C2110 *UNIX and programming*

6th lesson

Linux and what did not fit into the introduction

Petr Kulhánek

kulhanek@chemi.muni.cz

National Centre for Biomolecular Research, Faculty of Science,
Masaryk University, Kamenice 5, CZ-62500 Brno

Contents

- **Processes**
  - **background and foreground processes, signals**
- **File System**
  - **quotas, removable media (USB drives), links,** sshfs
- **Remote File Transfer**
  - **scp, WinSCP,** wget

Processes

foreground/background processes

Running applications on the foreground

```
$ gimp
```

the foreground processes **block terminal**, because they use its standard input and output

Running of applications in background

```
$ gimp &
```

the background processes **do not block terminal**

at the end (behind the arguments and redirection) of command, enter ampersand.

Terminal (useful keyboard shortcuts):

Ctrl + Z pauses the process, the next process fate can be checked using commands:

jobs	lists processes that shell manages
bg	moves the process to the background
fg	moves the process to the foreground
disown	disconnects a process from the terminal (the process is not terminated when the shell is terminated)

Examples, Exercise I

```
$ kwrite           # starts the kwrite in the foreground
^Z                 # suspends the application (Ctrl + Z)
[1]+  Stopped                        kwrite
$ jobs # lists applications that are in the background or suspended
[1]+  Stopped                        kwrite
$ bg %1            # runs suspended application 1 in the background
[1]+ kwrite &
$ jobs
[1]+  Running kwrite &
$ fg %1            # Application 1 running in the background
                   # moves to the foreground
kwrite             # terminal is blocked, must be used (Ctrl + Z)
```

Job specifications:

%n or %name (sometimes even the job number itself)

number or name of the tabled jobs (**jobs**)

Signals and processes

Terminal (useful shortcuts):

Ctrl+C sends the signal SIGINT (interrupt) to the running process, the process is forcibly terminated in most cases

Command kill:

```
$ kill [-signal] PID
```

the process number to which the signal is to be sent (can be found by **ps**, **top**, **pstree**)

signal specification: N (signal number) -name (signal name) -SIGNAME (SIG + signal name)

Useful signals:

TERM	**15**	termination request (the process may respond to the signal)
INT	**2**	interruption request (**Ctrl+C**) (the process may react to the signal)
KILL	**9**	termination (**the process can't ignore the signal, it is forcibly terminated**)
STOP		suspends process (**Ctrl + Z**) (process can not ignore the signal)
CONT		restores the suspended process (the process can not ignore the signal)

Examples, Exercise II

```
$ ps -u kulhanek
 PID TTY          TIME CMD
...
 5440 pts/8    00:00:00 bash
 5562 pts/8    00:00:00 kwrite
 5566 pts/8    00:00:00 ps

$ kill 5562    # terminates the process kwrite

$ kwrite       # starts the application kwrite on the foreground
[1]+  Stopped kwrite
```

other terminal

```
$ ps -u kulhanek
 PID TTY          TIME CMD
...
 8401 pts/1    00:00:00 kwrite

$ kill -STOP 8401    # suspends kwrite
```

Examples

```
$ time kwrite

real    0m3.188s
user    0m0.349s
sys     0m0.044s
```

actual process run time

consumed machine time of the process

consumed machine time in system calls

```
$ time sleep 10

real    0m10.002s
user    0m0.000s
sys     0m0.002s
```

```
$ time sleep 0.01

real    0m0.012s
user    0m0.000s
sys     0m0.002s
```

Running the process has certain overhead

List of commands

top continuously displays processes sorted by processor load (termination by q key)

ps lists the processes running on the terminal or the specified specifications (ps -u user_name)

pstree lists processes (tree structure)

type prints the path to the standard application/command

kill sends a signal to a process, can be used to terminate problematic programs

time prints process runtime

ssh runs command on a remote computer

jobs prints background processes

fg transfers a process to the foreground

bg transfer a process to the background

sleep starts a process that waits for the specified time

for advanced

disown disconnects the process from the terminal

nohup runs process without interacting with the terminal (C2115)

wait waits for background processes to finish (C2115)

Exercise III

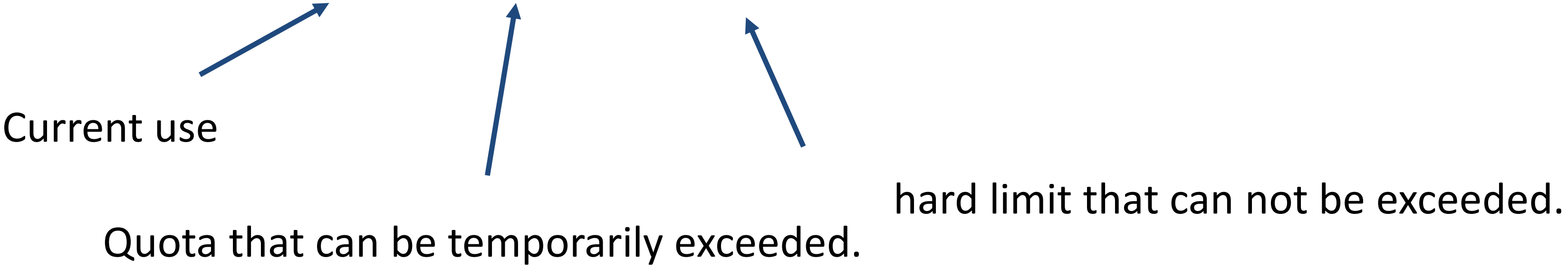
1. In the first terminal, start vmd program.
2. In the second terminal, display the process tree (pstree) including the PID, find the application process number, and try to terminate the TERM and KILL signals..
3. Repeat the exercise, but to suspend the application with STOP signal. Restore the application with CONT cont.
4. Run gimp at the foreground. Move it to the background without its termination

File system

Quotas

For your home directories, quotas are set to use disk space on **wolf.ncbr.muni.cz:/home/** partition. You can find the current usage and quota settings by using by command **quota**:

```
[kulhanek@wolf ~]$ quota -vs
Disk quotas for user kulhanek (uid 18773):
     Filesystem  blocks   quota   limit   grace   files   quota   limit
wolf.wolf.inet:/home/
                  1550M   1954M   2051M           20453       0       0
```



Current use

Quota that can be temporarily exceeded.

hard limit that can not be exceeded.

Exceeding of quotas may result in **nonfunctional login** to a graphical interface. In this case, login to text terminal (eg: Ctrl + Alt + F1) and move files to different disk partition (eg. a temporary directory /scratch/username or delete useless files).

Disk devices

An overview of the use of file systems, disk drives, and their attachment points provides the **df** command..

```
Filesystem                     Type       Size  Used Avail Use% Mounted on
/dev/mapper/server1-root       ext4        15G  8.4G  5.5G  61% /
none                           tmpfs      4.0K     0  4.0K   0% /sys/fs/cgroup
udev                           devtmpfs   3.9G  4.0K  3.9G   1% /dev
tmpfs                          tmpfs      793M  888K  792M   1% /run
none                           tmpfs      5.0M     0  5.0M   0% /run/lock
none                           tmpfs      3.9G  952K  3.9G   1% /run/shm
none                           tmpfs      100M   36K  100M   1% /run/user
/dev/mapper/server1-vbox       ext4        64G   52G  9.5G  85% /win
/dev/mapper/server1-scratch    ext4       598G  2.8G  565G   1% /scratch
wolf.wolf.inet:/software/ncbr nfs4        197G  156G   33G  83% /software/ncbr
wolf.wolf.inet:/home/          nfs4       493G  371G   98G  80% /home
```

device
file system type
mount point

File system type

ext3,ext4 third / fourth extended file system (native Linux file system)

nfs3, nfs4 network file system

vfat Virtual File Allocation Table (file system used in MS Windows)

ntfs New technology File System (developed by Microsoft for its operating systems)

!!! not case-sensitive !!! - Be careful when copying files of varying character sizes.

USB disks

USB drives are automatically connected to **/media/username** in a graphical environment.

```
[kulhanek@wolf01 ~]$ df -Th
Filesystem                     Type      Size  Used Avail Use% Mounted on
................................................................................
wolf.wolf.inet:/home           nfs4      280G  164G  102G  62% /home
/dev/sdg1                      vfat      962M  841M  122M  88% /media/kulhanek/B19A-1CA2
```

You can disconnect the disk in the graphical environment or by **umount** command. The command argument is mount point of the device.

```
[kulhanek@wolf01 ~]$ umount /media/kulhanek/B19A-1CA2
```

The disk can be disconnected only if it is not being used (no file can be opened, no process has to be set (under) the directory from the connection point including the connection point as the working directory). A list of processes using a given directory (mount point) can be obtained by the **lsof** (or fuser) command.

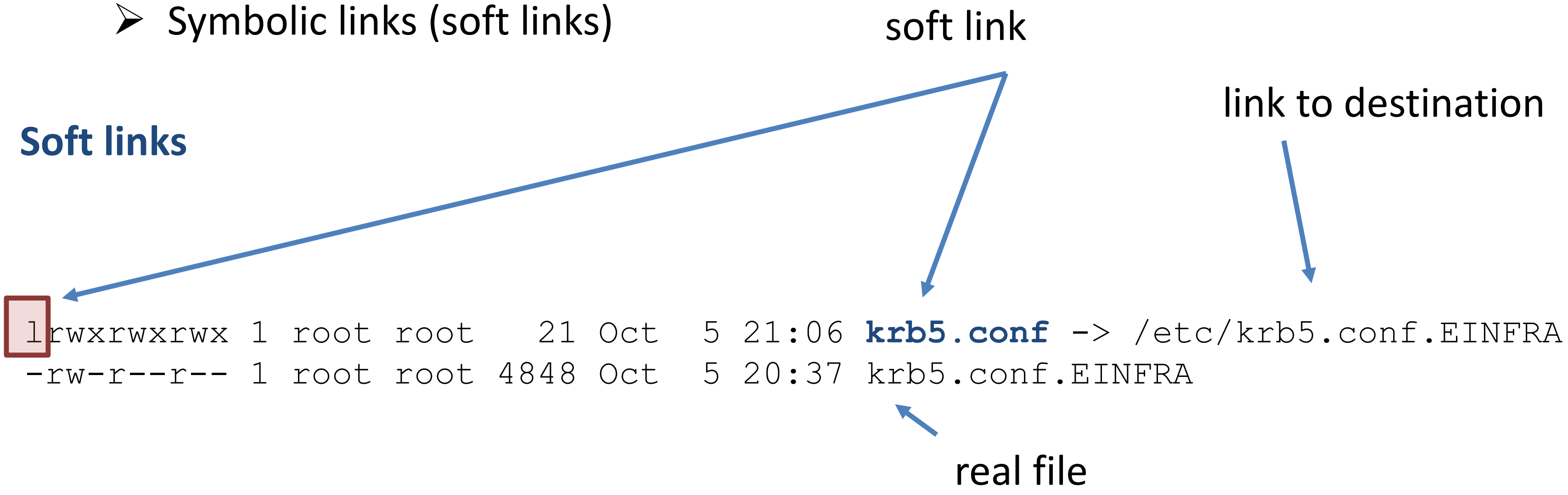
```
[kulhanek@wolf01 ~]$ lsof /media/kulhanek/B19A-1CA2/
COMMAND PID     USER    FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    31521 kulhanek  cwd   DIR    8,97     4096  518 /media/kulhanek/B19A-1CA2/GoslarFinal
bash    31893 kulhanek  cwd   DIR    8,97     4096  518 /media/ kulhanek/ B19A-1CA2/GoslarFinal
vi      32011 kulhanek  cwd   DIR    8,97     4096  518 /media/ kulhanek/ B19A-1CA2/GoslarFinal
```

Links

Links :

- Hard links
- Symbolic links (soft links)

Soft links



soft link

link to destination

```
lrwxrwxrwx 1 root root   21 Oct  5 21:06 krb5.conf -> /etc/krb5.conf.EINFRA
-rw-r--r-- 1 root root 4848 Oct  5 20:37 krb5.conf.EINFRA
```

real file

Properties of soft links:

- contain information about the path to the target object (file, directory, ...)
- from the point of view of the system behave as a target object
- access rights are derived from the target object
- destination object does not need to exist
- Is created by ln command with -s option, for example:

```
ln -s /etc/krb5.conf.EINFRA krb5.conf
```

List of commands

File system:

cd	**changes the current working directory**
pwd	**displays the path to the current working directory**
ls	**displays the contents of directory**

basic operations

mkdir	creates a directory
rmdir	deletes a directory (must be empty)
cp	copies files or directories
mv	moves files or directories
rm	removes files or directories
find	finds for files or directories

permissions

id	lists groups of given user
getent	lists users, user groups and other information
chmod	changes permissions of a file or directory
chown	changes the file's or directory’s owner
chgrp	changes the access group for file or directory
umask	default permissions for newly created files or directories

List of commands

File system (continue): **advanced functions**

quota	displays information about quota setting for mount points
du	prints the size of the directory or files
stat	lists detailed file or directory information
df	prints information about attached partitions
lsof	lists processes that have open files / directories on a mount point (directory)
sshfs	connects remote file system to a local directory tree using ssh protocol
mount	mounts the device to the local directory tree, lists connected devices (mount of MetaCenter data storages - C2115)
link	creates a link to a file or directory
unlink	remove a link to a file or directory

Remote file transfer

- scp
- WinSCP
- wget (home work)

Remote copy

The **scp** command is used for remote copying.

[] – can be omit

Syntax:

```
$ scp [-r] zdroj cil
```

Source and destination can be files or directories. When copying directories, you must use the **-r** (recursive) option.

Remote destination or host is identified by the machine name separated from the file name or directory by the colon.

```
[user@]hostname:[path/]file
```

Example of use:

```
$ scp pokus.txt wolf01.ncbr.muni.cz:/scratch/kulhanek

$ scp wolf01.ncbr.muni.cz:/scratch/kulhanek/pokus.txt .
```

Exercise IV

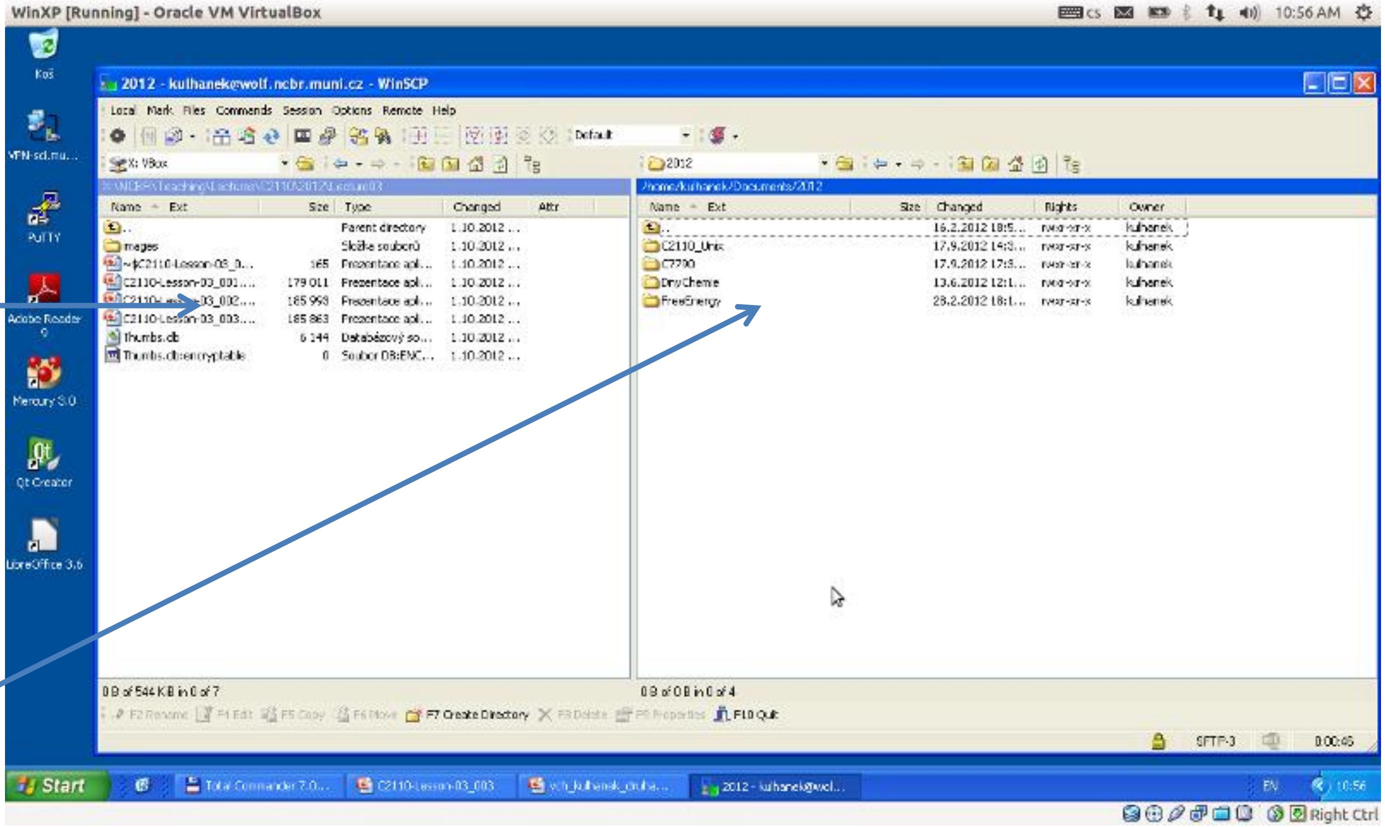
1. Make a copy of the directory ~/Documents to directory /scratch/username (make sure you have some file in directory Documents. e.g. presentation of this lecture).
2. Copy the content of the directory /scratch/username/Documents to the remote machine of your choice to the directory /scratch/username/wolfXX where wolfXX is the remote machine host. Use the scp command to copy.
3. Remove /scratch/username/Documents and /scratch/username/wolfXX

use two terminals

WinSCP

WinSCP http://winscp.net/eng/docs/lang:cs

Program for transfer files between MS Windows and computers supporting SFTP protocols or SCP (mostly Unix and Linux type).



local machine

remote machine

WinXP [Running] - Oracle VM VirtualBox

2012 - kulhanek@wolf.ncbr.muni.cz - WinSCP

X:\NCBR\Teaching\Lectures\C2110\2012\Lecture03

Name Ext	Size	Type	Changed	Attr
..		Parent directory	1.10.2012 ...	
mages		Složka souborů	1.10.2012 ...	
~$C2110-Lesson-03_0...	165	Prezentace apl...	1.10.2012 ...	
C2110-Lesson-03_001....	179 011	Prezentace apl...	1.10.2012 ...	
C2110-Lesson-03_002....	185 993	Prezentace apl...	1.10.2012 ...	
C2110-Lesson-03_003....	185 863	Prezentace apl...	1.10.2012 ...	
Thumbs.db	6 144	Databázový so...	1.10.2012 ...	
Thumbs.db:encryptable	0	Soubor DB:ENC...	1.10.2012 ...	

0 B of 544 KB in 0 of 7

/home/kulhanek/Documents/2012

Name Ext	Size	Changed	Rights	Owner
..		16.2.2012 18:5...	rwxr-xr-x	kulhanek
C2110_Unix		17.9.2012 14:3...	rwxr-xr-x	kulhanek
C7790		17.9.2012 17:3...	rwxr-xr-x	kulhanek
DnyChemie		13.6.2012 12:1...	rwxr-xr-x	kulhanek
FreeEnergy		28.2.2012 18:1...	rwxr-xr-x	kulhanek

0 B of 0 B in 0 of 4

Text files MS Win ⇔ Linux

Text files created under MS Windows and Linux **are not** fully **compatible** because each operating system uses different coding for **end of lines**.

Linux:	\n	(line feed 0x0A)
MS Windows:	\r+\n	(carriage return 0x0D, line feed 0x0A)

To convert files, you can use the programs **d2u** or **u2d** (on cluster WOLF)

1) Activation of cats module

```
$ module add cats
```

2) File conversion MS Windows => Linux

```
$ d2u soubor.com
```

3) File conversion Linux => MS Windows

```
$ u2d soubor.log
```

More information:
http://en.wikipedia.org/wiki/Newline

Exercise V

1. Copy file 1SS9.pdb from the directory /home/KULHANEK/Downloads/ to your home directory.
2. Start your virtual machine with MS Windows XP (/win/win7uc/start).
3. Run the application WinSCP.
4. Download 1SS9.pdb from your home directory to the virtual machine. Open file in Notepad (Notepad). Is content of the file displayed correctly?
5. Correct end-line coding in the file 1SS9.pdb using u2d (module cats) and reopen file in the virtual machine using Notepad. Is content of the file displayed correctly now?

Home works

Download file from web

You can use **wget** to download files from the web. Remote machines must provide files using **ftp** , **http,** or**https** .

```
$ wget [-O output_name] url
```

big O

url (uniform resource locator)
www file identifier

Example of use:

```
$ wget http://www.rcsb.org/pdb/files/1SS9.pdb
```

code of PDB structure

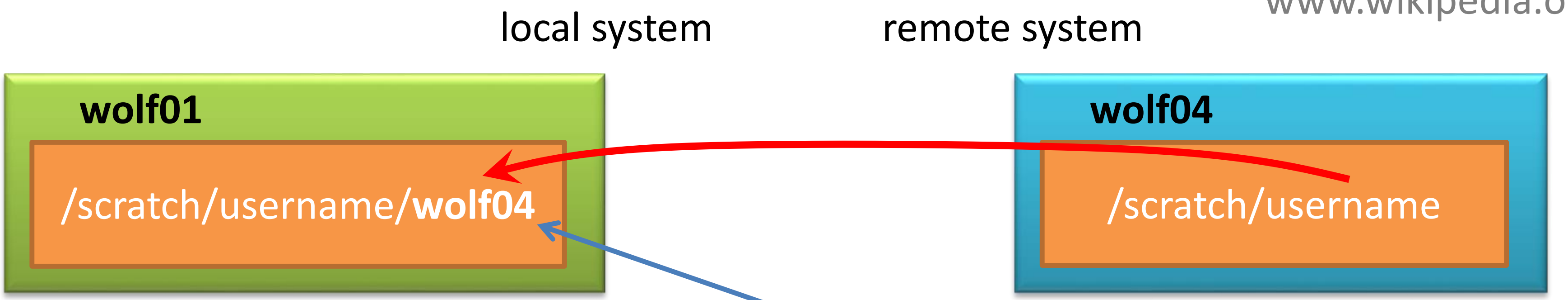
Exercise I

1. Download pdb structure 1SS9 to file called structure.pdb using wget command.
2. Open file structure.pdb in the program vmd.
3. Download to scratch directory using command wget the installation image (ISO file) of Ubuntu Server 14.04.5. Verify checksum of file via Md5sum command

sshfs

sshhf or ssh File System is used to connect **remote filesystem directory to local dictionary tree using** the encrypted SSH connection. On the server there is not necessary configure more than SSH. On the client, there is need to install the package sshfs

www.wikipedia.org



creation of mount point (should be empty directory)

connection:

```
$ mkdir /scratch/username/wolf04
$ sshfs wolf04:/scratch/username /scratch/username/wolf04
```

remote file system

local mount point

disconnection:

```
$ fusermount -u /scratch/username/wolf04
```

Exercise II

local system

1. Go to the directory /scratch/username
2. Create directory “remote”
3. Mount remote filesystem /scratch/username into the directory
4. Check the connection by df and mount commands
5. Go to the "remote” directory
6. Create a file test.txt in it

remote system

1. Go to the directory /scratch/username
2. Display content of dictionary
3. Remove file test.txt

local system

7. Display the contents of a directory, what is the size of file test.txt?
8. Disconnect the remote file system

as remote system, use neighboring work station
use two terminals

MS Windows as a client - overview

Login to Unix from MS Windows (text terminal) :

putty	(http://www.chiark.greenend.org.uk/~sgtatham/putty/)
ssh	(eg. environment Cygwin; http://www.cygwin.com/)

Data transfer between Unix and MS Windows:

WinSCP	(http://winscp.net)
scp	(eg. environment Cygwin; http://www.cygwin.com/)

Display export from Unix to MS Windows (X11 server):

Xming	(http://sourceforge.net/projects/xming/)
cygwin	(http://www.cygwin.com/)

Login from Unix to MS Windows (remote Desktop):

rdesktop