

VirtualBox 安装 Solaris 11 "Nevada" 操作系统

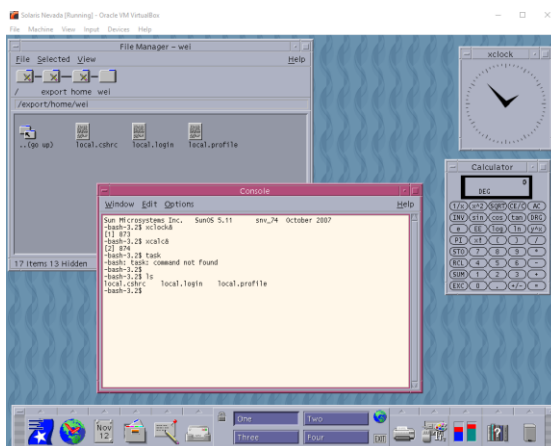
William Nov. 12, 2022

安装Solaris 操作系统看似容易实际上很麻烦。网上下载的大部分 Solaris*.iso 最后都装成了一个 Console Interface. 要启动 Graphics Interface 是麻烦事情, 要做很多操作 (见 Appendix: Install Solaris-Desktop). 终于找到了一个 Solaris 11, “Nevada” Version. 一开始安装就进入 Graphics Interface.

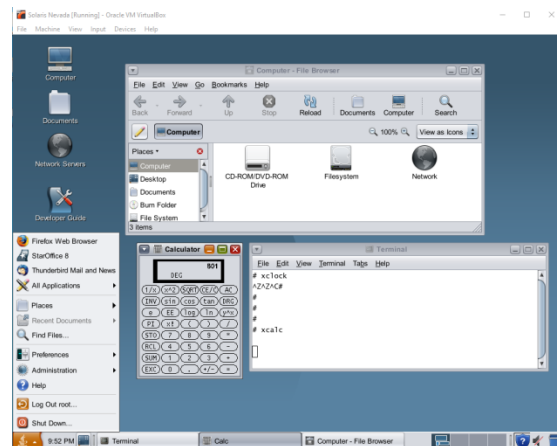
Solaris 11 "Nevada" Build 74 (x86)

<https://archive.org/details/sol-nv-b74-x86>

Solaris 11, Nevada 有两种 Graphics Interface, GNOME & CDE. 在Login 的时候可以选择使用 GNOME 或者 CDE. “Nevada” 还带有办公室软件, StarOffice 8.

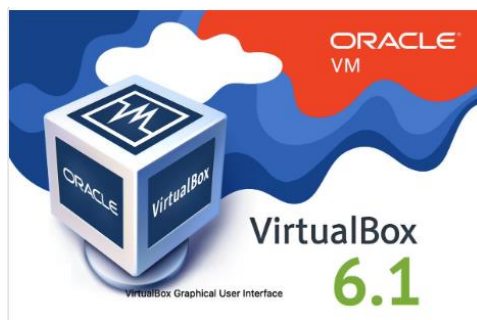


GNO Graphics Interface



DCE Graphics Interface

I. VirtualBox 6.1

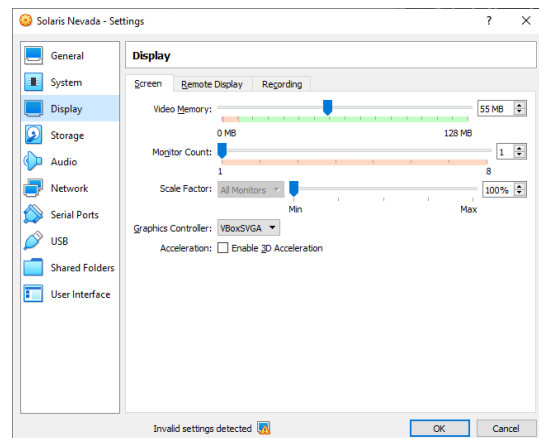
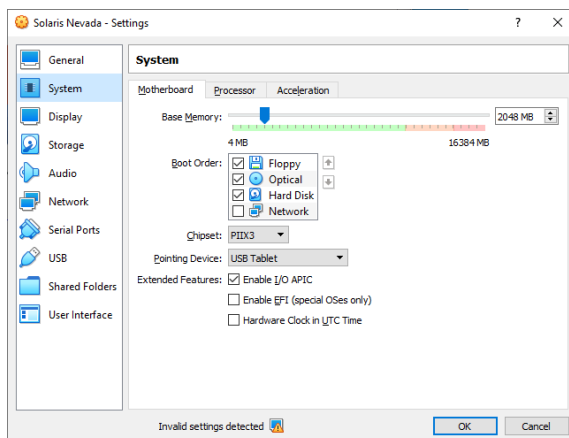
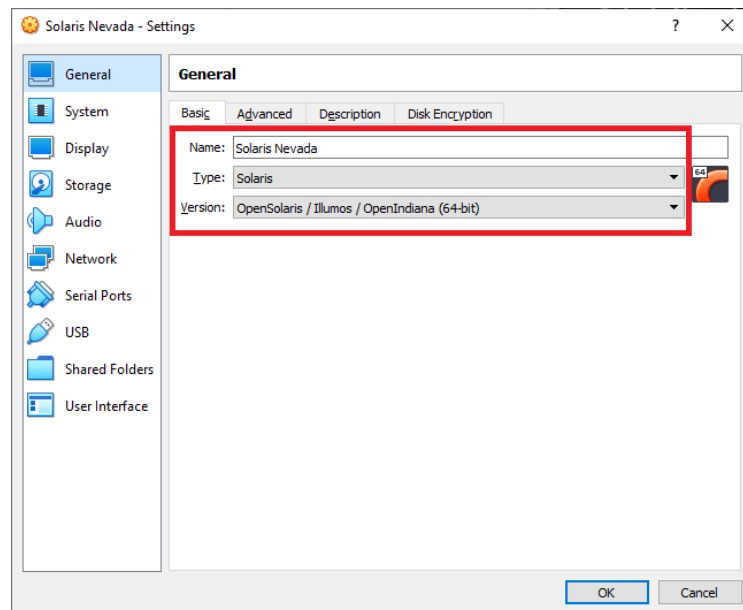


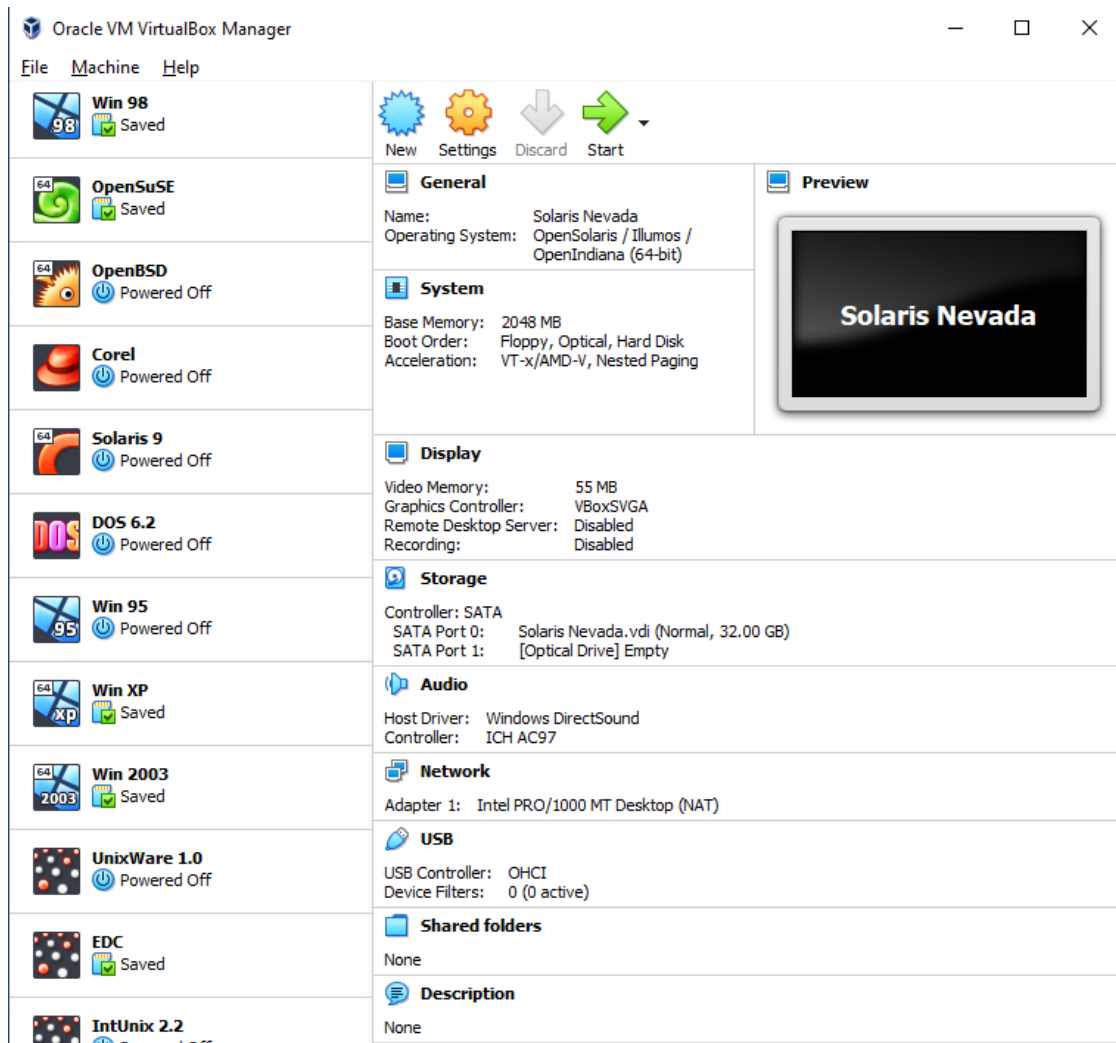
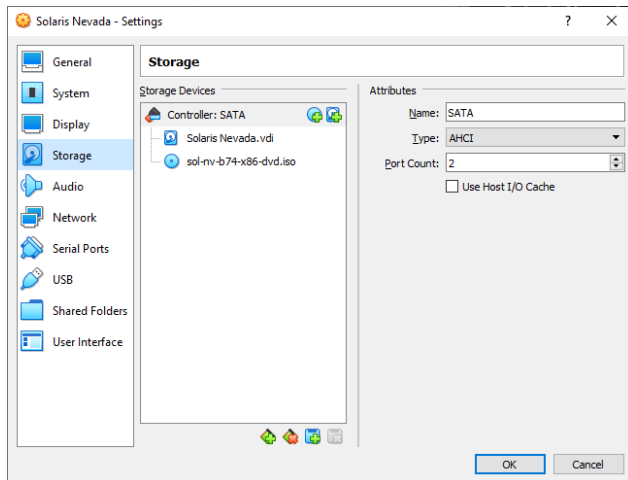
II. Download ISO File

[sol-nv-b74-x86-dvd.iso](#) 3,482,880 KB

III. Settings

如果 Set 名字中间有 Solaris, Nevada, 下面两项就自动设成如图的样子: Solaris, OpenSolaris/Illumos/OpenIndiana(64-bit).

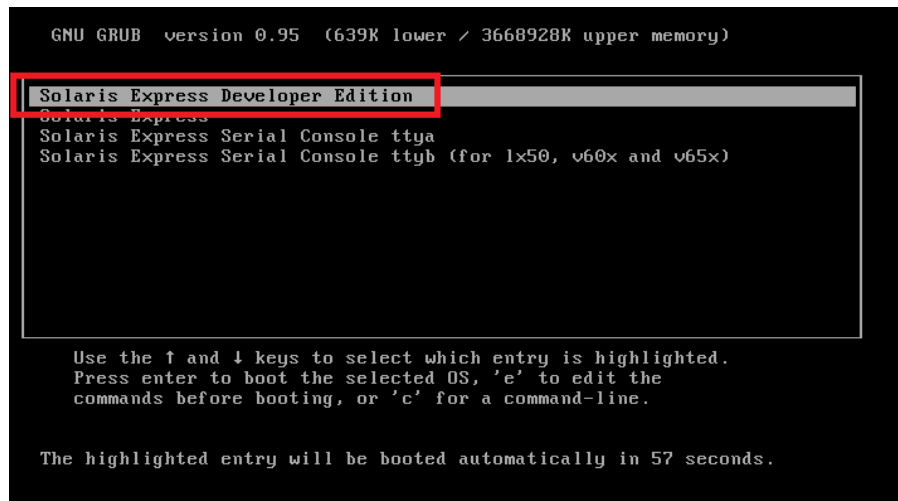




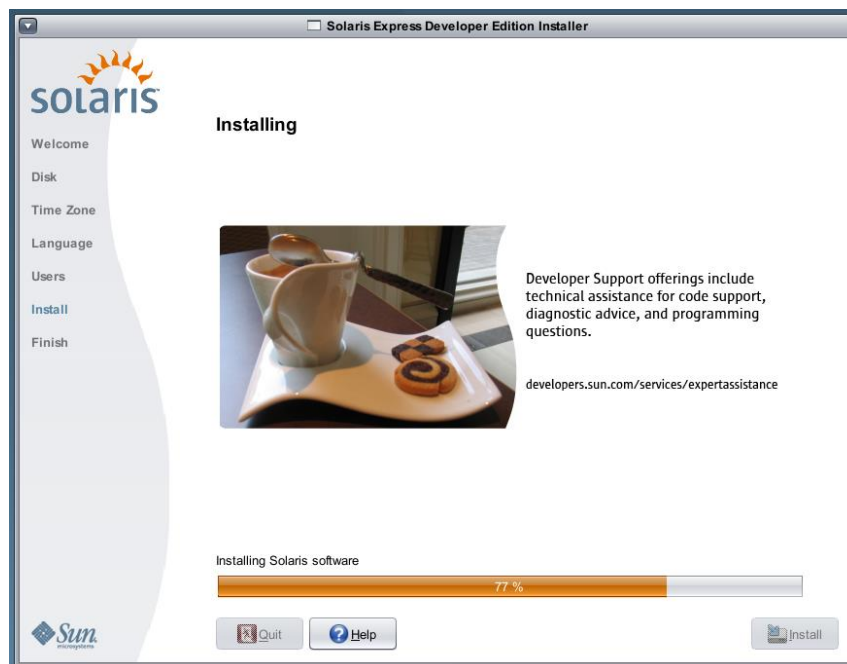
IV. Install Solaris 11, “Nevada”

把CD ROM设为文件 `sol-nv-b74-x86-dvd.iso` 就能自动Boot.

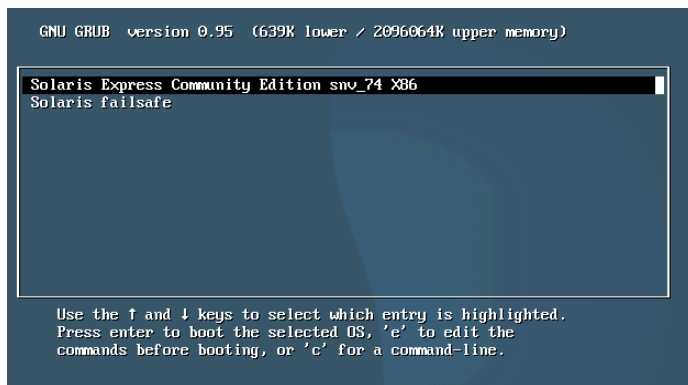
下图是关键的选择点: “Solaris Express Developer Edition”



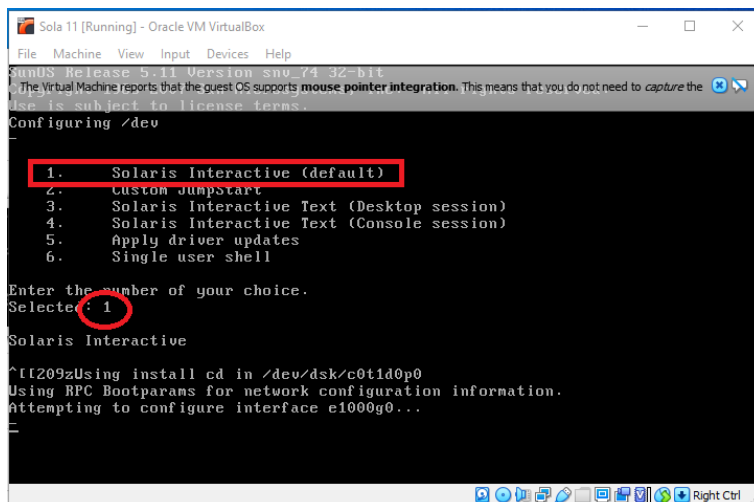
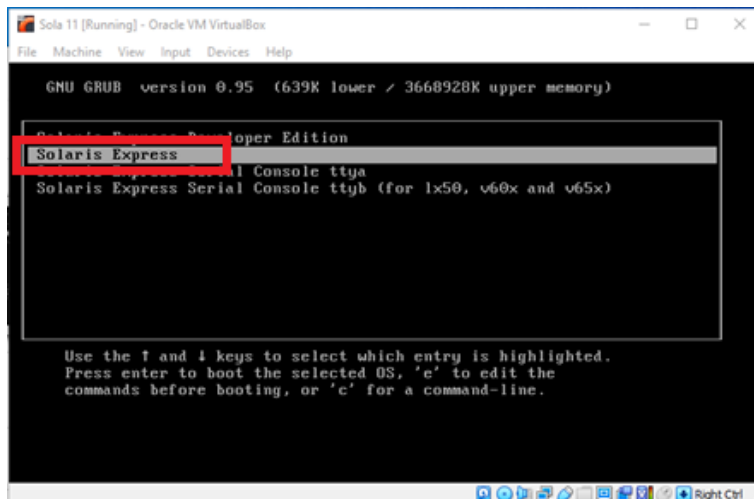
以后每一项都选择 Default, 可以自动安装完毕.



安装完毕, 去掉 ISO文件, Reboot, 可以看见下面的屏幕.



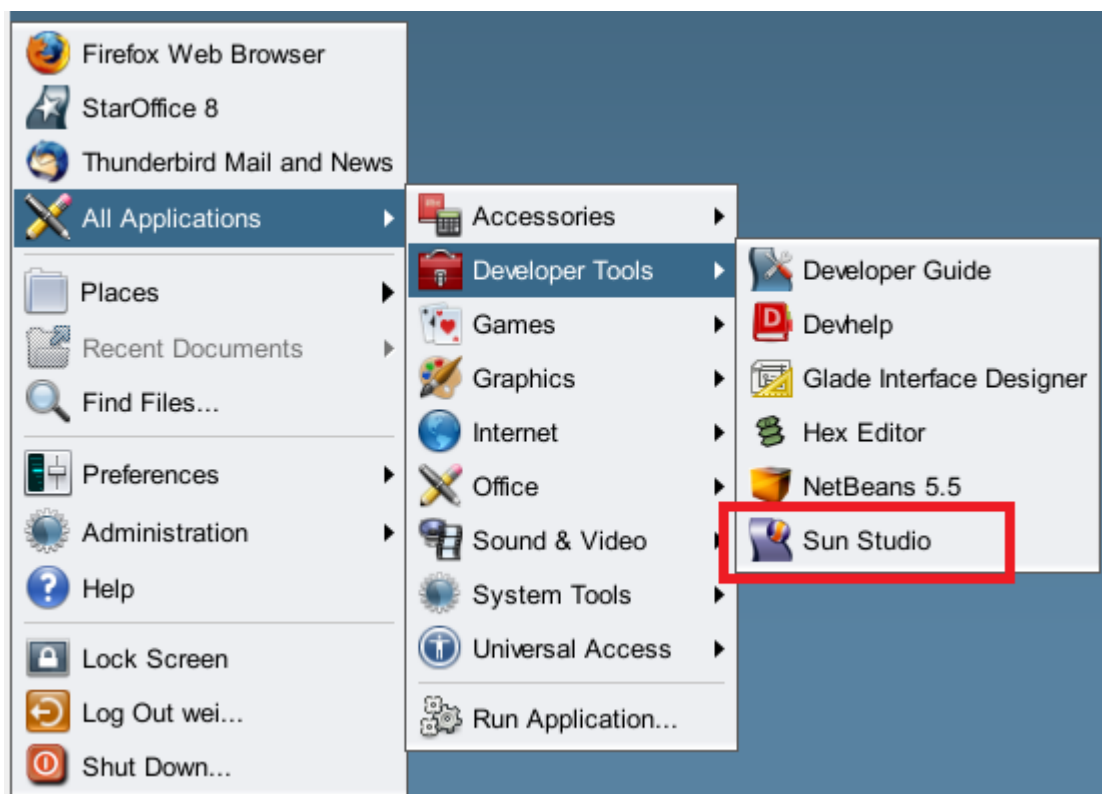
也可以不装 Developer Software, 可以节省许多 Disk Space. 下图是关键的两个选择点:

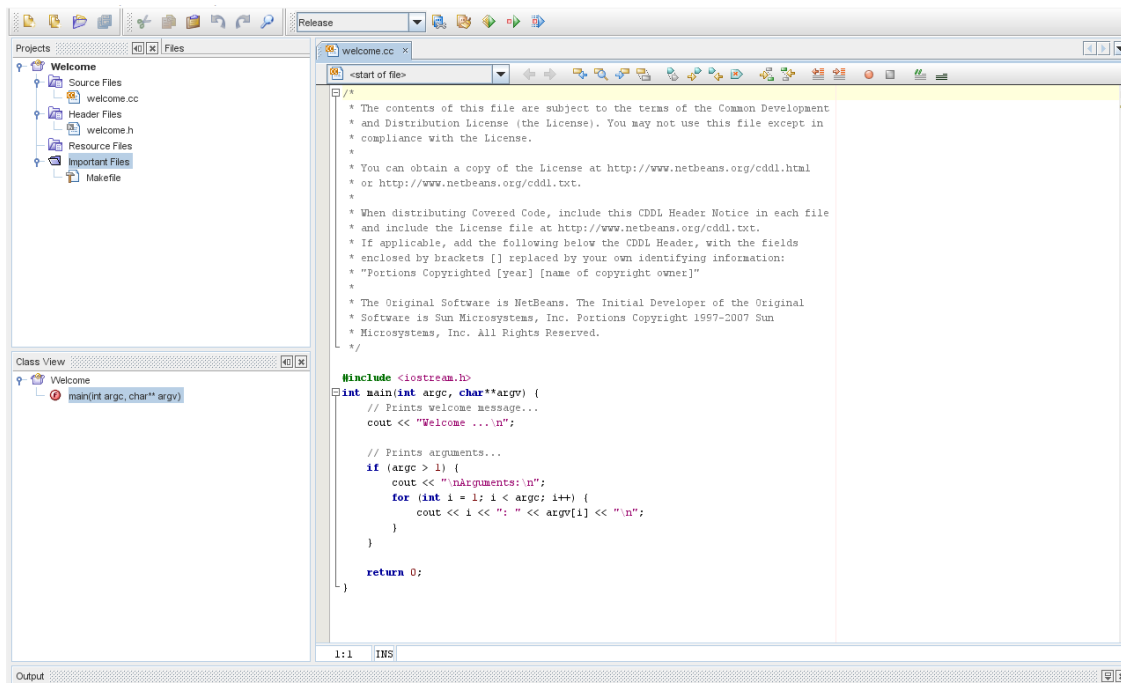


以后每一项都选择 Default，可以自动安装完毕。只是中间有几项等待时间比较长。但是最终是能够安装完毕的。

V. Sun Studio

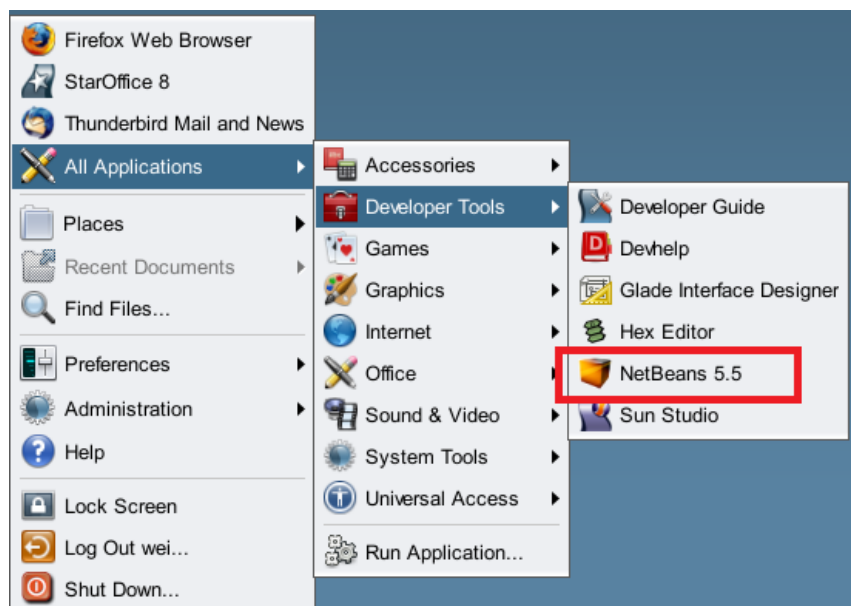
如果装了 Developer Software，就能看到下面的工具 Sun Studio，非常有用。可以用作开发 C，C++，Fortran 以及其他语言的工具。

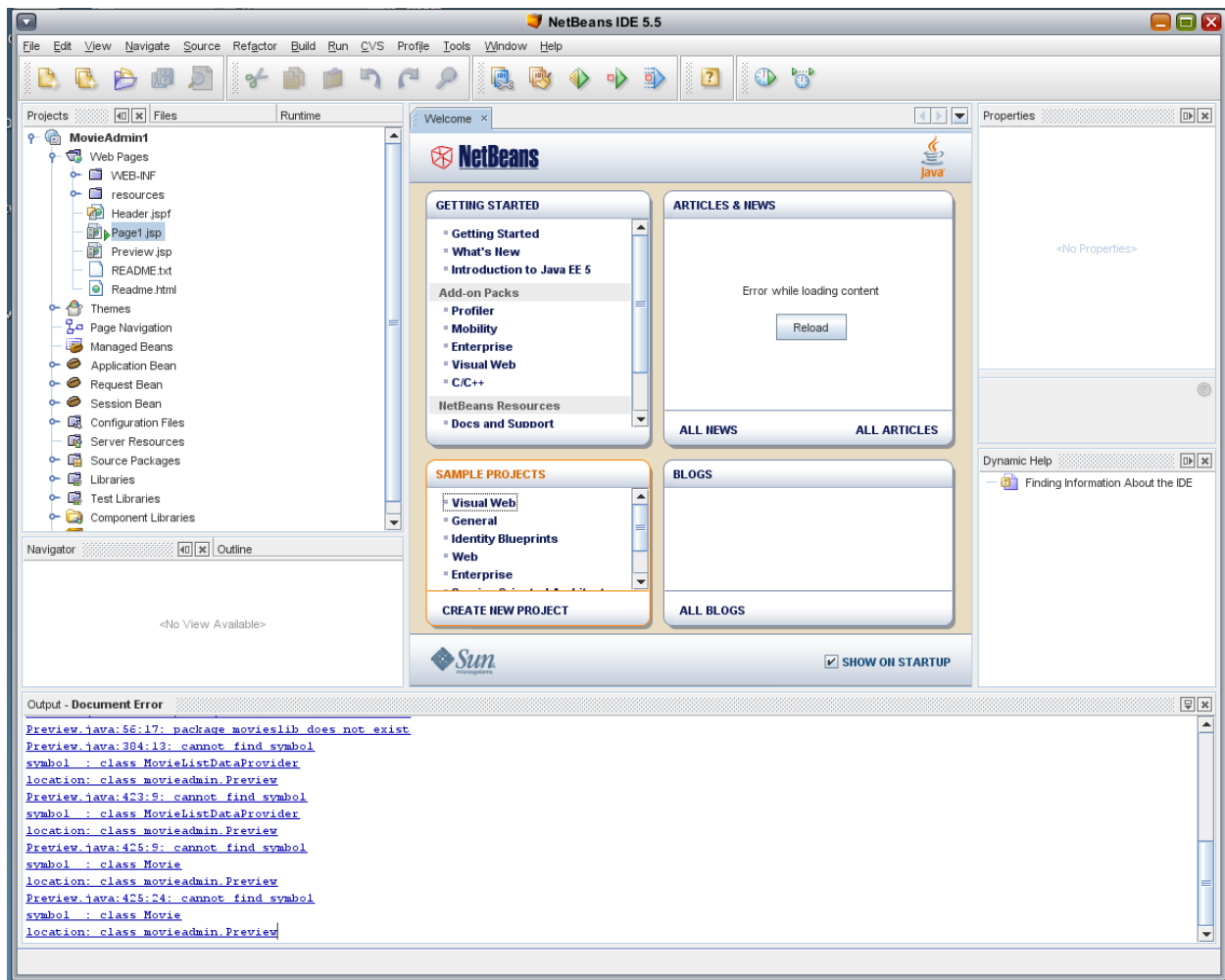




VI. NetBeans

如果装了 Developer Software, 也能看到下面的工具 NetBeans 5.5. NetBeans IDE is a leading IDE from Oracle Corp, primarily aimed at Java developers. It is suitable for the rapid development/deployment of desktop, mobile and web applications that may be based on Java.





Appendix: Install Solaris-Desktop

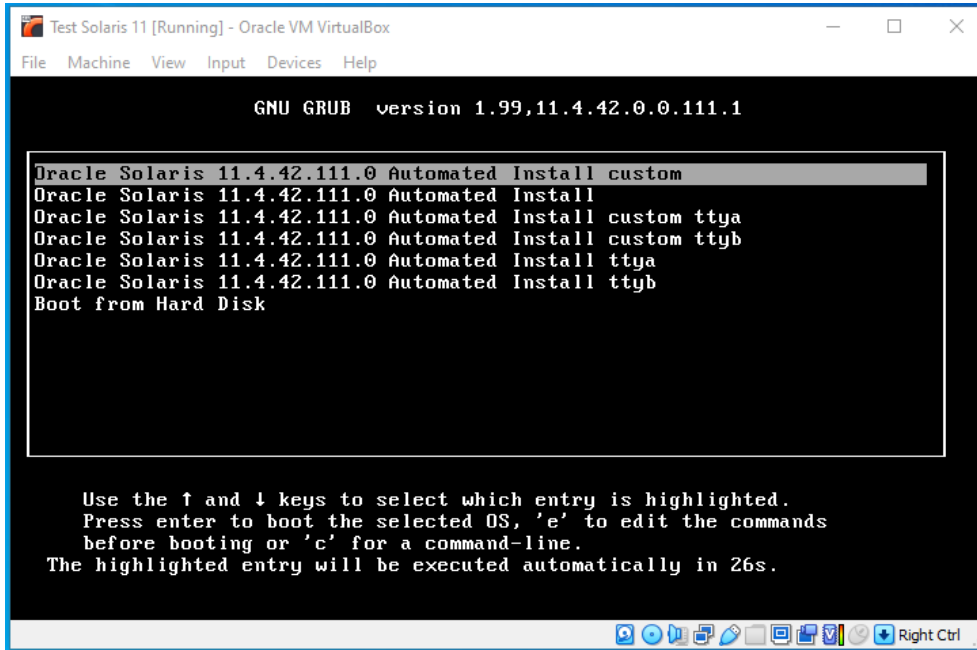
1) Solaris 11.4.42 ISO

Solaris-11.iso 457 MB

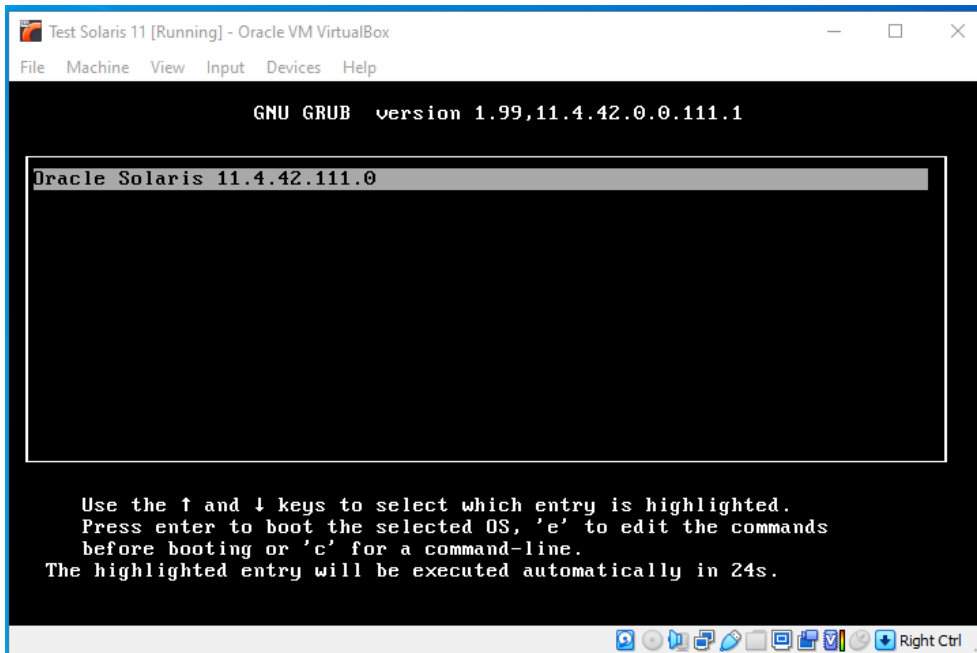
<https://www.oracle.com/solaris/solaris11/downloads/solaris-downloads.html#>

这个iso文件装好 OS 以后只显示 Console Login 所以必须要自行安装 Desktop.

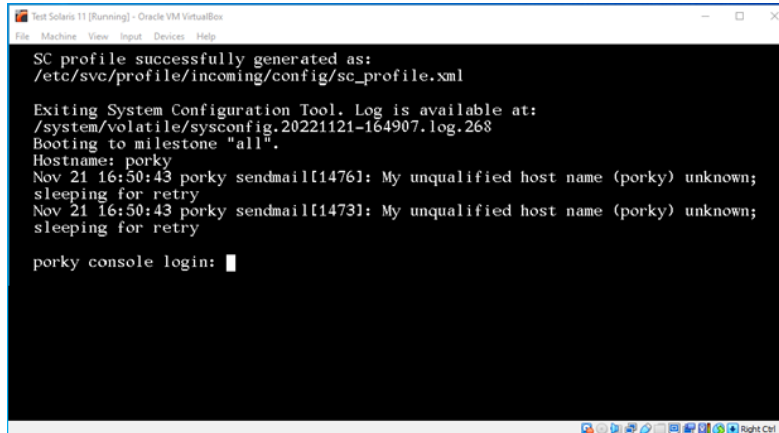
2) Boot from iso file



3) Installation Completed. Be-boot System.



4) See Console Login Screen



这时候看到的的就是 Console Screen. 下面开始安装 Solaris Desktop.

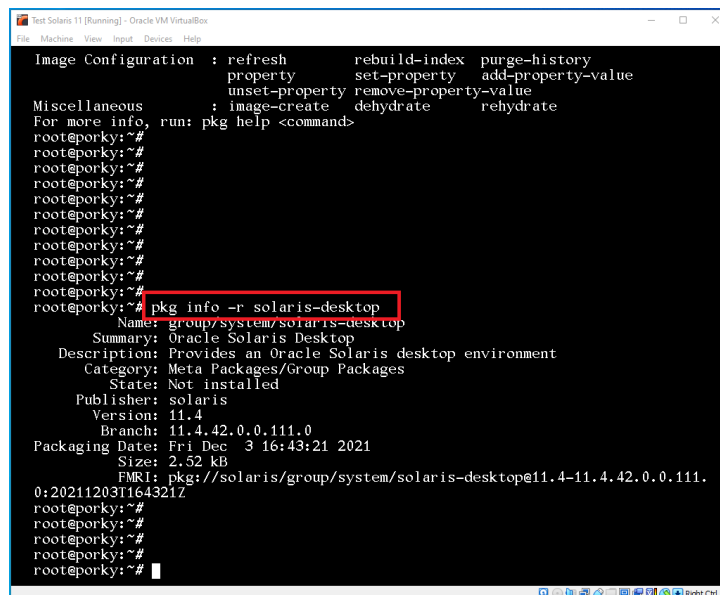
5) Login as wei/weixu826, and type:

su - <CR>

Root password: root-root <CR>

6) Check Solaris-desktop pkg, type:

pkg info -r solaris-desktop <CR>



可以看到这个软件包是存在的

7) 开始安装: `pkg install solaris-desktop <CR>`

The screenshot shows a terminal window titled "Test Solaris 11 [Running] - Oracle VM VirtualBox". The terminal output shows a series of "root@porky:~#" prompts. The command "pkg install solaris-desktop" is entered and highlighted with a red box. Below it, the following installation details are displayed:

```

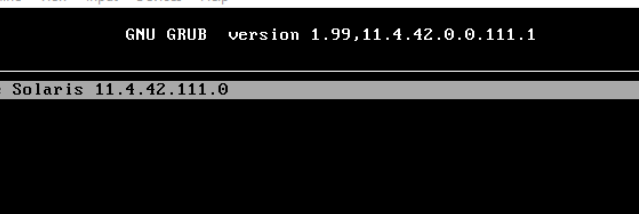
Packages to install: 283
Mediators to change: 1
Services to change: 13
Create boot environment: No
Create backup boot environment: No

```

At the bottom of the terminal, a table shows the download progress for the packages:

DOWNLOAD	PKGS	FILES	XFER (MB)	SPEED
web/browser/firefox	223/283	60944/62768	848.1/893.1	6.9M/s

8) 装完以后 Re-Boot System.



Test Solaris 11 [Running] - Oracle VM VirtualBox

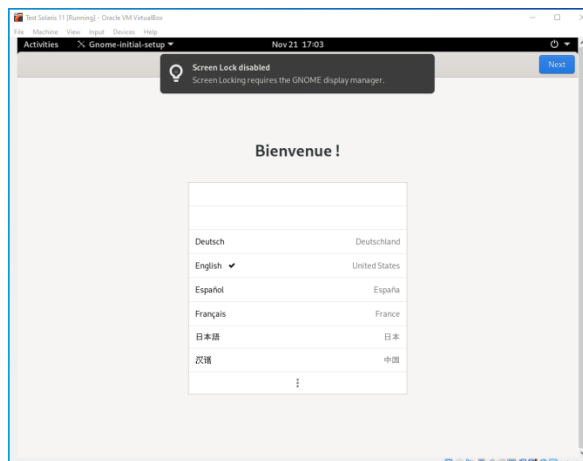
File Machine View Input Devices Help

GNU GRUB version 1.99,11.4.42.0.0.111.1

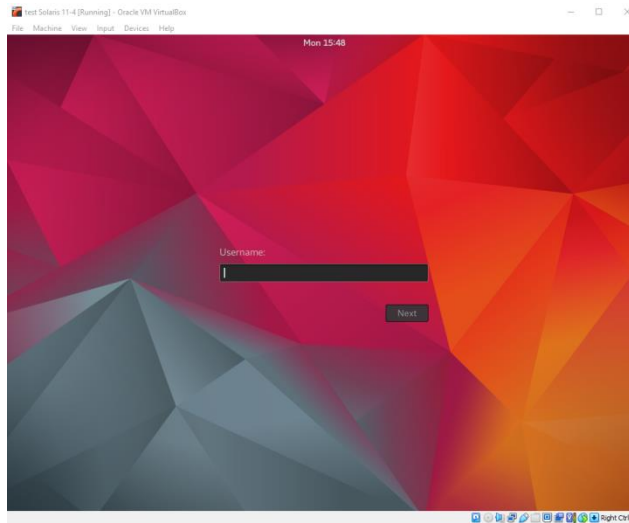
Oracle Solaris 11.4.42.111.0

Use the ↑ and ↓ keys to select which entry is highlighted.
 Press enter to boot the selected OS, 'e' to edit the commands
 before booting or 'c' for a command-line.
 The highlighted entry will be executed automatically in 24s.

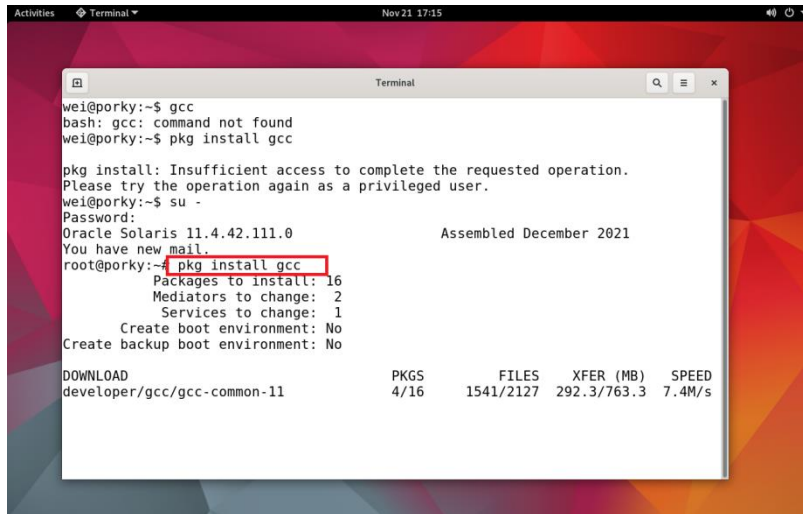
9) 系统初始化



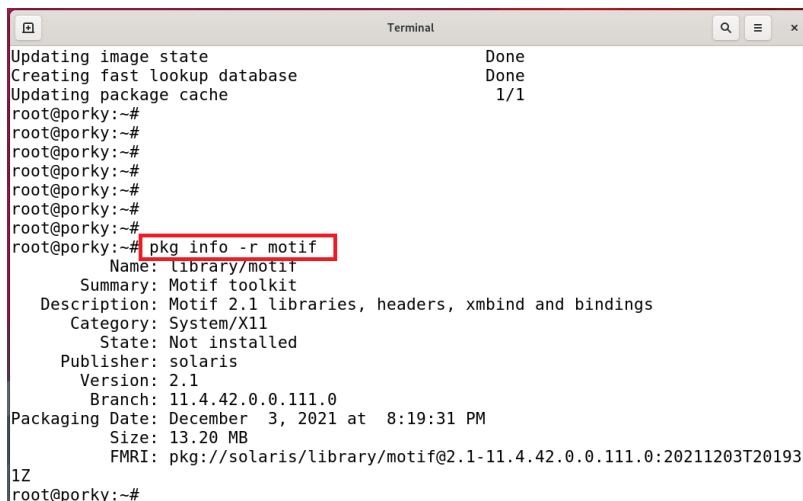
10) Solaris Desktop 安装完毕



11) Install gcc Develop Package: `pkg install gcc <CR>`



12) Install Motif Develop Package: `pkg install motif <CR>`



13) Test Motif

The complete program listing for the `push.c` program is as follows:

```
#include <Xm/Xm.h>
#include <Xm/PushB.h>

/* Prototype Callback function */

void pushed_fn(Widget , XtPointer ,
               XmPushButtonCallbackStruct *);

main(int argc, char **argv)
{
    Widget top_wid, button;
    XtAppContext app;

    top_wid = XtVaAppInitialize(&app, "Push", NULL, 0,
                               &argc, argv, NULL, NULL);

    button = XmCreatePushButton(top_wid, "Push_me", NULL, 0);
    /* tell Xt to manage button */
                                XtManageChild(button);

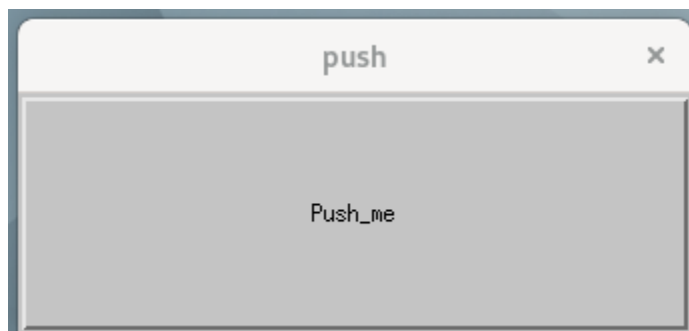
                                /* attach fn to widget */
    XtAddCallback(button, XmNactivateCallback, pushed_fn, NULL);

    XtRealizeWidget(top_wid); /* display widget hierarchy */
    XtAppMainLoop(app); /* enter processing loop */
}

void pushed_fn(Widget w, XtPointer client_data,
               XmPushButtonCallbackStruct *cbs)
{
    printf("Don't Push Me!!\n");
}
```

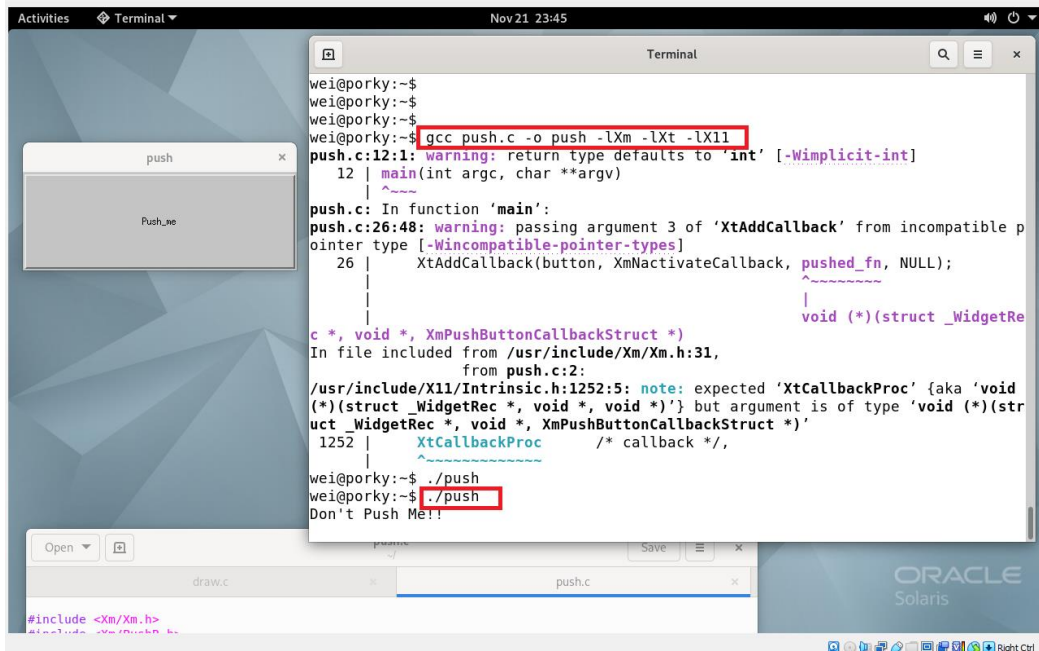
Build: `gcc push.c -o push -lXm -lXt -lX11 <CR>`

Run: `./push <CR>`



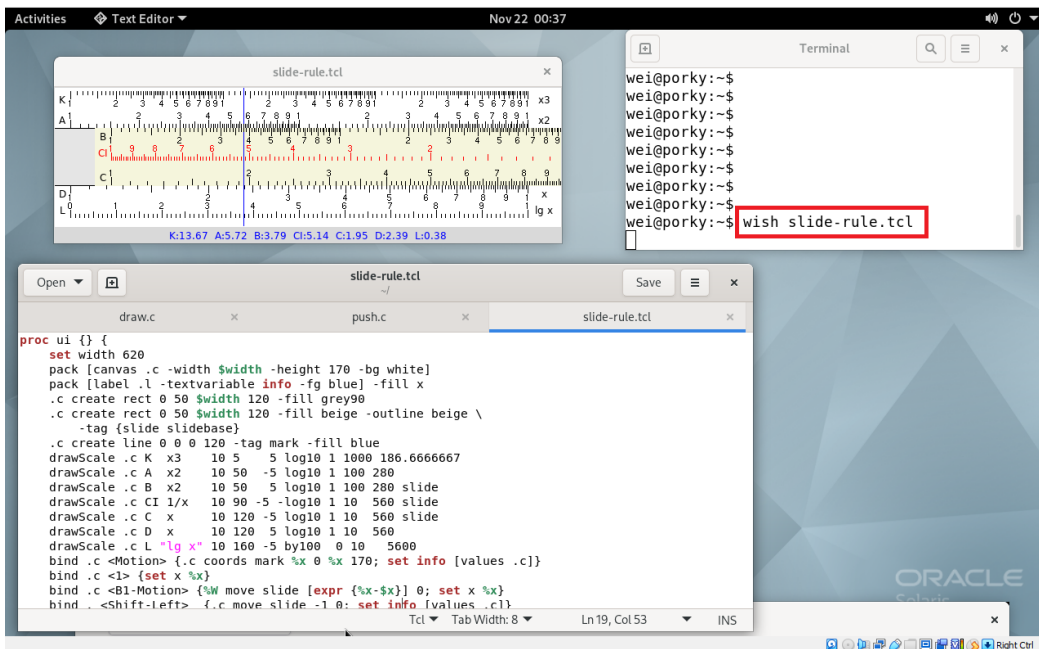
14) Solaris Screen

现在的 gcc 和早年的 cc 有些不同，所以编译的时候产生了很多 Warning Message，但最后编译还是成功了。



15) Run Tk/Tcl

Solaris 11.4.42 已经装有 Tk/Tcl. 下图是运行计算尺 slide-rule.tcl 程序: `wish slide-rule.tcl <CR>`



```

#!/bin/env tclsh
package require Tk
proc ui {} {
    set width 620
    pack [canvas .c -width $width -height 170 -bg white]
    pack [label .l -textvariable info -fg blue] -fill x
    .c create rect 0 50 $width 120 -fill grey90
    .c create rect 0 50 $width 120 -fill beige -outline beige \
    -tag {slide slidebase}
    .c create line 0 0 0 120 -tag mark -fill blue
    drawScale .c K x3 10 5 5 log10 1 1000 186.666667
    drawScale .c A x2 10 50 -5 log10 1 100 280
    drawScale .c B x2 10 50 5 log10 1 100 280 slide
    drawScale .c C l 1/x 10 90 -5 -log10 1 10 560 slide
    drawScale .c C x 10 120 -5 log10 1 10 560 slide
    drawScale .c D x 10 120 5 log10 1 10 560
    drawScale .c L "lg x" 10 160 -5 by100 0 10 5600
    bind .c <Motion> {c coords mark %x 0 %x 170; set info [values .c]}
    bind .c <1> {set x %x}
    bind .c <B1-Motion> {%W move slide [expr {%x-$x}] 0; set x %x}
    bind .c <Shift-Left> {c move slide -1 0; set info [values .c]}
    bind .c <Shift-Right> {c move slide 1 0; set info [values .c]}
    bind .c <Left> {c move mark -1 0; set info [values .c]}
    bind .c <Right> {c move mark 1 0; set info [values .c]}
}

proc drawScale {w name label x y dy f from to fac {tag {}}} {
    11
    set color [expr {[string match -* $f]? "red": "black"}]
    $w create text $x [expr $y+2*$dy] -text $name -tag $tag -fill $color
    $w create text 600 [expr $y+2*$dy] -text $label -tag $tag -fill $color
    set x [expr {[string match -* $f]? 580: $x+10}]
    set mod 5
    set lastlabel ""
    set lastx 0
    for {set i [expr {$from*10}]} {$i<=$to*10} {incr i} {
        if {$i>100} {
            if {$i%10} continue ;# coarser increments
            set mod 50
        }
        if {$i>1000} {
            if {$i%100} continue ;# coarser increments
            set mod 500
        }
        set x0 [expr $x+[$f [expr {$i/10.}]]*$fac]
        set y1 [expr {$i%(2*$mod)==0? $y+2.*$dy: $i%$mod==0? $y+1.7*$dy: $y+$dy}]
        set firstdigit [string index $i 0]
        if {$y1==-$y+$dy && abs($x0-$lastx)<2} continue
        set lastx $x0
        if {$i%($mod*2)==0 && $firstdigit != $lastlabel} {
            $w create text $x0 [expr $y+3*$dy] -text $firstdigit \
            -tag $tag -font {Helvetica 7} -fill $color
            set lastlabel $firstdigit
        }
        $w create line $x0 $y $x0 $y1 -tag $tag -fill $color
    }
}

proc values w {
    set x0 [lindex [$w coords slidebase] 0]
    set x1 [lindex [$w coords mark] 0]
    set lgx [expr {($x1-20)/560.}]
    set x [expr {pow(10,$lgx)}]
    set lgxs [expr {($x1-$x0-20)/560.}]
    set xs [expr {pow(10,$lgxs)}]
    set res K:[format %.2f [expr {pow($x,3)}]]
    append res " A:[format %.2f [expr {pow($x,2)}]]"
    12
    append res " B:[format %.2f [expr {pow($xs,2)}]]"
    append res " C:[format %.2f [expr {pow(10,-$lgxs)*10}]]"
    append res " C:[format %.2f $xs]"
    append res " D:[format %.2f $x]"
    append res " L:[format %.2f $lgx]"
}

proc pow10 x {expr {pow(10,$x)}}
proc log10 x {expr {log10($x)}}
proc -log10 x {expr {-log10($x)}}
proc by100 x {expr {$x/100.}}
#-----
ui
bind . <Escape> {exec wish $argv0 &; exit}

```

