Overview of UWIN:

Product overview:

UWIN:

- UWIN is a migration toolkit from Unix to windows

  Both for application development & also for users.
- Provides the user a UNIX runtime environment on Windows NT/9x/2000.

  The UNIX shell interface is provided for the user (The Korn Shell with a $ prompt).
- Provides the developer a UNIX development environment on Windows NT/9x/2000

Product objective:

- To bridge the gap between UNIX and Windows by providing a user and a development environment.

  UWIN gives the power of 2 operating systems in a single desktop environment.
- Provide an environment for smooth migration of UNIX applications onto Windows.

  The source code of an application, developed for Unix can be rebuilt on UWIN by just compiling it once. This reduces all the porting time needed for the migration of the application from Unix to windows.

Need for UWIN:

- Increasing popularity of Windows as a desktop and server platform.
- Protects investment made in existing UNIX based applications and solutions.

  All the application developed using Unix Os can be easily ported onto windows Os environment by just recompiling it under UWIN, thereby avoiding the large amount of rework required for migration.
- UNIX users and developers using Windows box.

  Unix loyalists who were forced to use windows as there desktops will get a Unix look and feel & help for a smooth migration from Unix to windows.

Who needs UWIN:

- ISP, Internet Developers – Who need tools and scripting support.

Used for website hosting, online registration etc can be done using UWIN.

- Corporate for “write once deploy many” policy and gain “time to market advantage”.

  All the applications written for one OS can be deployed on to the other without going through any development cycle.
- ISV (independent software vendors) – Targeting UNIX applications to Windows segment
- Customers who want to capitalize on their UNIX assets/investment as they move to Windows
- UNIX users and developers working on Windows.

Footnote: What is the standard to which UWIN is complaint to?

Posix 1.003 standard

What are the portions developed by Wipro?

Pseudo terminals, vt100 support, services UMS (UWIN master service), UCS (UWIN client services), design for device implementation, etc.

Features & Benefits of UWIN:

UWIN for users:

- An UWIN user gets a complete Unix enviro nment on his desktop over the windows platform. Around 300 Unix utilities are running through UWIN. All the utilities are taken from the same source, which runs Unix, compiled on UWIN, & that binary will be packed with the UWIN package. Because of this we get the same power for all the utilities which one gets on Unix.
- Choice of shells – ksh, bash, tcsh, csh.
- Telnet server for remote access to win NT.

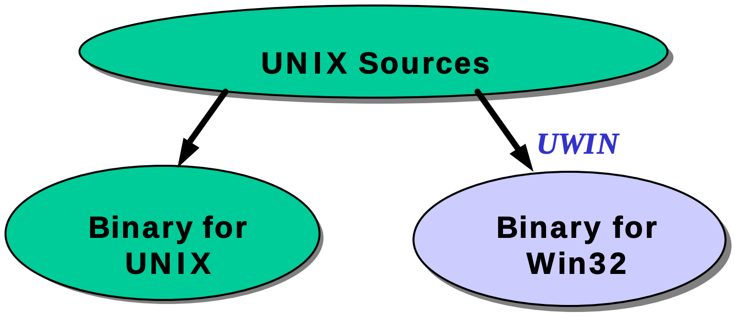
  Windows doesn’t have the feature of remotely accessing the other computer. UWIN will provide this feature of (both telnet server & client) Unix on to windows.
- Easy access to Windows registry as /reg.

  Windows registry is mounted on UWIN. This can be accessed through cd /reg from the shell.
- UNIX scripting support on Windows
- Thin client computing using X11

  This feature gives a GUI over Unix. Through this feature one an export the whole desktop on to a different system.
- Offers a dual environment on a single PC (without dual-boot)
- Windows ease of use + UNIX powerful shell tools and scripting
- Enhances productivity and quality of work

UWIN for developers:

- Develop and execute UNIX applications.
- Maintain single source base for both platforms.



- Develop hybrid applications using Win32 and UNIX APIs.
  Windows utilities can be merged with Unix applications. From a Unix source one can make a call to win32 API's as UWIN runs on an user mode above the win32 subsystem.
- Generate Windows DLLs from UNIX sources.
- cc or gcc compilers with gdb debugger.
  Cc is a wrapper over Microsoft cl compiler. For debugging this MSVC can be used.
- Debug UNIX applications
  –cc compiler (wrapper over cl) and Visual Studio IDE
  –gcc compiler and gdb debugger
- Utilize powerful scripting features in ksh93.

Native Interoperability:

Interaction of UWIN with windows, the OS over which UWIN is running.

- Same quality tools and utilities as on UNIX.
- Same GUI with X11 client.
- Enhances productivity and quality of work.

Footnote:

Using X11 client the user can get the GUI of the remote terminal.

In UWIN x11, server is not ported but it is available freely & xterm, which is the client side of X11, is used in UWIN. If an X11 server is running on a Unix server then the GUI of that can be exported on the client side using UWIN.

Cost Effectiveness:

- No change in source code.

The application must compile to Posix standards in order to have a common source base.

- Need not go through the development cycle again.
- Better benefits over competitors.

Difference from Unix systems:

- UNIX type of interface to Windows resources

  UWIN works on windows Os, only those, which can be done within the OS limits, can be done on UWIN.
- Access to Windows registry
- Mount domain/workgroup shares to local file system or access them through UNC (universal naming code)
- Run Windows console based applications.

Superiority over competitors:

- Better performance.
- Highest number of X/Open APIs supported.
- Better integration with underlying operating system.

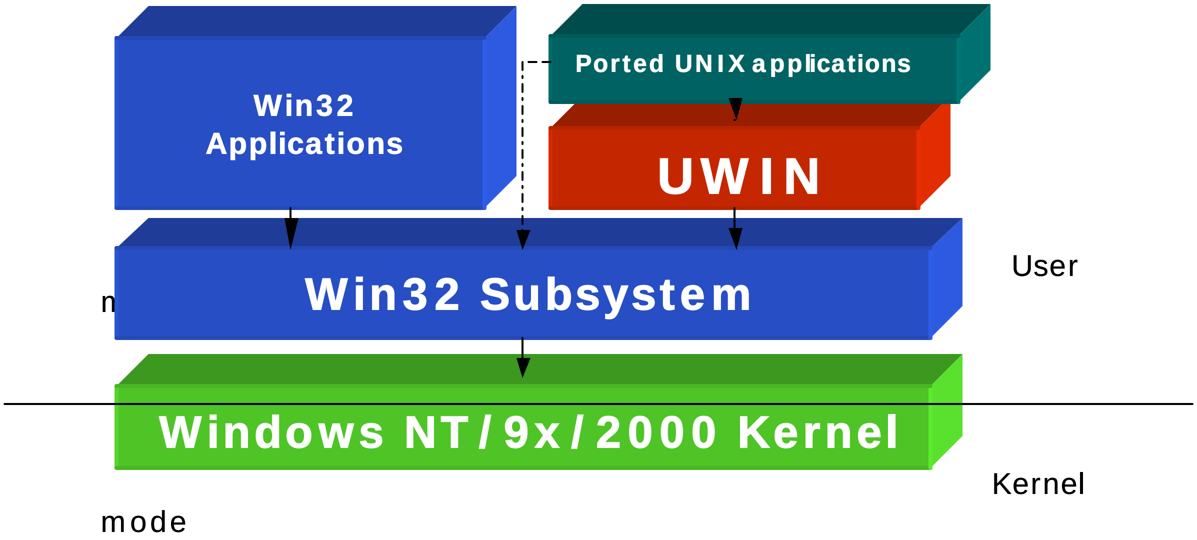
  This is achieved by having UNC, mounted registry etc.

List of ported utilities:

- Apache 1.3.6, Perl 5.05, Lynx 2.8
- CVS 2.0, Tcl/Tic/blt, VIM 5.3
- GNU inet utilities (inetd, telnetd, ftpd, etc)
- Flex, Rpcgen, Groff/Troff, atp 1.2b
- Patch 2.1, Gawk 3.02
- Tcsh, zlib library, pgSQL
- Redhat package manager (rpm).
- X11 R6.4 client libraries, Xterm.

Technical aspects:

UWIN architecture:



The architecture of UWIN, as shown in the above figure, depicts the way the control flows from UWIN on to the Os kernel. All the ported Unix applications, which sit over UWIN, can directly interact with the win32 subsystem of Win NT for win32 api calls. They can also talk to the win32 subsystem through UWIN, during the course of execution for the applications for which Unix environment is required. Because of this feature UWIN applications can enjoy both windows & Unix kind of environments. Finally all the applications executed by UWIN have to be forwarded to the kernel through Win32 subsystem.

Packages:

- Base Toolkit
  –UNIX environment on Windows NT/9x/2000
  –Includes over 300 UNIX shell tools and utilities
  –UNIX inet daemons/clients

This package is for a user who wants to get the Unix environment on his windows OS. This package doesn't have the compiler required for application development. The user can execute all the utilities included in the package but cannot built or compile his own application.

- SDK (Software Development Kit)
  –Adds applications development support
  –Over 60 UNIX shared libraries

–Cc compiler – a wrapper over native Microsoft compiler (cl)

- Runtime
  –Set of libraries providing UNIX API support on Windows
  –Customized for individual requirements
- Add-on
  –Set of ported utilities
- Integrated package
  - This is a combination of base toolkit & SDK, which is available as a complete package of around 12 Mb & available with a license agreement.

Composition:

This gives a brief description of UWIN after it is installed on a target machine.

- System components
  –posix.dll, ast52.dll, ast54.dll, UWIN.cpl
  These are the libraries required for UWIN to work. They are stored in the Win NT system32 directory.
  Posix.dll exports all the Unix system calls.
  Ast54.dll gives the C libraries.
  UWIN.cpl corresponds to the control panel applet provided for UWIN.
- File system
  –Log files (/tmp: UWIN_log, ums.out, ucs.out)
  Problems associated with posix.dll are logged on in UWIN_log.
  Problem associated with UWIN master service is logged on in ums.out.
  Problems associated with UWIN client service are logged on in ucs.out
  –Shell extensions (/usr/fun – vi_keybind, pushd, popd,dirs, reset)
  vi_keybind for mapping the arrow keys on to UWIN.
  Reset for resetting the VT100 terminal.
  –Documentation (/usr/doc: license.txt, UWIN.html, /usr/man)
  –Auto mounted directories (/, /sys,/dev, /msdev, /proc, /reg, /win)
  / Corresponds to the root directory where UWIN is installed.
  /sys maps to the system directory of the OS.
  /proc a virtual directory which lists all the processes running under UWIN.
  /dev gives the information about the devices.

/win maps to windows directory.

–GNU tools and other ported utilities (/usr/local)

- Registry keys

  –HKLM\SOFTWARE\AT&T Labs\UWIN\<version-number>

  –Console, InetConfig, IPC, Resources, Telnetd, UMS.

Footnote:

- All the drives on the computer are mounted on the root directory /. For going to any drive like D or C we can specify, cd /d to go to that drive.
- All the device related information is available in /dev directory of UWIN.
- All the binaries are located in /bin directory on UWIN.
- UWIN takes all the user & group information from the Win NT database.
- All the peripherals like floppy drive, tape drive, CD drive etc are all accessible from UWIN also.

All the features supported by the underlining OS, Windows, are also supported by UWIN.

Note on Different kinds of user:

- Unix involves only user name and passwords.
- Windows, along with user name & passwords also involves domains & trusted domains. While logging in we can specify whether it's a local account, domain account or a trusted domain account.

Trusted domains:

- Sharing of resources between two domains can be done if both the domains are trusted domains. All the resources accessible by one domain are accessed by the other trusted domain through this feature of Win NT. All the domains, listed as trusted, with not ask for the login name and password while accessing.

Technical features:

- Process control and management.

  Controlling the process involves killing a process, running a process in background, stopping a process etc.
- File descriptor semantics.

  Everything is a file descriptor in Unix, unlike in windows where it is all objects (following OOPS concepts) by which all the controls are through Handles. In UWIN the handles of windows are internally mapped on to these file descriptors.

- Unix signal semantics.

  Windows doesn’t have the concept of signals; it cannot send a signal to a process. But Unix works a lot with signals so here also mapping is done through UWIN.
- Terminal interface.

  Console terminal interface for ksh shell for invoking UWIN. A telnet session involves a pseudo terminal interface. For modem, dumb terminals, UWIN provides different terminal interfaces.
- UCB sockets on Winsock
- Pathname mapping from Unix to Windows
- Devices - character and block.

  UWIN supports both character & block devices. Console, pty, tty are the character devices supported by UWIN. Tape drives, floppies, form the block devices. All these devices are available in /dev.
- Unix permissions to WinNT permissions
- IPC (Inter Process Communication)

  UWIN supports message queue, shared memory & semaphores.
- WinNT Domain support
- Dynamic link libraries
- Error mapping.

  Error mapping between Unix & windows API is done by mapping the error numbers. In posix.dll there is a file, errno.h that takes care of mapping the error numbers.
- inet daemons/clients
- Registry and process file system
- Windows clipboard and mouse based cut & paste
- Mixed mode programming

Utilities in UWIN:

- Scripting

  –awk, sed, perl
- Remote access

  –telnet, rlogin, rsh, ftp, and others
- Task scheduling

  –at, cron
- Version maintenance

  –cvs
- Shells

  –ksh93, bash, csh, tcsh
- Editors

  –vi, ed
- Email servers/clients

–Sendmail, mailx

UWIN does not have a mail server, it is nor ported.

- Web server
  –Apache
- Shell/File utilities
  –grep, cp, mv, diff, find, ln, file, ps, pax, nmake, gunzip, cut, chmod, chown, chgrp, passwd, tar, cpio, kill, expr, mount
- X11
  –X11 client libraries, xterm
- Control panel applet
  –General, System, Master, Inetd, Client
- Compilers/Debuggers
  –Microsoft Visual C++, Borland C/C++, GNU gcc

Services in UWIN:

Services in UWIN are mainly meant for setuid support. They will be running as Win NT services.

- UMS (UWIN Master Serv ice)
  –Auto startup on reboot
  –setuid() support
  –Generates /etc/passwd & /etc/group files
  –Starts /etc/inetd.exe by running /etc/rc script
  –/etc/mkpasswd for re-generation of password/group files
- UCS (UWIN Client Service)
  –Manual start from UMS
  –setuid() support
  –Installed in three ways:
    - telnet to the account
    - Manual installation
    - Control panel applet

UWIN on Win NT & Win9X:

- Security issues
  –No concept of security on Win9x
- User database
  –No user database on Win9x
- Inet daemons
  –No Inet daemons on Win9x
- Services
  –No UMS and UCS on Win9x

What cannot be done with UWIN:

- Compilation generates Windows binaries.
  –Cannot take it on UNIX systems or vice-versa

Binaries generated from UWIN cannot be taken on to Unix, as the binaries are windows equivalent & therefore cannot be used on Unix. UWIN gives only source level compatibility & not binary level.

- Viewing GUI applications over telnet
  –Except console based applications
- Telnet into Win9x box
  –No inet daemons
- /etc/passwd & /etc/group files (only with Win 9X)
  –No auto generation

Support viewpoint:

What to lookout first:

- Information about UWIN
  –uname: system information
  Gives the version of UWIN, the system details, the OS ets
  –What: version of posix.dll
  As majority of changes takes place in Posix.dll the version & make help in giving the missed out patches for the user.
- Problem with UNIX APIs of UWIN
  –/tmp/UWIN_log
  If the problem is with the system calls that UWIN is providing, then it will be logged in UWIN_log file.
- Problem with UWIN services
  –ums.out and ucs.out
- Problem with UWIN installation
  –/tmp/install_log

Trouble shooting tools/utilities:

- /bin/trace
  –To trace UWIN applications
  –Provides information on all UNIX APIs called by it.
- /etc/traceit
  –To trace daemons
  –Provides information on all UNIX APIs called by it
- Debuggers
  –Microsoft Visual Studio IDE (when compiled with cl)
  –GNU gdb (when compiled with gcc)

Competitors:

- NutCRACKER
  –Similar implementation
  –Has proprietary APIs
  –Costs more
  –COM/DCOM support

  - –http://www.datafocus.com
- Interix
  - –Tie-up with Microsoft, privately held
  - –Part of NT kernel, works only on Windows NT
  - –Bought over by Microsoft
  - –http://www.microsoft.com
- Cygwin
  - –Public domain under GNU public license
  - –Bought over by Red hat
  - –http://sources.redhat.com/cygwin

UWIN On The Internet:

- Wipro website
  - –http://www.wipro.com/UWIN
- AT&T Research Labs website
  - –http://www.research.att.com/sw/tools/UWIN
- GTLink website
  - –http://www.gtlinc.com/Products/UWIN/UWIN.html