

Introducción a la programación en Perl para bioinformática

Semestre 2014-1
PDC Biomédicas y PDC Bioquímicas - UNAM

Introducción al biocómputo en sistemas UNIX/Linux

Pablo Vinuesa (vinuesa@ccg.unam.mx)
Centro de Ciencias Genómicas-UNAM, México
<http://www.ccg.unam.mx/~vinuesa/perl4bioinfo/>


Introducción al biocómputo en sistemas Unix/Linux

1. Qué es UNIX y Linux
2. Tengo una PC que corre windows, ¿cómo puedo correr Linux en mi máquina?
3. ¿Cómo hago trabajar a UNIX/Linux? - el shell: comandos y conceptos básicos
4. Uso de tuberías de comandos UNIX/Linux para procesar archivos de secuencias

Introducción al biocómputo en sistemas UNIX/Linux

· ¿Qué es UNIX? - fuente: Wikipedia

- Unix (registrado oficialmente como UNIX®) es un [sistema operativo portable, multitarea y multiusuario](#)
- su desarrollo inicia en 1969 por un grupo de empleados de los [laboratorios Bell](#) de [AT&T](#), entre los que figuran [Ken Thompson](#), [Dennis Ritchie](#) y [Douglas McIlroy](#)
- UNIX es un [Sistema Operativo no libre](#) muy popular, porque está basado en una arquitectura que ha demostrado ser técnicamente estable.



Ken Thompson y Dennis Ritchie
Fuente: Wikipedia

Introducción al biocómputo en sistemas UNIX/Linux

· La evolución de las familias UNIX - fuente: Wikipedia

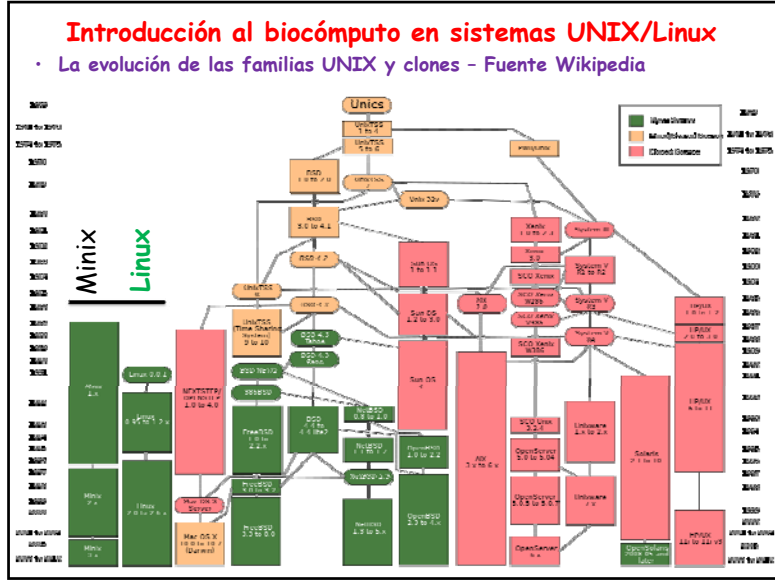
Existen varias familias del sistema operativo UNIX que han evolucionado de manera independiente a lo largo de los años. Cada familia se distingue no tanto por sus diferencias técnicas como por sus diferencias en propiedad intelectual. Se observa que todas las familias se han visto contaminadas, directa o indirectamente, por otras familias. Las familias UNIX más significativas son:

AT&T: la familia que tuvo su origen en el UNIX de AT&T. Considerada la familia UNIX "pura" y original. Sus sistemas operativos más significativos son UNIX System III y UNIX [System V](#).

BSD: familia originada por el licenciamiento de UNIX a Berkely. BSD se reescribió para no incorporar propiedad intelectual originaria de AT&T en la versión 4. La primera implementación de los protocolos [TCP/IP](#) que dieron origen a [Internet](#) son la pila (stack) TCP/IP BSD.

AIX: esta familia surge por el licenciamiento de UNIX System III a [IBM](#).




Xenix: familia derivada de la adquisición de los derechos originales de AT&T primero por parte de Microsoft y de esta los vendió a SCO.



Introducción al biocómputo en sistemas UNIX/Linux

- ¿Qué es Linux? - Evolución de sistemas UNIX y similares a UNIX (fuente: Wikipedia)

GNU: En 1983, Richard Stallman anunció el Proyecto GNU, un ambicioso esfuerzo para crear un sistema similar a Unix, que pudiese ser distribuido libremente. El software desarrollado por este proyecto -por ejemplo, GNU Emacs y GCC- también han sido parte fundamental de otros sistemas UNIX. (vean conferencias de R. Stallman en youtube ...)

POWERED BY
Linux

Richard Stallman Linus Torvalds

Linux: En 1991, cuando Linus Torvalds empezó a proponer el núcleo Linux y a reunir colaboradores, las herramientas GNU eran la elección perfecta. Al combinarse ambos elementos, conformaron la base del sistema operativo (basado en POSIX) que hoy se conoce como GNU/Linux. Las distribuciones basadas en el núcleo, el software GNU y otros agregados entre las que se pueden mencionar a Red Hat Linux y Debian GNU/Linux se han hecho populares tanto entre los aficionados a la computación como en el mundo empresarial y científico. Obsérvese que Linux tiene un origen independiente, por lo que se considera un 'clónico' de UNIX y no un UNIX en el sentido histórico.

Introducción al biocómputo en sistemas UNIX/Linux

- Tengo una PC que corre windows, ¿cómo puedo correr Linux en mi máquina?

- Puedes instalar Linux en una nueva partición (Lo más recomendable).
descargas gratuitas de distribuciones desde:
=> 1. **Ubuntu 13.04** - <http://www.ubuntu.com/getubuntu/download>
Ver instrucciones de instalación aquí:
<http://www.ubuntu.com/download/desktop/install-desktop-latest>
OpenSuse - <http://software.opensuse.org/>
Fedora - <http://fedoraproject.org/es/get-fedora>
Biolinux 7 - <http://nebc.nerc.ac.uk/tools/bio-linux/bio-linux-7-info>
- Puedes instalar cygwin sin necesidad de particionar el disco.
<http://www.cygwin.com/>
- Puedes descargar Knoppix "liveCD" que no requiere instalación en el disco duro
<http://www.knopper.net/knoppix/index-en.html>
<http://bioknoppix.hpcf.upr.edu/>

Introducción al biocómputo en sistemas UNIX/Linux

- Tengo instalado un sistema operativo UNIX o Linux en mi laptop, ¿Cómo accedo al servidor ivory para hacer los ejercicios?

Abre una terminal y teclea el siguiente comando:

```
$ ssh -Y usuario@132.248.32.18
```

¿Cómo obtento información o ayuda sobre las opciones de los comandos?

```
$ man ssh # otra opción es ssh --help
```

```
SSH(1) BSD General Commands Manual SSH(1)
```

NAME
ssh lu2014 OpenSSH SSH client (remote login program)

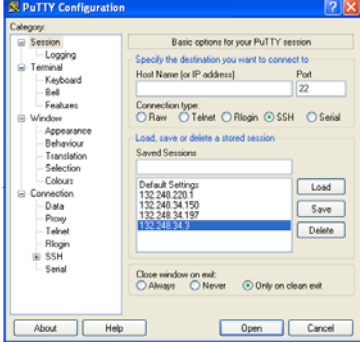
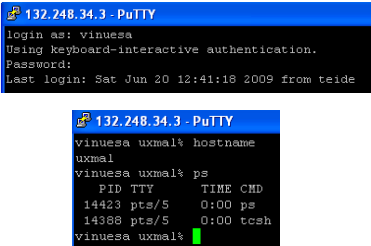
SYNOPSIS
ssh [-i246AaCfGkKMNqstTvXxYy] [-b bind_address] [-c cipher_spec] [-D [bind_address]:port] [-e escape_char] [-F config_file] [-I pkcs11] [-i identity_file] [-L [bind_address]:port:host:hostport] [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port] [-R [bind_address]:port:host:hostport] [-S ctl_path] [-W host:port] [-w local_tun[:remote_tun]] [user@]hostname [command]

DESCRIPTION
ssh (SSH client) is a program for logging into a remote machine and for executing commands on a remote machine. It is intended to replace rlogin and rsh, and provide secure encrypted communications ...

Introducción al biocómputo en sistemas UNIX/Linux

- Tengo una laptop que corre windows, ¿cómo puedo acceder a un servidor Linux o UNIX desde mi máquina?

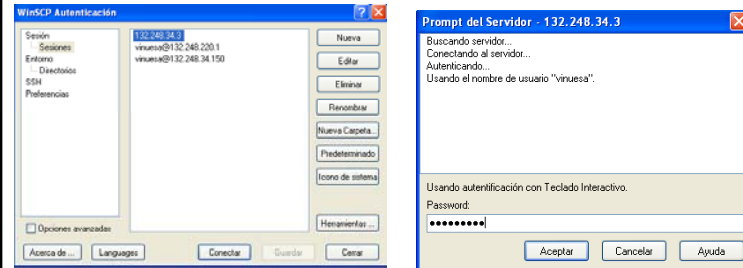
- Instala PuTTY <http://www.chiark.greenend.org.uk/~sgtatham/putty/>
- Lee <http://www.thegeekstuff.com/2009/07/10-practical-putty-tips-and-tricks-you-probably-didnt-know/>

- PuTTY, abre una consola en windows para establecer comunicación por ssh (Secure Shell) con un servidor UNIX

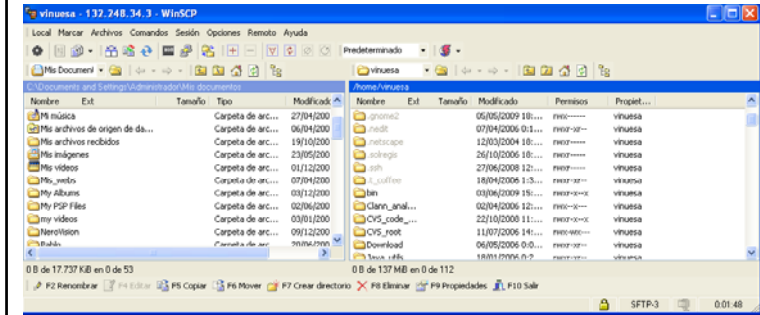
Introducción al biocómputo en sistemas UNIX/Linux

- Tengo una PC que corre windows, ¿cómo puedo mover archivos a/o de un servidor Linux a mi máquina?
- Winscp abre una consola en windows para intercambiar archivos con servidores UNIX/Linux a través de un ambiente gráfico y scp. <http://sourceforge.net/projects/winscp/>



Introducción al biocómputo en sistemas UNIX/Linux

- Tengo una PC que corre windows, ¿cómo puedo mover archivos a o de un servidor Linux a mi máquina?
- Winscp abre una consola en windows para intercambiar archivos con servidores UNIX/Linux a través de un ambiente gráfico mediante scp



Ya puedes transferir archivos entre máquinas y directorios (carpetas) usando el protocolo **secure copy (scp)**.

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos, unos ejemplos
- ¿Qué máquina es ésta a la que estoy conectado?
- hostname** - read or set the hostname or the NIS domainname

```
vinuesa@ivory:/$ hostname
ivory
vinuesa@ivory:/$ hostname -i # corre también hostname --help
132.248.32.18
vinuesa@ivory:/$
```

- uname** - Print certain system information

```
vinuesa@ivory:~$ uname
Linux
vinuesa@ivory:~$ uname -a
Linux ivory 3.2.0-41-generic #66-Ubuntu SMP Thu Apr 25 03:27:11 UTC 2013
x86_64 x86_64 x86_64 GNU/Linux
```

```
vinuesa@ivory:~$ free -g
              total        used         free      shared    buffers     cached
Mem:           251          247            4            0             0
```

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos, unos ejemplos
- ¿Quién soy, quien más está conectado y qué están haciendo?
- w** - print summaries of system usage, currently logged-in users and what they're doing

```
vinuesa@ivory:~$ w
17:35:39 up 90 days,  3:20,  7 users,  load average: 0.00, 0.01, 0.05
USER      TTY      FROM          LOGIN@   IDLE   JCPU   PCPU   WHAT
lsoto    pts/0    189.138.233.40  15:46   56:19  0.58s  0.07s  ssh -X lsoto@10.0.6.200
ricardo  pts/2    132.248.32.212  Mon10   7:41m  1.05s  0.02s  more genes_sin_cog.faa
ricardo  pts/3    132.248.32.212  10:02   2:28   1.75s  1.51s  -bash
vinuesa  pts/4    132.248.34.197  16:26   1:08m  0.52s  0.52s  -bash
poliver  pts/5    132.248.32.62   18Jun13 47days 6:08   1.19s  -bash
vinuesa  pts/6    132.248.220.197 16:28   0.00s  0.96s  0.01s  w
vinuesa  pts/7    132.248.220.197 17:12   20:45  0.55s  0.55s  -bash
```

- who** - display information about the current status of the system

```
vinuesa@ivory:~$ who
lsoto    pts/0    2013-08-07 15:46 (189.138.***)
ricardo  pts/2    2013-08-05 10:03 (132.248.***)
ricardo  pts/3    2013-08-07 10:02 (132.248.***)
vinuesa  pts/4    2013-08-07 16:26 (132.248.***)
poliver  pts/5    2013-06-18 08:29 (132.248.***)
vinuesa  pts/6    2013-08-07 16:28 (132.248.***)
vinuesa  pts/7    2013-08-07 17:12 (132.248.***)
```

Introducción al biocómputo en sistemas UNIX/Linux

Introducción al shell y la línea de comandos

Introducción al biocómputo en sistemas UNIX/Linux

· Explorando el nuevo ambiente ... El shell y la interpretación de comandos

1. ¿Cómo hago trabajar a UNIX/Linux? - el shell e interpretación de comandos

- El **shell** es un programa que toma comandos desde la consola y los pasa al sistema operativo para su ejecución. Casi todas las distros de Linux proveen un programa de shell del proyecto GNU llamado **bash**. Bash es un acrónimo para *Bourne Again Shell*, una versión moderna y mejorada del shell original de UNIX escrito por Steve Bourne.
- Una de las características más sobresalientes de estos sistemas operativos es la amplia gama de comandos (herramientas) con las que cuentan. **Más de 200 comandos** se distribuyen con el sistema operativo estándar. Cada una está especializada en hacer unas pocas cosas, pero muy bien, generalmente con múltiples opciones.
- El verdadero poder de estos sistemas operativos, a diferencia de Windows, radica en la elegancia y sencillez con la que **estos comandos pueden ser combinados** para realizar funciones mucho más sofisticadas que las que hace un comando individual. Es decir, **el shell es programable**

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos
- ¿Dónde encuentro una lista y descripción básica de los comandos disponibles ?

1. Mira estas entradas en Wikipedia:

http://en.wikibooks.org/wiki/Linux_Guide/Linux_commands

http://en.wikipedia.org/wiki/List_of_Unix_programs

2. Y corre estos comandos para ver parte de los programas del sistema instalados en el servidor o en tu máquina:

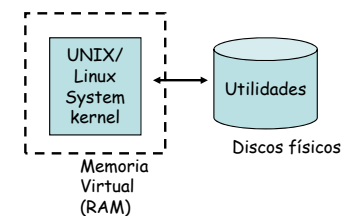
```
ls /bin
```

```
ls /usr/bin
```

Introducción al biocómputo en sistemas UNIX/Linux

· explorando el nuevo ambiente ... El shell y la interpretación de comandos

- ¿Qué es el shell?



Los sistemas UNIX/Linux se dividen lógicamente en dos piezas: el **kernel** y las **utilidades**.

El kernel es el corazón del sistema y reside en la memoria de la computadora desde el momento que se arranca y hasta que se apaga.

Las utilidades (comandos), residen en el disco físico y se cargan en memoria sólo cuando son llamadas.

El shell también es un programa. Se carga automáticamente en memoria desde que uno hace el login a una máquina para que el usuario pueda interactuar con ella.

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos

- Las responsabilidades del shell -

```

    graph TD
      Shell[SHELL  
bash, csh,  
ksh, sh, tcsh]
      Shell --- LPI[lenguaje de programación interpretado]
      Shell --- EP[ejecución de programas]
      Shell --- SNVA[sustitución de nombres de variables y archivos]
      Shell --- CA[control del ambiente]
      Shell --- CC[conectividad entre comandos (pipes)]
      Shell --- RI[redireccionamiento de I/O]
  
```

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos

- Ejecución de programas por el shell - ejemplos

- formato básico de un comando

```

    comando [argumento1 arg2 arg3 ...]
    -bash-3.1$ cp file1 tmp1/
  
```

```

    graph LR
      C[comando] --- A[argumentos]
      A --- F[file1]
      A --- T[tmp1/]
  
```

```

    -bash-3.1$ echo los espacios separan argumentos, necesito al menos 1espacio
    los espacios separan argumentos, necesito al menos 1espacio
    -bash-3.1$
  
```

- Además de **argumentos**, los comandos pueden tener **opciones**, las cuales preceden a los argumentos y llevan un guión sencillo delante de una o más opciones.

```

    -bash-3.1$ ls -Flsh tmp1/ # igual que ls -F -l -s -h tmp1/
    total 19K
    1.0K -rw-r--r-- 1 vinuesa cifn-ux 10 Jun 21 00:13 archivo1.txt
    1.0K -rw-r--r-- 1 vinuesa cifn-ux 10 Jun 21 00:13 archivo2.txt
    1.0K -rw-r--r-- 1 vinuesa cifn-ux 20 Jun 21 00:14 archivos2-1_concatenados.txt
    15K -rw-r--r-- 1 vinuesa cifn-ux 15K Jun 21 00:44 lo_nuevo.html
    1.0K -rw-r--r-- 1 vinuesa cifn-ux 130 Jun 21 2009 ls.out
    -bash-3.1$
  
```

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... El shell y la interpretación de comandos

- Tecleando comandos para el shell - el ciclo de comandos

```

    graph LR
      S1[shell] --> D1[$]
      D1 --> D2[$ls]
      D2 --> S2[shell]
      S2 --> C1[ls]
      C1 --> S3[shell]
      S3 --> D3[$ls File1.txt File2.pl]
      D3 --> S4[shell]
  
```

El shell encuentra y ejecuta los programas (binarios o ejecutables) que se encuentran en los directorios guardados en la **variable de ambiente \$PATH**

Ejecuta en la terminal el siguiente comando:

```

    (echo imprime el valor de una variable y/o una lista de argumentos)
    -bash-3.1$
    vinuesa@ivory:/$ echo $PATH
    /usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
    home/meme/meme_4.8.1/bin
    vinuesa@ivory:/$
  
```

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente - las variables de ambiente

```

    vinuesa@ivory:/$ printenv #imprime el valor de vars. de ambiente de una sesión
    TERM=xterm
    SHELL=/bin/bash
    XDG_SESSION_COOKIE=a86cbb420b73aba569dc2fc0000543-1375910884.979616-1232558530
    SSH_CLIENT=132.248.220.197 52275 22
    SSH_TTY=/dev/pts/6
    USER=vinuesa
    LS_COLORS=rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=01;35:do=01;35:bd=40;33;01:
    MAIL=/var/mail/vinuesa
    PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/
    /home/meme/meme_4.8.1/bin
    PWD=/
    LANG=en_US.UTF-8
    SHLVL=1
    HOME=/home/vinuesa
    LOGNAME=vinuesa
    SSH_CONNECTION=132.248.220.197 52275 132.248.32.18 22
    LESSOPEN=| /usr/bin/lesspipe %s
    DISPLAY=localhost:15.0
    LESSCLOSE=/usr/bin/lesspipe %s %s
    _=/usr/bin/printenv
    OLDPWD=/bin
    vinuesa@ivory:/$
  
```

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente - las variables de ambiente y su configuración

Podemos configurar el ambiente editando o añadiendo variables de ambiente al archivo de configuración `~/.bashrc` ó `~/.bash_profile`

En Ubuntu tenemos un ambiente pre-configurado gracias a los archivos que se encuentran en el directorio `/etc/skel`

Haz un `ls -a /etc/skel`

Y veamos el contenido de uno de ellos ejecutando el comando

`less /etc/skel/.bashrc` # (`less` es un paginador)

Veremos más adelante ejemplos de configuración del ambiente mediante edición de archivos `~/.bashrc` ó `~/.bash_profile`. El símbolo `~` equivale a `/home/usuarioX`, es un alias para `$HOME`

Los archivos de configuración como `.bashrc` tienen un `.` delante del nombre. Esto los convierte en archivos "ocultos", que no son visibles mediante un simple `ls`.

Requerimos `ls -a` para visualizarlos. Ejecuta los siguientes comandos para verificarlo:

```
ls ~
ls -a ~
```

Introducción al biocómputo en sistemas UNIX/Linux

Vean también: http://en.wikipedia.org/wiki/List_of_Unix_programs

- Comandos y conceptos básicos - una selección de comandos

1.- moviéndonos por el sistema y trabajando con archivos

- `ls` - lista información sobre archivos y directorio
- `cat` - despliega contenidos de un archivo o los concatena
- `less` - un paginador que despliega el contenido de un archivo página a página
- `wc` - cuenta líneas, palabras y caracteres
- `cp` - copia archivos
- `mv` - renombra o mueve archivos
- `rm` - elimina un archivo o directorio
- `chmod` - cambia permisos de archivos y directorios
- `tar` - crea un "jarro" de archivos y/o directorios
- `zip` - comprime archivos
- `head` - despliega la cabecera del archivo
- `tail` - despliega la cola del archivo
- `file` - muestra la clasificación de un archivo

2.- trabajando con directorios

- `pwd` - print working directory
- `mkdir` - crea un directorio
- `cd` - cambia de directorio
- `rmdir` - elimina directorio (sólo si están vacíos)
- `find` - busca archivos y directorios en base a características definidas por el usuario

Introducción al biocómputo en sistemas UNIX/Linux

Vean también: http://en.wikipedia.org/wiki/List_of_Unix_programs

- Comandos y conceptos básicos

2.- trabajando con directorios (cont.)

- `pwd` - print working directory
- `mkdir` - crea un directorio
- `cd` - cambia de directorio
- `rmdir` - elimina directorio (sólo si están vacíos; usa `rm -fr dir/` para eliminar directorios)
- `df` - despliega información de uso de disco
- `du` - despliega información de uso de disco por archivo

3.- trabajando con texto

- `grep` - busca patrones en archivos
- `cut` - selecciona caracteres o campos de archivos
- `sort` - ordena y/o conjunta archivos
- `uniq` - muestra líneas únicas
- `tr` - reemplaza caracteres indicados
- `sed` - edición no interactiva de archivos
- `awk` - filtrado de archivos por campos

4.- trabajando con procesos y comandos

- `top` - despliegue dinámico de estatus de procesos
- `ps` - despliegue de estatus de procesos
- `kill` - mata procesos por PID
- `nice` - cambia la prioridad de un comando
- `which` - muestra dónde se ubica un comando en el PATH
- `history` - muestra historial de comandos

Introducción al biocómputo en sistemas UNIX/Linux

Vean también: http://en.wikipedia.org/wiki/List_of_Unix_programs

- Comandos y conceptos básicos

5.- trabajando en la red con directorios y archivos remotos

- `ssh` - ejecuta comandos de manera segura en un sistema remoto
- `scp` - copia de manera segura uno o más archivos desde o hacia un sistema remoto
- `sftp` - copia de manera segura archivos desde un sistema remoto hacia una máquina local
- `wget` - descarga archivos desde una URL

6.- comandos para compilación de programas

- `configure` - configura código fuente de manera automática
- `gcc` - compila programas escritos en C y C++
- `make` - utilidad para construir binarios y librerías a partir de código fuente mediante la lectura de instrucciones contenidas en archivos llamados `makefiles` que especifican cómo derivar el programa diana.

Introducción al biocómputo en sistemas UNIX/Linux

- explorando el nuevo ambiente ... Trabajando eficientemente con el shell

¿Cómo me muevo en la línea de comandos?

- Usa **ctrl-e** para ir al final de la línea
- Usa **ctrl-a** para ir al principio de la línea

¿Cómo edito la línea de comandos?

- Usa la techa **backspace** para eliminar uno a uno caracteres (del final hacia el principio)
- Usa **ctrl-w** para eliminar una palabra completa (del final hacia el principio)
- Usa **ctrl-u** para eliminar la línea completa (del final hacia el principio)

¿Cómo aborto o suspendo la ejecución de un comando?

- Usa **ctrl-c** para abortar la ejecución del último comando
- Usa **ctrl-z** para suspender la ejecución del último comando

Repetición de la ejecución de un comando y completado de nombres de comandos/archivos

- Unix recuerda los comandos ejecutados: usa flecha arriba o abajo para moverte por el historial de comandos "history" file o escribe **history** | **grep comando**
- Usa **TAB** para completar automáticamente el nombre de comandos, archivos o directorios

NAVEGACIÓN DEL SISTEMA

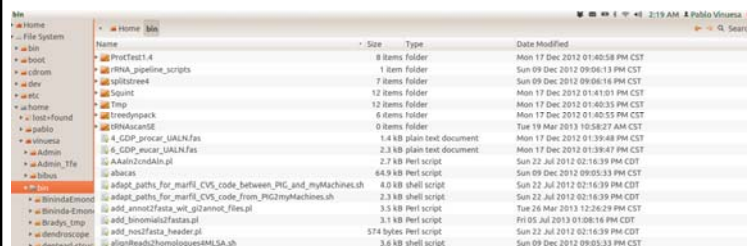
Comandos básicos:

- pwd** — Print name of current working directory.
- cd** — Change directory.
- ls** — List directory contents.

El sistema de archivos

Igual que en Windows, los sistemas operativos tipo UNIX como Linux organiza sus archivos en una estructura jerárquica de directorios. El primer directorio del sistema Se conoce como el **directorio raíz /**, que contiene archivos y subdirectorios.

Existen interfaces gráficas para navegar y explorar el sistema como se muestra abajo, pero la manera más poderosa y flexible de hacerlo es mediante comandos como **cd**, **ls** y **pwd**



Vista parcial del árbol de directorios usando el ambiente gráfico gnome con el navegador Nautilus

El sistema de archivos - su navegación

Comandos básicos: **pwd**, **ls**, **cd**

```
vinuesa@yaxche[~]$ pwd # imprime directorio actual
/home/vinuesa
```

```
vinuesa@yaxche[~]$ ls / # lista contenidos directorio raíz
bin  etc  initrd.img  lib64  mnt  root  selinux  tmp  vmlinuz
boot  export  initrd.img.old  lost+found  opt  run  srv  usr  vmlinuz.old
dev  home  lib  media  proc  sbin  sys  var
```

```
vinuesa@yaxche[~]$ cd /; pwd # separa múltiples comandos con ;
/
```

```
vinuesa@yaxche[~]$ ls
bin  etc  initrd.img  lib64  mnt  root  selinux  tmp  vmlinuz
boot  export  initrd.img.old  lost+found  opt  run  srv  usr  vmlinuz.old
dev  home  lib  media  proc  sbin  sys  var
```

```
vinuesa@yaxche[~]$ cd; pwd # cd sin argumento nos regresa a $HOME
vinuesa@yaxche[~]$ pwd
/home/vinuesa
```

Introducción al biocómputo en sistemas UNIX/Linux

- Comandos y conceptos básicos

I.- trabajando con archivos

- ls** - lista información sobre archivos y directorio

opciones importantes:

- a display all, including hidden files .file.txt
- d display information about directory
- l long format
- F classify
- h human readable sizes
- r reverse sorted order
- R recursively lists subdirectories
- s display sorted by size
- t display sorted by creation time
- x display files sorted by lines (default is by columns)
- 1 display files one per line

Ejemplo:

```
ls -ltr # imprime lista de archivos ordenados reversamente por tiempo de modificación
```

Introducción al biocómputo en sistemas UNIX/Linux

- Comandos y conceptos básicos

Explorando la estructura de directorios con el comando ls

```
pablo@Tenerife:~$ ls -F / # exploremos el directorio raíz '/'
bin/ cdrom@ etc/ initrd.img@ lib/ lib64@ media/ opt/
root/ selinux/ sys/ usr/ vmlinuz@ xorg.conf.new boot/ dev/
home/ initrd.img.old@ lib32/ lost+found/ mnt/ proc/ sbin/ srv/
tmp/ var/ vmlinuz.old@
```

```
pablo@Tenerife:~$ ls -F /bin # veamos el contenido de /bin
bash*          bzless@      dbus-cleanup-sockets*  egrep*       kbd_mode*    ls*
bunzip2*       bzmore*      dbus-daemon*           false*       kill*         lsmount*
busybox*       cat*         dbus-uidgen*           fgconsole*   ksh@         mkdir*
bzcat*         chgrp*       dd*                    fgrep*       less*         mknod*
bzcmp@         chmod*       df*                    fuser*       lessecho*    mktemp*
bzdiff*        chown*       dir*                   fusermount*  lessfile@    more*
bzegrep@       chvt*        dmesg*                 grep*        lesskey*     mount*
bzexe*         cp*          dnsdomainname*        gunzip*      lesspipe*    mountpoi
bzfgrep@       cpio*        domainname*            gzexe*       ln*           mt@
bzgrep*        csh@         dumpkeys*              gzip*        loadkeys*    mt-gnu*
bzip2*         dash*        echo*                  hostname*    login*       mv*
bzip2recover* date*        ed*                   ip*          lowntfs-3g*  nano*
```

... y muchos más

Introducción al biocómputo en sistemas UNIX/Linux

- Comandos y conceptos básicos

I.- trabajando con archivos - ¿dónde estoy en el árbol de directorios?

- pwd** - print working directory

```
-bash-3.1$ pwd
/home/vinuesa/public_html/tlem09
-bash-3.1$ ls -lFtr
total 472
drwxr-xr-x 2 vinuesa cifn-ux 1024 Apr  3 19:21 css/
drwxr-xr-x 2 vinuesa cifn-ux 1024 Apr 13 14:12 images/
-rw----- 1 vinuesa cifn-ux 15877 May 14 01:22 index.html.save
drwxr-xr-x 2 vinuesa cifn-ux 1024 May 25 12:03 docs/
-rw-r--r-- 1 vinuesa cifn-ux 17047 Jun 10 11:36 index.html
-rw-r--r-- 1 vinuesa cifn-ux 17640 Jun 10 11:58 recursos_bioinformatica.l
-rw-r--r-- 1 vinuesa cifn-ux 121862 Jun 13 14:53 posters_new.html
-rw-r--r-- 1 vinuesa cifn-ux 10061 Jun 14 00:46 profesores.html
drwxr-xr-x 2 vinuesa cifn-ux  96 Jun 15 18:08 perl_scripts/
...
-bash-3.1$
```

Introducción al biocómputo en sistemas UNIX/Linux

- Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y directorios (permisos)

```
bash-3.1$ ls -lFtr
-rw-r--r-- 1 vinuesa cifn-ux 10061 Jun 14 00:46 profesores.html # - archivo
drwxr-xr-x 2 vinuesa cifn-ux  96 Jun 15 18:08 perl_scripts/ # d directorio
```

modo de acceso o permisos:
 ↑ links
 dueño grupo bytes Fecha y hora nombre de archivo -rw-r-
 modificación Nombre de directorio drwxr

usuario
 grupo
 otros

Parámetros para el comando **chmod** (change mode)

User	Type	Rights
u - user	+ add	r - read
g - group	- delete	w - write
O - others		x - execute
a - all		

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos (permisos)

```
bash-3.1$ ls -lFtr
-rw-r--r-- 1 vinuesa cifn-ux 10061 Jun 14 00:46 profesores.html # - archivo
drwxr-xr-x 2 vinuesa cifn-ux   96 Jun 15 18:08 perl_scripts/   # d directorio
```

modo de acceso o permisos: ↑
links

usuario
grupo
otros

Parámetros para el comando **chmod** (change mode)

User	Group	Others	Read	Write	Execute
rwx	rwx	rwx	=4	=2	=1
4	2	1			

Ej. **chmod 651 archivo** -> lo hace u: r+w; g: r+x; o: x

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y cambio de permisos de acceso

```
-bash-3.1$ ls -l
drwxr-xr-x 2 vinuesa cifn-ux 1.0K Jun 20 20:13 .
drwxr-xr-x 8 vinuesa cifn-ux 1.0K Jun 20 20:13 ..
-rw-r--r-- 1 vinuesa cifn-ux   76 Jun 17 14:00 inscritos_forma_inscr.lliner
-rw-r--r-- 1 vinuesa cifn-ux  392 Jun 14 20:02 poster_lliners.txt

-bash-3.1$ chmod u+x,g-r,o-r inscritos_forma_inscr.lliner

-bash-3.1$ ls -l
-rwx----- 1 vinuesa cifn-ux   76 Jun 17 14:00 inscritos_forma_inscr.lliner
-rw-r--r-- 1 vinuesa cifn-ux  392 Jun 14 20:02 poster_lliners.txt

-bash-3.1$ chmod 760 poster_lliners.txt
-bash-3.1$ ls -l
-rwxr----- 1 vinuesa cifn-ux   76 Jun 17 14:00 inscritos_forma_inscr.lliner
-rw-r--r-- 1 vinuesa cifn-ux  392 Jun 14 20:02 poster_lliners.txt
-bash-3.1$
```

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y cambio de permisos de acceso

Para que un archivo que contiene un programa (sea un binario o un script) pueda ser ejecutado desde cualquier directorio del sistema tiene que cumplir 2 condiciones:

1. El archivo tiene que estar en el PATH
2. El usuario tiene que tener permisos de lectura y ejecución para dicho archivo

Comprueba los permisos de los binarios estándar de Linux ejecutando:

```
vinuesa@vinuesa-laptop:~$ ls -l /bin
total 9040
-rwxr-xr-x 1 root root 959120 Mar 28 12:02 bash
-rwxr-xr-x 3 root root 31112 Dec 15 2011 bunzip2
-rwxr-xr-x 1 root root 1832016 Nov 16 2012 busybox
-rwxr-xr-x 3 root root 31112 Dec 15 2011 bzcat
lrwxrwxrwx 1 root root 6 Dec 15 2011 bzcmp -> bzdiff
...
```

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y cambio de permisos de acceso

- 1) Al escribir un archivo de texto con un editor estándar, como al escribir un programa en Bash o Perl, el sistema operativo por defecto le otorga los permisos **rw-rw-r-- (662)**

Por tanto para poder ejecutar el script necesitamos primero cambiarle los permisos, generalment con

chmod 755 mi_script (rwxr-xr-x)

- 2) Si el script no queda guardado en un directorio del PATH, tendremos que indicar

la ruta de acceso al mismo, sea la ruta absoluta o relativa

./mi_script # desde el dir actual

/ruta/completa/a/mi_script # ruta absoluta

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y cambio de permisos de acceso

Veamos un ejemplo: vamos a escribir y ejecutar nuestro primer script de shell.
Teclea lo siguiente en la terminal:

```
vinuesa@ivory:~/cursos/perl4bioinfo$ cat > hello_shell.sh # enter
echo "Hola $USER!"
echo -n 'hoy es: '; date
echo "usas el shell: $SHELL"
echo -n 'y tu computadora es: '; uname
^D # esto es Ctrl-D
```

```
vinuesa@ivory:~/cursos/perl4bioinfo$ ls -l
-rw-rw-r-- 1 vinuesa vinuesa 133 Aug  7 17:52 hello_shell.sh # (662)
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

1.- trabajando con archivos - atributos de archivos y cambio de permisos de acceso

```
vinuesa@ivory:~/cursos/perl4bioinfo$ ls -l
-rw-rw-r-- 1 vinuesa vinuesa 133 Aug  7 17:52 hello_shell.sh # (662)
```

Por tanto para poder ejecutar el script necesitamos primero cambiarle los permisos,

generalment con **chmod 755 mi_script**

```
vinuesa@ivory:~/cursos/perl4bioinfo$ chmod 755
hello_shell.sh
vinuesa@ivory:~/cursos/perl4bioinfo$ ls -l
-rwxr-xr-x 1 vinuesa vinuesa 133 Aug  7 17:52
hello_shell.sh
```

y ahora corro el script

```
vinuesa@ivory:~/cursos/perl4bioinfo$ ./hello_shell.sh
Hola vinuesa!
hoy es: Wed Aug  7 18:03:31 CDT 2013
usas el shell: /bin/bash
y tu computadora es: Linux
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

1.- trabajando con archivos - sustitución de nombres de archivos

```
-bash-3.1$ ls
inscritos_forma_inscr.lliner  posters_old13Jun.html  posters_old4.html
poster_lliners.txt           posters_old3.html      posters_old.html

-bash-3.1$ ls *txt # lista sólo los archivos terminados en .txt
poster_lliners.txt

-bash-3.1$ ls *old?.html
posters_old3.html posters_old4.html
-bash-3.1$

-bash-3.1$ ls *[0-9]*
inscritos_forma_inscr.lliner poster_lliners.txt posters_old13Jun.html
posters_old3.html posters_old4.html

-bash-3.1$ rm *[2-9]* # Elimina todos los archivos que contien
# dígitos del 2-9 en sus nombres

-bash-3.1$ ls
posters_old.html poster_lliners.txt posters_old.html

-bash-3.1$ cp *.* ~/temp # copia los archivos que quedan a ~/temp
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

1.- trabajando con archivos

• cat - Visualizar o concatenar el contenido de archivos

Usage: cat [OPTION] [FILE]...
Concatenate FILE(s), or standard input, to standard output.

-b, --number-nonblank	number nonblank output lines
-n, --number	number all output lines
-s, --squeeze-blank	never more than one single blank line
-T, --show-tabs	display TAB characters as ^I

• **nedit** es un buen editor gráfico;
prueba a correr: **nedit hello_shell.sh &**

• **(vi)/vim**, son los editores estándar de UNIX/Linux (no gráfico), también **pico** y **emacs**

• **less** - es un paginador (muestra archivos por pantalla)

• **more** - es otro paginador (más viejo y con menor funcionalidad, por tanto: "less is more")

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

1.- trabajando con archivos

- **less** - un paginador con muchas opciones (ver 'man less')

```

MOVING

f  ^F  ^V  SPACE  * Forward one window (or N lines).
b  ^B  ESC-v  * Backward one window (or N lines).
nG                               * Go TO LINE no. N
-----
SEARCHING

/pattern      * Search forward for (N-th) matching line.
?pattern      * Search backward for (N-th) matching line.
-----
QUIT

q

Ejemplo:
pablo@Tenerife:~$ less /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
...

```

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

2. - trabajando con directorios: rutas absolutas y relativas

- La **ruta absoluta** de un archivo o directorio es aquella que **apunta hacia éste desde el directorio raíz /**, tal y como se ve en la salida del comando pwd.
- La **ruta relativa** es aquella que parte del directorio actual, el cual se representa por un punto **.** y el directorio parental (uno arriba) mediante dos puntos **..**

```

# iniciamos desde este directorio: /home/vinuesa/cursos/perl4bioinfo

# vamos a nuestro $HOME usando la ruta relativa, es decir, 2 dirs arriba
vinuesa@ivory:~/cursos/perl4bioinfo$ cd ../../
vinuesa@ivory:~$ pwd
/home/vinuesa
vinuesa@ivory:~$ cd cursos/perl4bioinfo/ # volvemos al dir perl4bioinfo

# vamos a nuestro $HOME usando la ruta absoluta, es decir, desde /
vinuesa@ivory:~/cursos/perl4bioinfo$ cd /home/vinuesa
vinuesa@ivory:~$ pwd
/home/vinuesa
vinuesa@ivory:~$

```

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

2. - trabajando con directorios: rutas absolutas y relativas

- Siempre podemos regresar a nuestro home tecleando cualquiera de las siguientes órdenes:

```

# iniciamos desde este directorio: /home/vinuesa/cursos/perl4bioinfo

# 1) vamos a nuestro $HOME usando cd sin argumento
vinuesa@ivory:~/cursos/perl4bioinfo$ cd
vinuesa@ivory:~$ pwd
/home/vinuesa
vinuesa@ivory:~$ cd cursos/perl4bioinfo/ # volvemos al dir perl4bioinfo

# 2) vamos a nuestro $HOME usando cd ~
vinuesa@ivory:~/cursos/perl4bioinfo$ cd ~
vinuesa@ivory:~$ pwd
/home/vinuesa
vinuesa@ivory:~$ cd cursos/perl4bioinfo/ # volvemos al dir perl4bioinfo

# 3) haciendo cd $HOME
vinuesa@ivory:~$ cd $HOME
vinuesa@ivory:~$ pwd
/home/vinuesa

```

Introducción al biocómputo en sistemas UNIX/Linux

Comandos y conceptos básicos

2. - trabajando con directorios: rutas absolutas y relativas

- Más ejemplos del uso de rutas absolutas y relativas:

```

# estamos en: /home/vinuesa/cursos/perl4bioinfo

# 1) queremos ver contenido del directorio /usr/bin
vinuesa@ivory:~$ ls /usr/bin # [ó: ls ../../../../usr/bin]

# 2) queremos copiar el archivo hello_shell.sh a $HOME/bin
vinuesa@ivory:~$ cp hello_shell.sh $HOME/bin
[ó: cp hello_shell.sh ../../bin]
[ó: cp hello_shell.sh ~/bin]

```

El usuario decide qué es más práctico, usar rutas relativas o absolutas. El punto está en que podemos ejecutar cualquier comando que lee o escribe archivos desde un directorio diferente al actual, y poner el resultado del comando en el directorio que nos convenga.

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

2. - trabajando con directorios: generación de directorios

Podemos en UNIX/Linux ejecutar varios comandos en una sola línea, separándolos con ;
Si la línea es muy larga, podemos introducir saltos de línea (return), escapándolo con \

mkdir => genera directorio; no dejar espacios en blanco en el nombre, usar guión bajo!!!
cd => cambia al directorio
touch => genera archivos vacíos, para nuestra práctica
mv => mueve o renombra un archivo o directorio

```
vinuesa@ivory:~$ mkdir practicas_UNIX; cd practicas_UNIX; \
touch file1.txt file2.txt file3.txt; cd ../; ls -laF practicas_UNIX;
total 0
drwxrwxr-x 2 vinuesa vinuesa 4096 Aug  8 11:52 ./
drwxr-xr-x 6 vinuesa vinuesa 4096 Aug  8 11:52 ../
-rw-rw-r-- 1 vinuesa vinuesa  0 Aug  8 11:52 file1.txt
-rw-rw-r-- 1 vinuesa vinuesa  0 Aug  8 11:52 file2.txt
-rw-rw-r-- 1 vinuesa vinuesa  0 Aug  8 11:52 file3.txt

vinuesa@ivory:~$ ls -d practicas_UNIX/
practicas_UNIX/
vinuesa@ivory:~$ ls practicas_UNIX/
file1.txt file2.txt file3.txt
vinuesa@ivory:~$ mkdir borrame; mv practicas_UNIX/ borrame/
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

2. - trabajando con directorios: copiado y borrado de directorios

```
# estamos en /home/vinuesa y revisamos el contenido del dir
vinuesa@ivory:~$ ls
borrame cursos
```

```
# veamos el contenido de borrame
vinuesa@ivory:~$ ls -F borrame/
practicas_unix/
```

```
# ahora copiamos practicas_unix al directorio actual ($HOME)
vinuesa@ivory:~$ cp -r borrame/practicas_unix .
```

```
# borramos el directorio borrame
vinuesa@ivory:~$ rm -rf borrame
```

```
# vemos contenido del dir practicas_unix y borramos todos los
archivos que contiene
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

2. - trabajando con directorios: copiado y borrado de directorios

estamos en /home/vinuesa y revisamos el contenido del dir
practicas_unix y borramos todos los archivos que contiene

```
vinuesa@ivory:~$ ls -lF practicas_unix/
total 0
-rw-rw-r-- 1 vinuesa vinuesa 0 Aug  8 12:15 file1.txt
-rw-rw-r-- 1 vinuesa vinuesa 0 Aug  8 12:15 file2.txt
-rw-rw-r-- 1 vinuesa vinuesa 0 Aug  8 12:15 file3.txt
```

```
vinuesa@ivory:~$ rm practicas_unix/*.*
vinuesa@ivory:~$ ls -lF practicas_unix/
total 0
vinuesa@ivory:~$
```

```
# una vez vacío, podemos usar rmdir para eliminar el dir vacío
vinuesa@ivory:~$ rmdir practicas_unix/ # o usar rm -rf pract*
vinuesa@ivory:~$ ls
Cursos
vinuesa@ivory:~$
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos

2. - trabajando con archivos y directorios: resumen

- Podemos ejecutar varios comandos en una sola línea, separándolos con ;
- Si la línea es muy larga, podemos introducir saltos de línea, escapándolo con \

mkdir => genera directorio; no dejar espacios en blanco en el nombre, usar guión bajo!!!

cd => cambia al directorio

touch => genera archivos vacíos, para nuestra práctica

mv file2 dir1; mv file3 file1 => mueve o renombra un archivo o directorio

cp file1 dir1 => copia file1 a dir1

cp -r dir1 dir2 => copia dir1 y su contenido (-r recursivamente) a dir2

rm dir1/file1 => borra file1 en dir1/

rm -rf dir1 dir2 => elimina los directorios dir1 y dir2

find . -type d => muestra recursivamente los subdirectorios del directorio indicado

ls -d dir* => muestra sólo los nombres de los directorios que empiecen por dir

ls dir1 => muestra el contenido del directorio dir1

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - I/O****3.- Standard Input / Standard Output**

```
# salida del comando who a STDOUT (pantalla por lo general)
-bash-3.1$ who
root pts/1 2009-03-17 12:39 (:0.0)
heladia pts/3 2009-05-18 23:55 (itzamna.ccg.unam.mx)
vinuesa pts/4 2009-06-20 19:36 (uxmal.ccg.unam.mx)
rzayas pts/7 2009-06-02 10:57 (kay.ccg.unam.mx)
rzayas pts/8 2009-06-02 10:58 (kay.ccg.unam.mx)

# redireccionamos la salida de who a un archivo con >
-bash-3.1$ who > users.out

# podemos ver el contenido de users.out con cat ó less ó more
-bash-3.1$ cat users.out
root pts/1 2009-03-17 12:39 (:0.0)
heladia pts/3 2009-05-18 23:55 (itzamna.ccg.unam.mx)
vinuesa pts/4 2009-06-20 19:36 (uxmal.ccg.unam.mx)
rzayas pts/7 2009-06-02 10:57 (kay.ccg.unam.mx)
rzayas pts/8 2009-06-02 10:58 (kay.ccg.unam.mx)
-bash-3.1$
```

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - I/O****3.- Standard Input / Standard Output**

```
# uso del comando cat para concatenar texto o archivos al final de otro usando >>
# tecleamos el comando y enter; escribimos el texto y salimos con Ctrl-D
-bash-3.1$ cat >> users.out
estas son líneas adicionadas al final del archivo gracias a '>>'
y una segunda línea
y una tercera

# veamos el contenido de users.out con less
bash-3.1$ less users.out # (equivalente a less < users.out)
root pts/1 2009-03-17 12:39 (:0.0)
heladia pts/3 2009-05-18 23:55 (itzamna.ccg.unam.mx)
vinuesa pts/4 2009-06-20 19:36 (uxmal.ccg.unam.mx)
rzayas pts/7 2009-06-02 10:57 (kay.ccg.unam.mx)
rzayas pts/8 2009-06-02 10:58 (kay.ccg.unam.mx)
estas son líneas adicionadas al final del archivo gracias a '>>'
y una segunda línea
y una tercera
(END)

# de esta manera añadimos el contenido de file1 al final de users.out
-bash-3.1$ cat file1 >> users.out
```

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - I/O****3.- Standard Input / Standard Output - más ejemplos de I/O con cat**

```
# el comando echo imprime a STDOUT su argumento(s); redirigimos salida a archivo1.txt
# simplemente para tener un archivo de texto con contenido
-bash-3.1$ echo 'línea uno' > archivo1.txt
-bash-3.1$ less archivo1.txt
línea uno

# lo mismo lo podemos hacer con el comando cat, como ya hemos visto anteriormente
-bash-3.1$ cat > archivo2.txt
línea dos
^D # usen CTRL-D para interrumpir la escritura a archivo2.txt con cat
-bash-3.1$ cat archivo2.txt archivo1.txt # concatena ambos archivos
línea dos
línea uno

# redirigimos con > la salida de cat a un archivo
-bash-3.1$ cat archivo2.txt archivo1.txt > archivos2-1_concatenados.txt

-bash-3.1$ cat archivos2-1_concatenados.txt
línea dos
línea uno
```

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - Pipes (tuberías)****4.- pipes '|' - conecta la salida (stdout) de un comando directamente con la entrada estándar (stdin) de otro comando, filtrando la salida del primero por el segundo programa. Típicos programas de filtrado son grep, cut, sort, sed, awk, head, tail ...**

```
# veamos estos comandos de filtrado en acción usando el archivo /etc/passwd
vinuesa@ivory:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
...

# cuantas entradas o líneas hay en dicho archivo? => contamos con wc
vinuesa@ivory:~$ cat /etc/passwd | wc
115 147 5876 # líneas palabras caracteres

# con opción -l cuenta sólo las líneas
vinuesa@ivory:~$ cat /etc/passwd | wc -l
115
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos - herramientas de filtrado usadas en pipes

7.- Veremos las siguientes herramientas: **grep**, **cut**, **sort** y **uniq** (exploren uds. `tr` y `sed`)

• **grep**

```
-bash-3.1$ grep --help # selección de opciones (hay muchas más)
Usage: grep [OPTION]... PATTERN [FILE] ...
Search for PATTERN in each FILE or standard input.
Example: grep -i 'hello world' menu.h main.c
```

Regex selection and interpretation:

```
-P, --perl-regexp    PATTERN is a Perl regular expression
-e, --regexp=PATTERN use PATTERN as a regular expression
-f, --file=FILE      obtain PATTERN from FILE
-i, --ignore-case    ignore case distinctions
-w, --word-regexp    force PATTERN to match only whole words
-x, --line-regexp    force PATTERN to match only whole lines
```

Miscellaneous:

```
-v, --invert-match   select non-matching lines
```

Output control:

```
-n, --line-number    print line number with output lines
--line-buffered      flush output on every line
-L, --files-without-match only print FILE names containing no match
-l, --files-with-matches only print FILE names containing matches
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos - herramientas de filtrado usadas en pipes

7.- Veremos las siguientes herramientas: **grep**, **cut**, **sort** y **uniq** (exploren uds. `tr` y `sed`)

• **cut**

```
-bash-3.1$ cut --help # selección de opciones (hay muchas más)
```

```
Usage: cut [OPTION]... [FILE]...
```

Print selected parts of lines from each FILE to standard output.

Mandatory arguments to long options are mandatory for short options too.

```
-c, --characters=LIST select only these characters
-d, --delimiter=DELIM use DELIM instead of TAB for field delimiter
-f, --fields=LIST      select only these fields; also print any line
                       that contains no delimiter character, unless
                       the -s option is specified
With no FILE, or when FILE is -, read standard input.
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos - herramientas de filtrado usadas en pipes

7.- Veremos las siguientes herramientas: **grep**, **cut**, **sort** y **uniq** (exploren uds. `tr` y `sed`)

• **sort**

```
-bash-3.1$ sort --help # selección de opciones (hay algunas más)
Usage: sort [OPTION]... [FILE]...
Write sorted concatenation of all FILE(s) to standard output.
```

Mandatory arguments to long options are mandatory for short options too.
Ordering options:

```
-b, --ignore-leading-blanks ignore leading blanks
-d, --dictionary-order      consider only blanks and alphanumeric characters
-f, --ignore-case           fold lower case to upper case characters
-g, --general-numeric-sort compare according to general numerical value
-i, --ignore-nonprinting    consider only printable characters
-M, --month-sort            compare (unknown) < 'JAN' < ... < 'DEC'
-n, --numeric-sort          compare according to string numerical value
-r, --reverse               reverse the result of comparisons
```

Other options:

```
-m, --merge                merge already sorted files; do not sort
-o, --output=FILE           write result to FILE instead of standard output
-t, --field-separator=SEP  use SEP instead of non-blank to blank transition
-u, --unique                with -c, check for strict ordering;
                           without -c, output only the first of an equal run
```

Introducción al biocómputo en sistemas UNIX/Linux

• Comandos y conceptos básicos - herramientas de filtrado usadas en pipes

• **uniq**

```
-bash-3.1$ uniq --help
Usage: uniq [OPTION]... [INPUT [OUTPUT]]
Discard all but one of successive identical lines from INPUT (or
standard input), writing to OUTPUT (or standard output).
```

Mandatory arguments to long options are mandatory for short options too.

```
-c, --count              prefix lines by the number of occurrences
-d, --repeated           only print duplicate lines
-D, --all-repeated[=delimit-method] print all duplicate lines
                        delimit-method={none(default),prepend,separate}
                        Delimiting is done with blank lines.
-f, --skip-fields=N      avoid comparing the first N fields
-i, --ignore-case        ignore differences in case when comparing
-s, --skip-chars=N       avoid comparing the first N characters
-u, --unique             only print unique lines
-w, --check-chars=N      compare no more than N characters in lines
--help                  display this help and exit
--version                output version information and exit
```

A field is a run of whitespace, then non-whitespace characters.
Fields are skipped before chars.

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - Pipes (tuberías)**comandos de filtrado (**grep**, **wc**) en acción usando el archivo `/etc/passwd` (cont.)

```
# cuantas entradas en /etc/passwd corresponden a cuentas de usuario ($HOME)
vinesa@ivory:~$ cat /etc/passwd | grep home | wc -l
89

# cuantas entradas en /etc/passwd corresponden a cuentas de usuario
# ($HOME) que NO correspondan a curso?
vinesa@ivory:~$ cat /etc/passwd | grep home | grep -v curso | wc -l
39

# cuantos usuarios usan el bash y cuántos usan otro shell?
vinesa@ivory:~$ cat /etc/passwd | grep home | grep -c bash
81
vinesa@ivory:~$ cat /etc/passwd | grep home | grep -vc bash
8
```

Introducción al biocómputo en sistemas UNIX/Linux**Comandos y conceptos básicos - Pipes (tuberías)**comandos de filtrado (**grep**, **cut**, **sort**, **uniq**) en acción
usando el archivo `/etc/passwd` (cont.)

```
# muestra los usuarios que no usan bash como shell ordenados alfabéticamente
vinesa@ivory:~$ cat /etc/passwd | grep home | grep -v bash | cut -d: -f1,7 | sort
alemc:/bin/tcsh
cemg:/bin/tcsh
javiermb:/bin/tcsh
jmanuel:/bin/tcsh
#nan:/bin/tcsh
syslog:/bin/false
#viri:/bin/tcsh
zuemy:/bin/tcsh

# genera estadísticas de uso de shell para todas las entradas en /etc/passwd
vinesa@ivory:~$ cat /etc/passwd | cut -d: -f7 | sort | uniq -c
82 /bin/bash
6 /bin/false
17 /bin/sh
1 /bin/sync
7 /bin/tcsh
2 /usr/sbin/nologin
```

Ejercicios de tarea:

- 1) Si no tienes una terminal UNIX ó Linux, instala Ubuntu o putty y conéctate al servidor ivory con tu cuenta. Se en ivory o en tu compu, repite todos los ejemplos de comandos de esta presentación.
- 2) Genera un directorio llamado bin en tu \$HOME. Instala y corre en él el script `hello_shell.sh` desde tu y captura la salida en un archivo llamado `hello_shell.out`. Visualiza el contenido de dicho archivo con alguno de los comandos aprendidos.
- 3) Cuenta cuántos archivos binarios (programas) existen en los directorios `/bin` y `/usr/bin`.

Has de entregar en un archivo de texto todos los comandos y sus salidas.