

q30的专栏

欢迎 huapuyu6! 退出 我的博客 配置 写文章 文章管理 博客首页

□□

空间

博客

好友

相册

留言

用户操作

[留言] [发消息] [加为好友]

订阅我的博客



q30的公告

文章分类

- C
- Diary
- ibm AIX
- Oracle
- True64
- Work blog
- 生活

电子产品

PCPOP报价

太平洋报价网

原 AIX开发环境 Visual Age C++6.0 收藏

随着IBM pSeries服务器市场占有率的不断增加，AIX操作系统也为越来越多的IT业者所喜爱。AIX作为UNIX家族中出色的一员，有着丰富的开发环境，AIX最新的版本为AIX 5L版本2，最常见的版本为4.3.3，以5L版本2为例，缺省安装直接支持J2EE，这里主要介绍AIX系统的C/C++开发部分。

首先应当清楚，AIX并不集成C/C++，要使用这部分功能，必须安装IBM C或者Visual Age C++(包含C编译器)，这篇文章只介绍Visual Age C++，将按照Visual Age C++的安装、配置、编译器的使用、程序调试的顺序介绍，使用的平台为AIX 5L。

安装Visual Age C++

Visual Age C++要在AIX 5L上使用，最低的版本为5.0.2，现在最新的版本为6.0。Visual Age C++是一个功能强大的集成开发环境，和Windows平台的Visual C++一样，从项目的管理、编辑、编译、调试等各方面，提供一系列完善的图形化的操作界面。要完全安装(包含IDE图形界面)Visual Age C++，可遵循以下步骤：

1) 确定IDE图形界面需要的Motif文件集已经安装，命令如下：

```
lspp -l X11.compat.adt.Motif12
```

如没有安装，应先从AIX操作系统安装介质安装这一文件集。

2) 将Visual Age C++的安装介质放入机器的光驱中

注意，如果是Visual Age C++的产品盘，并不需要mount CD盘

3) 开始安装Visual Age C++文件集，命令如下：

```
smitty install_latest
```

安装目录选择光区，使用F4键选择，如下，

如需要选择安装部分文件集，可在_all_latest菜单项下按F4键，

使用F7键选择要安装的文件集，按回车键即可。

上面介绍的是是一种完整的安装方式，如果只是使用C/C++的编译器，不许要安装Visual Age C++的图形界面，可借助于下面的命令：

```
installp -aXgd /dev/cd0 vacpp.cmp
```

这条命令的运行结果是安装我们需要的所有文件集。

如果所使用的安装介质并不是光盘，而是VACPP6.tar.Z这样的安装介质。可使用如下的方法进行安装。

个人信息

IBM pSeries 和 AIX 信息中心

QT参考文档

我的宝宝

我的日历

我老婆地址

存档

2006年03月(1)

2006年02月(10)

2005年10月(2)

2005年09月(4)

2005年08月(9)

2005年07月(2)

2005年03月(1)

2005年02月(2)

2005年01月(3)

2004年12月(3)

2004年11月(1)

2004年09月(25)

2004年08月(48)

2004年07月(2)

通过FTP的途径将VACPP6.tar.Z上传到目标SERVER.随后可以选择安装目录。

我们选择将VACPP6.tar.Z上传到目标SERVER的 /home目录下,并用uncompress命令解压缩。然后在/ home下建立一个临时目录。我们命名为VAC,(mkdir VAC).

在/home/VAC下执行tar -xf ../VACPP6.tar命令。

最后,在/home/vac/usr/sys/inst.images/下执行smitty installp,进行安装。

配置Visual Age C++

正确安装完Visual Age C++以后,还不能立刻使用,需要配置使用许可才可以正确使用。Visual Age C++的使用许可通过LUM (License Use Management) 系统管理,所以正确的配置方法是先配置LUM,再配置Visual Age C++的使用许可。

第一步: 配置LUM

LUM在AIX 4.3.3及5L版本中是一个必须的部件,系统的初始安装过程中已经安装,但是并没有配置,使用i4cfg命令配置,如下:

```
#cd /usr/opt/ifor/bin
```

```
#!/i4cfg -script
```

这条命令将配置LUM成为并发的节点锁定服务器工作方式 (Concurrent Nodelock Server) , 如果没有错误输出, 下一步是启动LUM服务器:

```
#!/i4cfg -start
```

之后,可以用以下命令检测LUM服务器的工作状态:

```
#!/i4cfg -list
```

第二步: 配置Visual Age C++的使用许可

正确安装以后,可以在/usr/vac和/usr/vacpp两个目录下看到后缀为lic的文件,这些文件记载了Visual Age C++的使用许可协议,需要做的就是将需要的许可协议添加到LUM的管理中。使用的命令为i4blt,使用方法为:

```
i4blt -a -f license_file_name -R u -T license_num
```

具体过程如下:

首先添加C的使用许可,

```
#cd /usr/opt/ifor/bin
```

```
#!/i4blt -a -f /usr/vac/cforaix_cn.lic -R u -T 10
```

然后再配置C++的使用许可,

```
#!/i4blt -a -f /usr/vacpp/vacpp_cn.lic -R u -T 10
```

在上面的例子中配置的许可数为10,正确的配置完后可通过如下命令验证许可数是否正确或许可协议有没有正确的被添加,

```
i4blt -sl cn
```

注意以上的配置是在AIX 5L版本上进行的，如在AIX 4.3.3版本上，则i4cfg、i4blt命令存在/var/ibm/bin目录下。

Visual Age C++ 的使用

安装、配置以后，可以使用C/C++编译器了，但当你运行cc时却发现，cc没有找到，不必怀疑安装的正确性，这是没有设置环境变量的缘故，在/usr/vac/bin目录下有一个脚本文件replaceCSET，它会完成这一切。

可以用一个最经典的小程序hello.c来测试一下编译器的正确性，程序内容如下：

```
#include <stdio.h>
```

```
main() {  
    printf("Hello!\n");  
}
```

编译、连接：

```
cc -o hello hello.c
```

将生成hello可执行程序。

一般情况下，我们习惯直接使用cc编译程序，实际上AIX的C/C++编译器还提供了其他的一些使用方法，如后缀为_r就有xlc_r、xlc_r，这里我们做一个完整的介绍。

AIX平台上缺省的C++编译器为xlc，而C的编译器有三个：

cc: 这是最习惯用的，扩展模式的C编译器

xlc: ANSI C编译器，使用UNIX的头文件

c89: ANSI C编译器，使用ANSI头文件

具体选择何种编译器，需根据程序的情况来选择，一般我们可以使用cc。另外，还有一些后缀供我们选择，结合以上的基本编译器，给我们提供了更大的选择面，如下：

_r	连接UNIX98标准线程库
_r4	连接POSIX Draft4 (DCE) 标准线程库
_r7	连接POSIX Draft7标准线程库
128	Double型类型为128位并使用连接相关的库
128_r	Double类型为128位并使用UNIX98线程库
128_r4	Double类型为128为并使用连接POSIX Draft4线程库
128_r7	Double类型为128位并使用连接POSIX Draft7线程库

另外，UNIX98已经兼容涵盖了UNIX95、POSIX标准的内容，所以，如果程序中使用到线程函数，只许增加后缀_r就可以了。举例说，编译连接ANSI C并使用POSIX线程库，可以使用xlc_r编译器，当然也可以使用xlc -lpthread的形式。

这里提到的只是最基本的东西，还有很多方面的内容并没有介绍到，例如如何生成使用动态库，这些更详细的东西需要参考相关的手册，或拨打IBM技术支持电话寻求技术支持。

程序调试

调试程序比较复杂，一方面需要我们借助一些调试工具，如**dbx**、**truss**、**tprof**等，另一方面也需要我们具有一些调试经验。

这里介绍一个用**dbx**调试运行中的程序的方法，重点在于调试步骤，而不在于**dbx**工具的使用，以一个简单的程序为例，程序为：

```
#include <stdio.h>

main() {
    long count=0;
    int loop_count;

    while(1) {
        count *= (count++);
        printf(" Test debug, count : %ld\n", count);
        for (loop_count = 0; loop_count < 10; loop_count++) {
            sleep(1);
        }
    }
}
```

程序名为**dbg.c**

调试步骤如下：

1) 编译**dbg.c**

```
xlc -g -o dbg dbg.c
```

2) 运行**dbg**程序

```
./dbg
```

3) 查看**dbg**程序的进程ID

```
ps -ef | grep dbg | grep -v grep
```

假设**dbg**的PID为12345

4) 调试**dbg**程序

```
dbx -a 12345
```

5) 这时我们运行**dbx**的命令调试**dbg**程序了，下面是几个常用的命令：

step：单步运行

stop at line_number : 在某一行设置断点
stop in subroutine_name: 设置断点到某一个函数
cont : 继续运行

6) 退出调试

注意退出时应使用**detach**而不是**exit**，否则会影响程序的运行，导致程序退出。

这篇文章只是简单介绍**AIX**的开发环境，实际工作中用到的内容将远远不止这些知识，需要开发者查阅大量的相关资料，或访问以下网址：

<http://www-900.ibm.com/developerWorks/cn/index.shtml>

发表于 @ 2004年09月21日 10:03:00 | [评论\(0\)](#) | [举报](#) | [收藏](#)

旧一篇:ORACLE网络与安全 FAQ | 新一篇:vac6.0 for aix 安装

发表评论 [“评论王争夺赛”活动，第4期开始啦！](#)

表情:



评论内容:

用户名: huapuyu6

匿名评论