是/export/home不符合要求

要选network,且选dhcp

选utf8语言,有中文输入法

选择liberat mono字体,好看

## 关于写代码,不要与windows共享,影响速率,会卡

## 网络

虚拟机要设置成nat模式,然后重启网络服务网络就可以了

## 登陆ssh慢

1.2 解决办法  
出现这种情况通常有两种解决办法。  
1. 将客户端所在网段加入到DNS的反向解析中  
只需加网段即可，不需要列出具体机器IP或机器名。当主机需要解析客户端时，主机会去DNS查询反向解析域。如果客户端所在网段已经加入到DNS中，虽然主机仍然无法解析，但会立即返回，不需要等待超时。  
2. 修改Solaris主机sshd\_config配置，使用ssh不执行客户端IP反向解析  
在/etc/ssh/sshd\_config文件中加入:  
LookupClientHostnames no  
然后重启ssh：  
svcadm restart ssh  
  
2. 与Kerberos和LDAP相关  
另一种情况是与Kerberos和Active Directory有关。如果Solaris配置为通过Active Directory进行用户登录验证，当Active Directory不可用时，会出现SSH登录慢的问题。  
2.1 现象  
SSH登录时提示输入密码较快，输入密码后很长时间才能登录成功。  
Console登录速度也很慢。  
2.2 解决办法  
修改/etc/pam.conf文件，注释或者删除以下条目：  
other auth sufficient pam\_krb5.so.1

## 更改默认sh为bash

/etc/passwd

将root那一行后边的:/sbin/sh

改为/bin/bash

## 增加库目录,库文件目录

crle -u -v -l /opt/SUNWspro/lib

## 备忘

./configure -prefix $MY\_PREFIX -opensource -confirm-license -nomake tests -R /opt/csw/lib/64 -R /opt/csw/X11/lib/64 -qt-xcb -platform solaris-g++-64 -verbose

## [gmake与make的区别](http://blog.chinaunix.net/uid-7739333-id-2044978.html)

分类： LINUX

这几天在做qtopia的移植发现还有个gmake，不清楚和make的区别，所以查来看看，并记下来，方便查找。  
～～～～～～～～～～～～～～～～  
gmake是GNU Make的缩写。  
Linux系统环境下的make就是GNU Make，之所以有gmake，是因为在别的平台上，make一般被占用，GNU make只好叫gmake了。  
比如在安装二进制文件进行编译时要使用make命令,但如果在Solaris或其他非GNU系统中运行，必须使用GNU make，而不是使用系统自带的make版本，这时要用gmake代替make进行编译。  
～～～～～～～～～～～～～～～～  
Solaris包含两种make工具：  
GCC (GNU Compiler Collection)工具套装包括一个 C 编译器和一个 C++ 编译器  
Solaris自身的 make 实用程序  
  
  
GNU gmake在 make 文件中将调用的C编译器名称为 **gcc**，C++编译器的名称为 **g++**   
make实用程序在make文件中将调用的C编译器名称为**cc**，C++编译器名称为**CC**  
  
gmake和make是用的编译器的命令行选项中有所区别  
除了少数几个基本选项(如“-c”和“-g”)之外，编译器接受的多数选项都不同  
  
gmake与make主要用在Linux与Solaris之间的应用程序移植中  
  
～～～  
注：

Solaris是Sun公司开发和发布的操作系统，是UNIX系统的一个重要分支。

起源：首先从BSD Unix移植而来，随后逐渐转移到了System V标准。

支持平台：目前，Solaris除可以运行在SPARC CPU上外，还可以运行

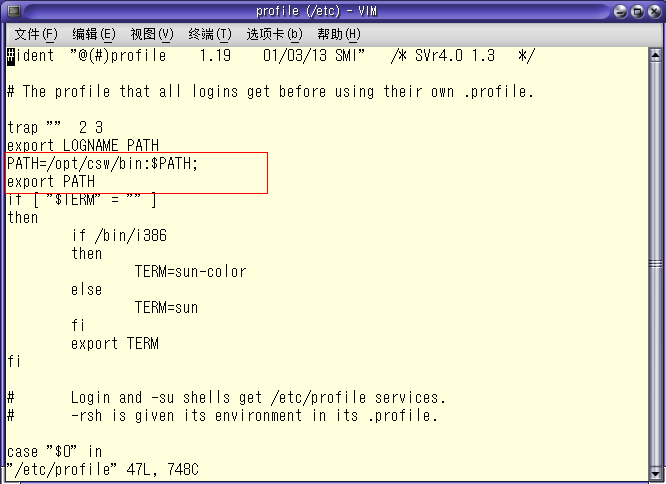
在X86 (IA32)的CPU上。最近又被移植到了最新的AMD64位CPU Opteron上.

市场地位：在服务器市场上，Sun的硬件平台具有高可用性和高可靠性，Solaris是

当今市场上处于支配地位的Unix类操作系统。

## Solaris 10 环境变量（PATH）的查看与设置

### 永久



### 临时

1.PATH的设置  
#PATH=$PATH:新的变量路径;export PATH  
例如：PATH=$PATH:/usr/ccs/bin;export PATH  
2.PATH的查看  
＃echo $PATH

## ftp服务,ssh服务,网络服务

1. FTP   
默认FTP是关闭的，启动命令：

**Shell代码  [收藏代码](javascript:void())**

1. # svcadm enable /network/ftp

此时查看FTP服务状态：

**Shell代码  [收藏代码](javascript:void())**

1. # svcs -l network/ftp

默认情况下，root用户无法登录，需要修改/etc/ftpd/ftpusers文件，把root那行前面加个#注释掉就可以了。   
2. SSH   
默认SSH是开启的。但是root用户无法登录，需要修改/etc/ssh/sshd\_config，把里面的 PermitRootLogin改为 yes，再重启ssh服务，   
重启命令：

**Shell代码  [收藏代码](javascript:void())**

1. # svcadm restart network/ssh

3. 重启网卡   
以前修改/etc/hosts文件的IP后都是通过重启机器来使设置生效的，现在也可以通过下面的命令让它生效，无需重启机器。

**Shell代码  [收藏代码](javascript:void())**

1. # svcadm restart physical

## 防火墙

**# svcs -a |grep network |egrep "pfil|ipf"**

enable         Oct\_22   svc:/network/ipfilter:default  
online          Oct\_22    svc:/network/pfil:default

**#svcadm disable  svc:/network/ipfilter:default**

## solaris10 sshd服务

bash-3.00# svcs -a | grep ssh  
disabled       20:51:27 svc:/network/ssh:default  
可以看到ssh 服务没有启动。  
  
可以使用如下命令，启动和停止sshd。  
  
启动 sshd  
  
svcadm enable svc:/network/ssh:default  
  
停止 sshd  
  
svcadm disable svc:/network/ssh:default  
  
sshd的配置文件在/etc/ssh下。  
  
默认的配置文件是不允许root账号登录的，可以编辑sshd\_config文件。把 PermitRootLogin no 改成 PermitRootLogin yes即可。  
  
windows下可以使用putty登录到**Solaris**上。

## 各种错误

solaris 下读取或者写入有上限2G，必须用fopen64

无缘无故挂了，肯定是数组超限，用动态分配

## solaris qt安装

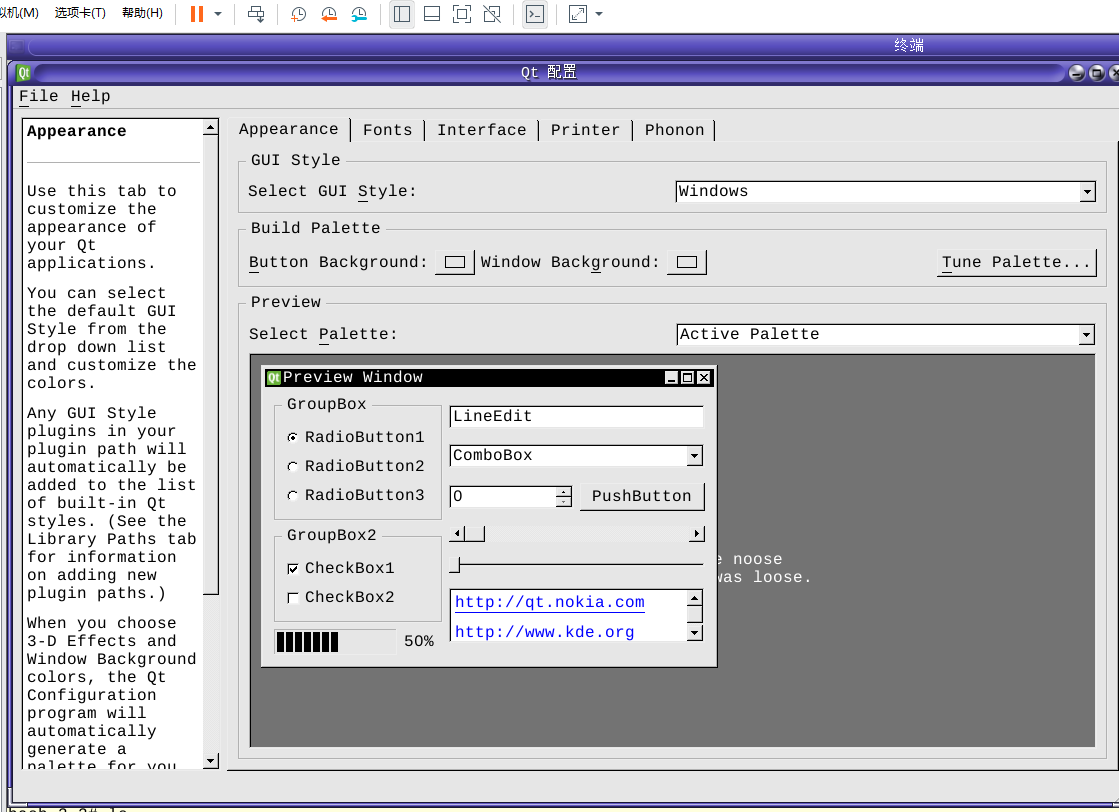
### Qt,qtcreator直接移植到其他solaris(新)

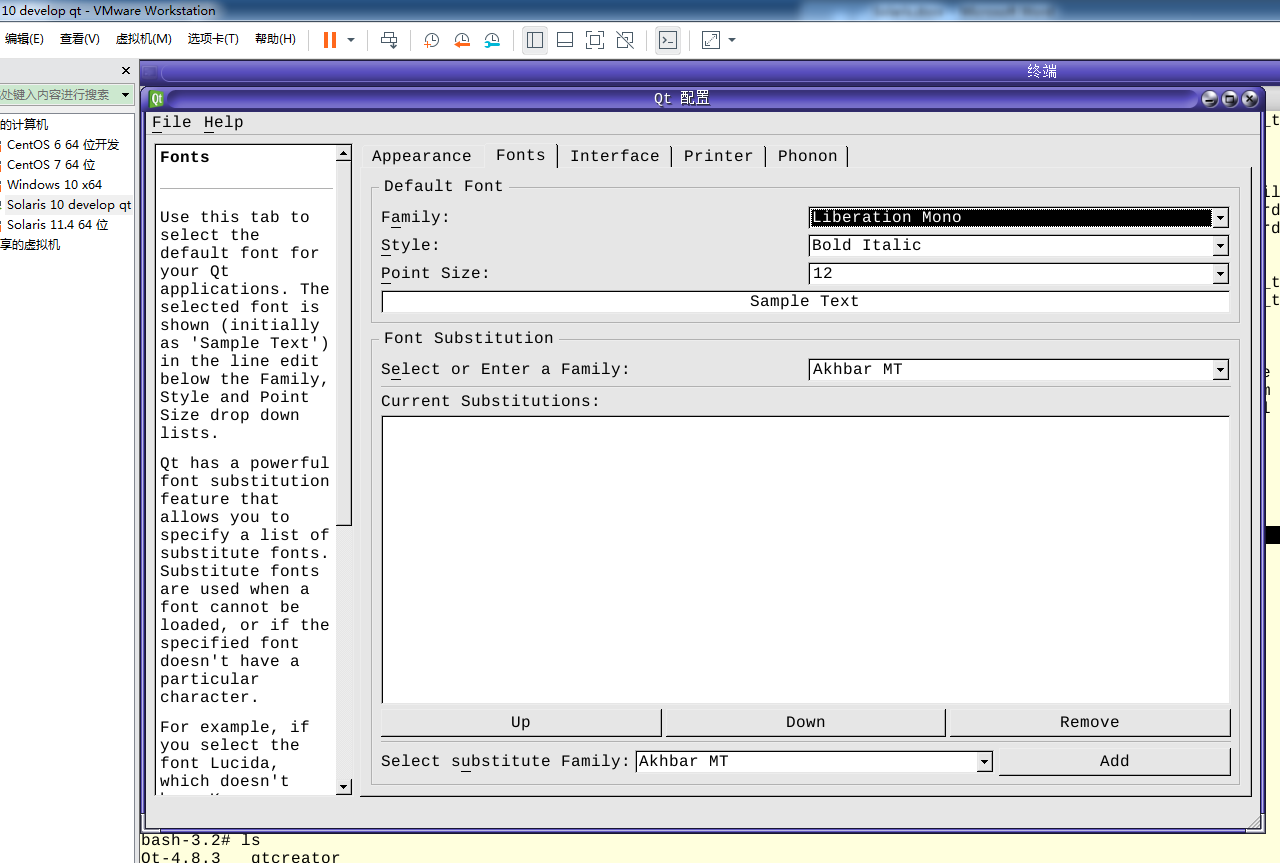
Qt直接把/usr/local/trolletch那个文件夹靠过来配置好就行了

Qtcreator直接用,不需要任何操作

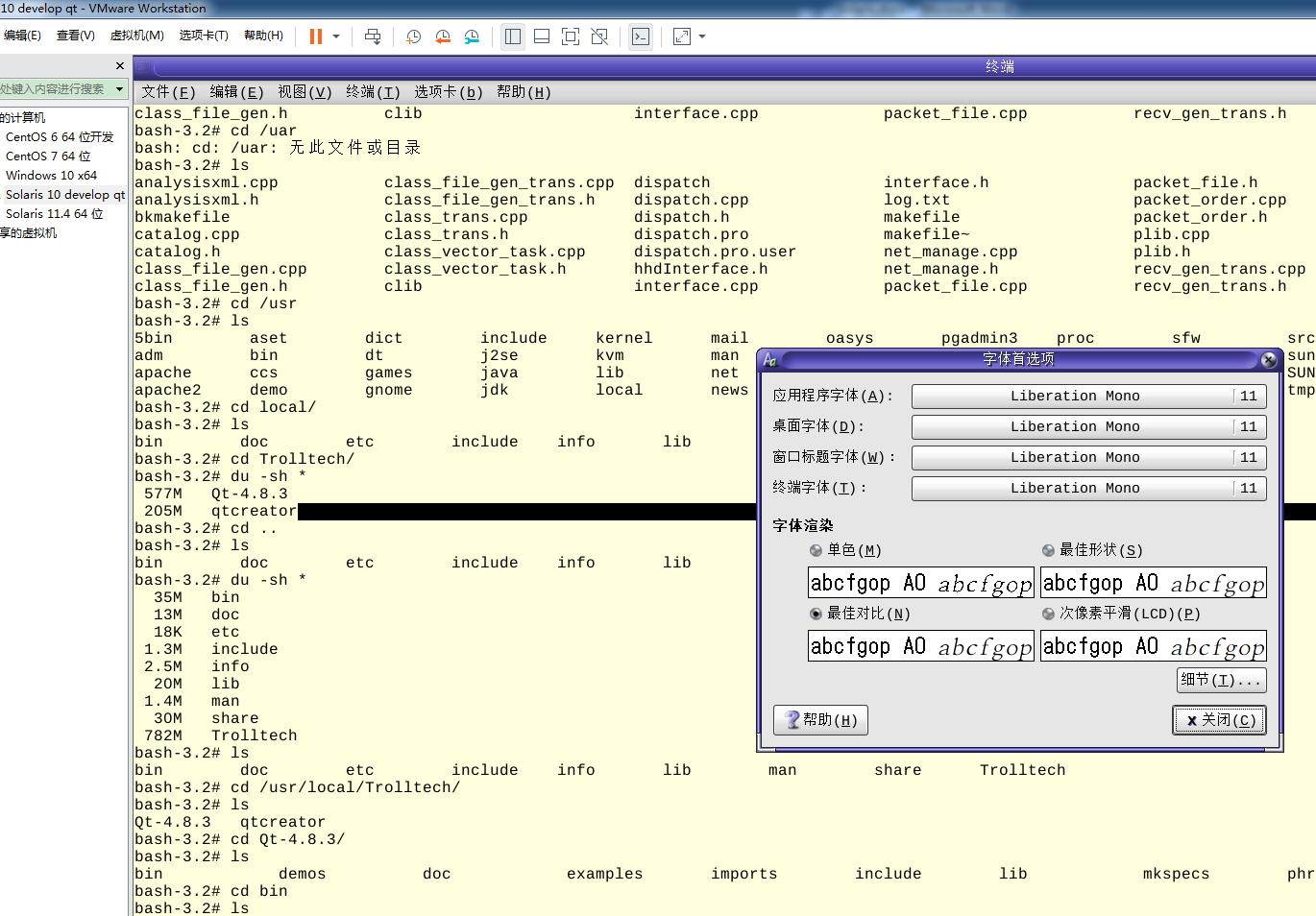
Qtcreator打不开缺库的话加入lib库目录即可(一般是加/usr/local/trolletch那个)

qtconfig

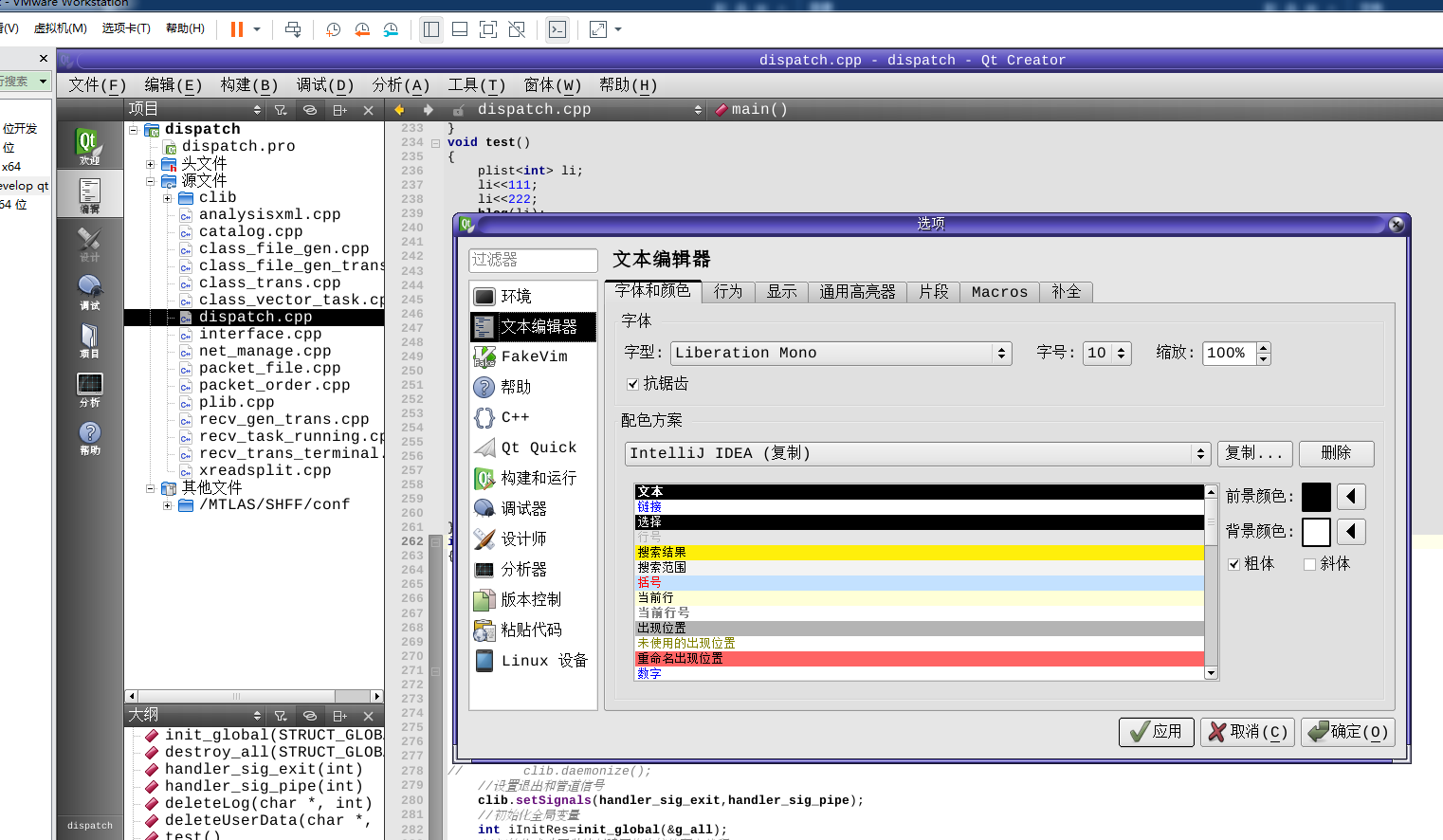




系统字体不好看,系统首选项中也改成这个字体



Qt里面文本编辑器也要改



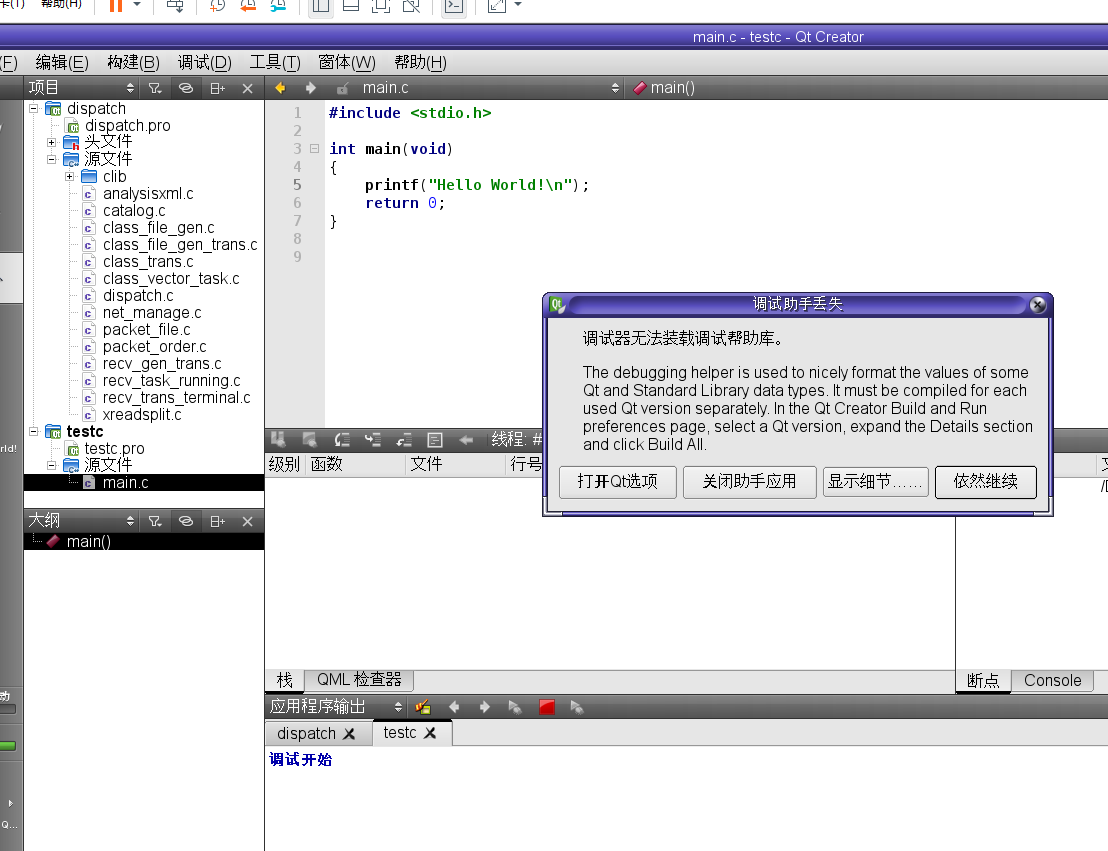
中文乱码:

系统要选中文,不然肯定乱码

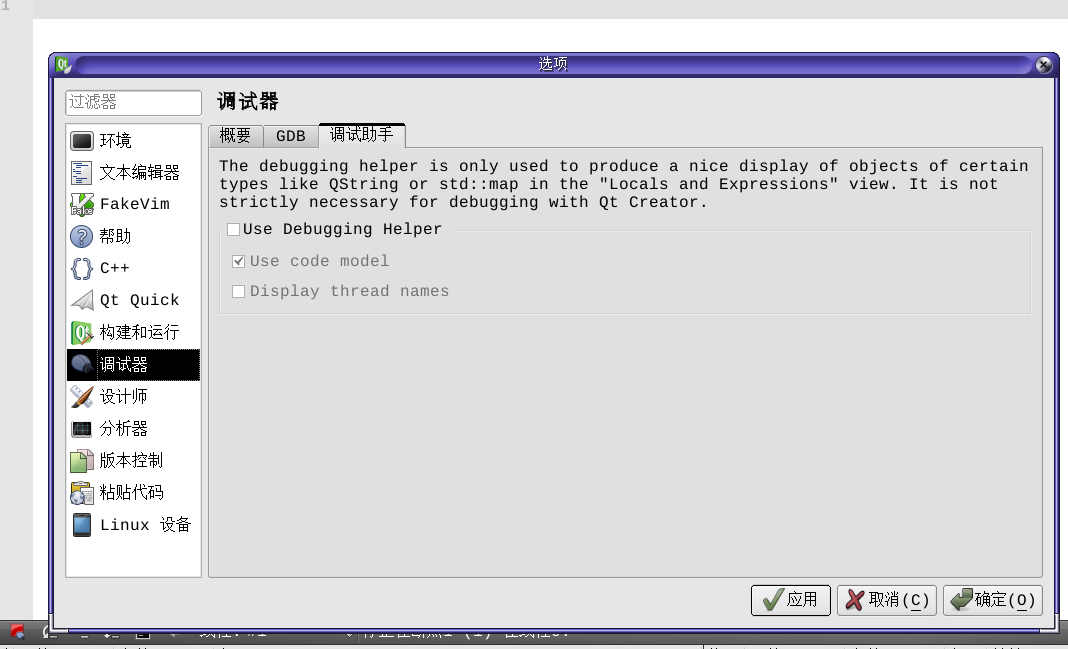
提示工具链找不到按下图配置

移植之前先把gdb装好





出现这个的话



把use debugging这个去掉,上边说了不影响

### 两个源码直接放桌面,都是编译好的

### /etc/profile

# The profile that all logins get before using their own .profile.

trap "" 2 3

export LOGNAME PATH

**PATH=/usr/ccs/bin:/usr/sfw/bin:$PATH;**

**export PATH**

if [ "$TERM" = "" ]

then

if /bin/i386

then

TERM=sun-color

else

TERM=sun

fi

export TERM

fi

### qt的源码直接make install

### qtcreator 再qmake

### make(不要make install)

### 启动缺少库时:库目录

crle -u -v -l solaris-xlfd-qt-creator-2.5.2-src/lib/qtcreator/

solaris qt

编译时不要去掉库，直接-platform solaris-g++

不能直接把库考到/usr/lib下，不然会打不开，用crle -u -v -l 增加库

qtcreator要用gmake,不然出错

qtcreator找不到插件不要乱考，加入源码下的lib/qtcreator库目录即可

同样的系统,只要把编译好的qt直接用就行,但是要make install

### QT4.8.3

#### 源码编译

./configure -platform solaris-g++

make

make install

make install是安装到/usr/trolltech目录

然后

crle -u -v -l /usr/local/Trolltech/Qt-4.8.3/lib

#### 错误

##### 找不到ar

加入PATH

PATH=/usr/ccs/bin/:$PATH;

export PATH

### qtcreator2.5.2

qmake 生成makefile，然后make

然后crle –u-v-l 源码lib目录即可

#### 问题

##### Building Qt Creator [Qt Creator](http://qt-project.org/wiki/Category:Tools::QtCreator) is the preferred IDE for all C++ development at SPT. Compiling Qt Creator 2.5.2 requires a few changes to the source files to account for the differences between system libraries available on Linux and Solaris. The qmake file that generates the qtcreator\_process\_stub application has a configuration for solaris, however the pattern specified is not matched on OpenIndiana. In particular the pattern should be solaris\* and not solaris-.\* as specified in the project file. Change solaris-.\* to solaris\* in process\_stubs.pro (line 17)

vi src/libs/utils/process\_stub.pro  
  17        solaris\*: LIBS += -lsocket

The virtualserialdevice\_posix class does not compile due to two missing items. The first of them is a missing constant, which for Solaris is in the sys/filio.h include file which needs to be included. The second is a missing cfmakeraw function which needs to be created. Since this function is used only in the virtualserialdevice\_posix class, we decided to implement the missing function as a private method of the class.  
  
Edit src/shared/symbianutils/virtualserialdevice.h and add a private cfmakeraw function.

vim src/shared/symbianutils/virtualserialdevice.h

  39  
  40  #include <termios.h> *// Moved from virtualserialdevice\_posix.cpp line 38*  
  77        void **cfmakeraw**( **struct** termios\* );

Edit src/shared/symbianutils/virtualserialdevice\_posix.cpp. Add the missing cfmakeraw method implementation and an include for sys/filio.h to bring in missing FIONREAD constant value.

   #include <sys/filio.h> *// Adds missing FIONREAD constant*

 void VirtualSerialDevice::**cfmakeraw**( **struct** termios\* termios\_p )

 {

     termios\_p->c\_iflag &= ~(IGNBRK|BRKINT|PARMRK|ISTRIP

                            |INLCR|IGNCR|ICRNL|IXON);

     termios\_p->c\_oflag &= ~OPOST;

     termios\_p->c\_lflag &= ~(ECHO|ECHONL|ICANON|ISIG|IEXTEN);

    termios\_p->c\_cflag &= ~(CSIZE|PARENB);  
    termios\_p->c\_cflag |= CS8;

 }

The next step is to enable the qmlprojectmanager plugin. The qmake project file configuration seems to lead to this plugin project being excluded with the Solaris Qt configuration. We just add the subdirs project to the plugins.pro file.

vi src/plugins/plugins.pro  
  47            plugin\_qmlprojectmanager \

Note that this line originally occurs at line 63 in the project file within the contains(QT\_CONFIG, declarative) block.  
  
We also need to enable the posix timer header include while building botan in the src.pro file. Add a “|solaris\*” to lines 287, and 571.

vi src/libs/3rdparty/botan/src/src.pro  
  287 linux\*.g++\*|freebsd\*.g++\*|solaris\* {

##### error: ISO C++ forbids casting between pointer-to-function and pointer-to-object

m\_refSockFD = reinterpret\_cast<RefSockFDPtr>(dnsSdLib.resolve("DNSServiceRefSockFD"));

m\_refSockFD = RefSockFDPtr(dnsSdLib.resolve("DNSServiceRefSockFD"));

##### QList is not a class, struct, or union type

template <class WizardClass>

QList<WizardClass\*> createMultipleBaseFileWizardInstances(const QList<BaseFileWizardParameters\*> parametersList, ExtensionSystem::IPlugin \*plugin)

{

QList<WizardClass\*> list;

foreach (BaseFileWizardParameters \*parameters, parametersList) {

WizardClass \*wc = new WizardClass(parameters, 0);

plugin->addAutoReleasedObject(wc);

list << wc;

}

return list;

}

或

推荐:

template <class WizardClass>

QList<WizardClass\*> createMultipleBaseFileWizardInstances(const QList<BaseFileWizardParameters> parametersList, ExtensionSystem::IPlugin \*plugin)

{

QList<WizardClass\*> list;

foreach (BaseFileWizardParameters parameters, parametersList) {

WizardClass \*wc = new WizardClass(parameters, 0);

plugin->addAutoReleasedObject(wc);

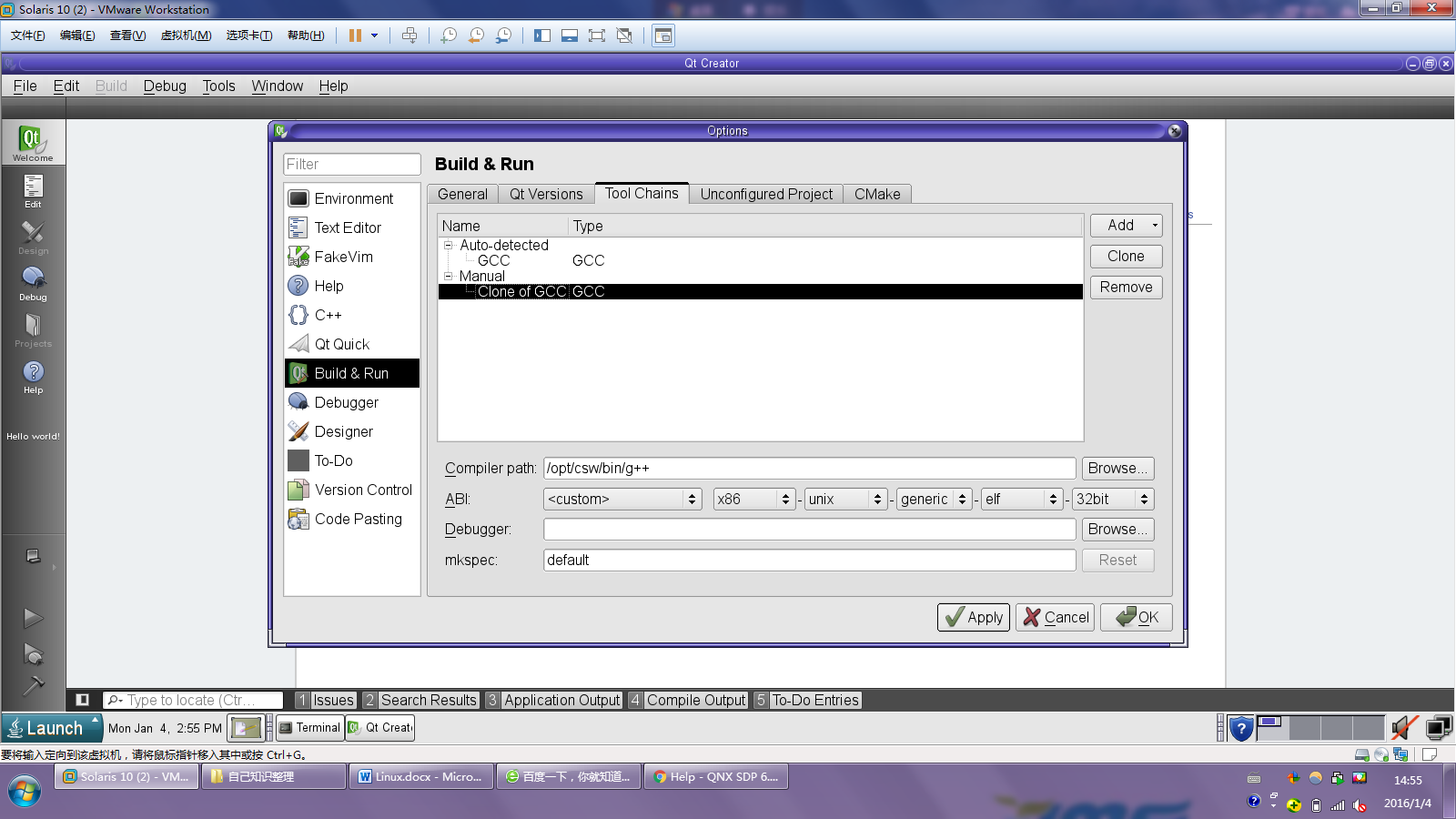
list << wc;

}

return list;

}

##### no toolchain



##### 字体模糊

qtconfig-windows

##### 后续

qtcreator make完成之后后安装(不要make install)完成后,在当前目录加库路径/lib/qtcreator

crle -u -v -l lib/qtcreator/

### 2 查看qt官方文档步骤：

* ./configure
* gmake
* gmake install

    如果编译器安装好，而且 ./configure成功，一般不出问题。自己在安装时出问题如下：  
    错误1：配置错误

     在./configure时出错：gmake：CC：cammond not found

     出错原因：因为设置编译器为CC，但实际编译器是gcc和g++。CC没有link到gcc和g++。

    解决办法：

    在 /etc/profile（全局起作用）或者 $HOME/.profile(相应用户起作用)文件中添加如下脚本

1. PATH=/usr/ccs/bin:.:/usr/sbin:/usr/bin:/usr/ucb:/etc:.:/opt/SUNWspro/bin:/usr/ccs/bin:/usr/openwin/bin:
2. export PATH
3. QTDIR=/usr/local/Qt-4.8.3
4. export QTDIR
5. PATH=$PATH:$QTDIR/bin
6. export PATH
7. QMAKESPEC=/opt/qt-x11-4.8.3/mkspecs/solaris-g++
8. export QMAKESPE

QMAKESPEC=/qt源码包存放目录/makespecs/选定的编译器， 注意要与solaris系统的编译器对应。如：用solaris-cc就如上报错，改为solaris-g++则通过 ./configure。

    错误2：gmake错误

     在solaris上使用gmake，但是到一半时出错：

**[html]** [view plaincopy](http://blog.csdn.net/ilearncpp/article/details/8235612)

1. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c: In function `HB\_Error hb\_buffer\_ensure(HB\_BufferRec\_\*, HB\_UInt)':
2. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:77: error: invalid conversion from `void\*' to `HB\_PositionRec\_\*'
3. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:81: error: invalid conversion from `void\*' to `HB\_GlyphItemRec\_\*'
4. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:86: error: invalid conversion from `void\*' to `HB\_GlyphItemRec\_\*'
5. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:97: error: invalid conversion from `void\*' to `HB\_GlyphItemRec\_\*'
6. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c: In function `HB\_Error hb\_buffer\_duplicate\_out\_buffer(HB\_BufferRec\_\*)':
7. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:115: error: invalid conversion from `void\*' to `HB\_GlyphItemRec\_\*'
8. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c: In function `HB\_Error hb\_buffer\_new(HB\_BufferRec\_\*\*)':
9. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:134: error: invalid conversion from `void\*' to `HB\_BufferRec\_\*'
10. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c: In function `HB\_Error \_hb\_buffer\_clear\_positions(HB\_BufferRec\_\*)':
11. ../3rdparty/harfbuzz/src/harfbuzz-buffer.c:215: error: invalid conversion from `void\*' to `HB\_PositionRec\_\*'
12. gmake[1]: \*\*\* [.obj/release-shared/harfbuzz-buffer.o] Error 1
13. gmake[1]: Leaving directory `/opt/qt-x11-4.8.3/src/corelib'
14. gmake: \*\*\* [sub-corelib-make\_default-ordered] Error 2

    原因：

   1  源代码文件 /opt/qt-x11-4.8.3/src/corelib 的makefile出错  ，从gmake进入或离开目录得知（make[1]: Leaving directory `/opt/qt-x11-4.8.3/src/corelib'），

1. CC=g++
2. CXX=g++

如此，编译c文件就用g++编译，但c++有严格的类型检查，因此出现如上错误。另外，在进入opt/qt-x11-4.8.3/src/gui时，也是同样的问题，同样修改makefile解决。

    2  可能是gcc版本原因，solaris 10自带gcc 3.4.3，重新安装了gcc3.4.6

   解决办法：

   1 修改相应目录下的makefile文件，使得 用gcc编译c源文件

1. CC=gcc
2. CXX=g++

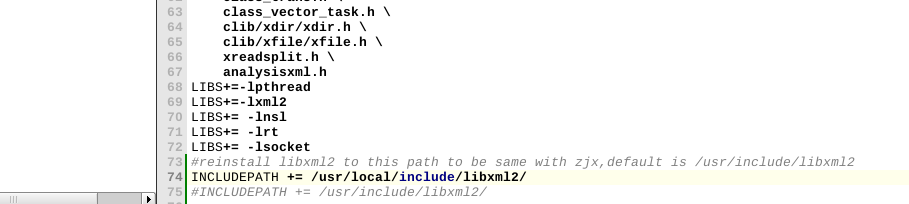
   2 下载gcc3.4.6源码，并安装，不要用 gcc3.4.3。 （可能用gcc3.4.3也能通过，这个不确定，可以试试）

## Qt移植,先把gdb装好再移植

把/usr/local/Trolxx/qt 文件夹直接拷贝过去就行了

/xlfd2的qt文件夹拷贝过去,然后crle -u那些缺的库就行了

## 安装libxml2



1. ./configure --prefix=/usr/local

装完之后就是/usr/local/include/libxml2

## 安装git

Install git on Solaris 10 and 11:

pkgadd -d http://get.opencsw.org/now

/opt/csw/bin/pkgutil -U

/opt/csw/bin/pkgutil -y -i git

/usr/sbin/pkgchk -L CSWgit # list files

## Gdb安装

类似gunzip gdb-6.8-sol10-x86-local.gz

pkgadd -d gdb-6.8-sol10-x86-local

在/etc/profile中修改追加环境变量/usr/loca/bin目录

然后命令行打开gdb,发现缺库,把那几个除了gcc的都装上

缺gcc的用当前qt用的gcc的库去找

Find找也行,一般是/usr/sfw/lib

Crl -u那个

这样gdb在命令行就可以使用了

## vim安装

加上 /usr/local/bin目录

安装7.2那个,然后

cp /usr/local/bin/vim /usr/bin

然后在~目录新建.vimrc

加入

set nocompatible,这是为了防止方向键混乱

对于root,/就是solaris的根目录

## solaris做服务端速度慢:

TCP参数调整:

ndd –set /dev/tcp tcp\_recv\_hiwat 65535000

## 普通用户目录

/export/home/sjcs

## Solaris11 安装gcc

pkg install --be-name gcc-be gcc-3

之后重启到be就有了

## 关于补全

其实，这不是系统之间的区别，而是因为是sloaris的初始shell的问题。只要把他改成bash就好了。  
  
最简单的办法是，直接运行bash命令。但是再打开终端的时候又会回复的原来的样子。  
一劳永逸的办法：修改/etc/passwd中，属于你的用户名的那一行中的shell为bash，这样当你以后登陆的时候就都是bash了！

## 解压tar.gz

**gzcat filename.tar.gz | tar xvf –**

压缩

先把文件或文件夹打成tar包  
tar -cvf xxx.tar ./src ./[bin](https://www.baidu.com/s?wd=bin&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1dBm1RLnj0sPvnkuyPWmyR30ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWcYPj6snWm3)  
压缩tar包  
gzip xxx.tar  
会生成xxx.tar.gz的压缩文件

## 解压gz

gunzip

## gcc,g++,make

最好是用2.5解决

cp /usr/sfw/bin/gcc /usr/bin/

cp /usr/sfw/bin/g++ /usr/bin/

cp /usr/ccs/bin/make /usr/bin/

或

ln -sf /usr/sfw/bin/g++ /usr/bin/g++

## 安装软件

pkgadd -d

### iperf

官网下载安装包(solaris版的),然后这样装就行了

## 卸载软件

以卸载pidgin为例，在启动菜单的pidgin启动项邮件属性面板中，看到命令为pidgin。

需要先定位程序所在位置：

        # which pidgin

         /usr/bin/pidgin

查看软件包信息：

        # pkgchk -lp /usr/bin/pidgin

         Pathname: /usr/bin/pidgin

         Type: regular file

         Expected mode: 0755

         Expected owner: root

         Expected group: other

         Expected file size (bytes): 910236

         Expected sum(1) of contents: 27126

         Expected last modification:  3月 08 03时33分12秒 2011

         Referenced by the following packages:

                 SUNWgnome-im-client

        Current status: installed

卸载软件包：

  www.2cto.com

        # pkgrm SUNWgnome-im-client