

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)

Formato para prácticas de laboratorio

PROGRAMA EDUCATIVO	PLAN DE ESTUDIO	CLAVE DE UNIDAD DE APRENDIZAJE	NOMBRE DE LA UNIDAD DE APRENDIZAJE
Ingeniero en Computación	2009-2	12124	Taller de Sistema Operativo Unix

PRÁCTICA No.	LABORATORIO DE	Taller de Sistema Operativo Unix	DURACIÓN (HORAS)
7	NOMBRE DE LA PRÁCTICA	Mandos para visualizar secciones de archivo	2

1. INTRODUCCIÓN

En ocasiones queremos visualizar únicamente parte del contenido de un archivo por lo que en esta práctica se aprenderá a utilizar aquellas órdenes que son ideales para este tipo de acciones como son *head*, *tail* y *cut*, éstas se utilizan para visualizar parte de un archivo, ya sea una línea o una sección completa, además también se cuenta con la orden *wc* la cual sirve para contar líneas, palabras y caracteres.

2. OBJETIVO (COMPETENCIA)

Aplicar los mandos pertinentes para resolver problemas que requieran conocer la cantidad de líneas, palabras y caracteres contenidas en un archivo o extraer las primeras o últimas líneas de un archivo.

3. FUNDAMENTO

Mando **head**
 sintaxis:

`head -[ncqv] nombrearchivo(s)`

Formuló LSC. Natalia Rodríguez Castellón, MSC. Elvia C. Márquez Salgado	Revisó M.C. Aglay González Pacheco	Aprobó	Autorizó Dr. David I. Rosas Almeida
Nombre y Firma del Maestro	Nombre y Firma del Responsable de Programa Educativo	Nombre y Firma del Responsable de Gestión de Calidad	Nombre y Firma del Director / Representante de la Dirección

Código: GC-N4-017 Revisión: 4

La orden **head** se utiliza para visualizar las **n** primeras líneas de un archivo, en ocasiones es muy útil ya que no es necesario visualizar todo el contenido del archivo sobre todo cuando estos suelen ser muy extensos. Por defecto **head** muestra las 10 primeras líneas si no le indicamos la cantidad, para lo cual se escribiría lo siguiente:

head nombreadarchivo

Opciones de la orden *head*

head -n archivo Muestra las primeras n líneas del archivo. Donde n representa el numero de líneas que se desea visualizar.

head -c n archivo Muestra los primeros n caracteres(bytes) del archivo.

head --help Muestra ayuda sobre como utilizar la orden *head*

head --version Muestra la versión que esta siendo utilizada de la orden head

Ejemplos:

Muestra las primeras 10 líneas del archivo programa.c

```
$ head programa.c
#include<sys/types.h>
#include<sys/stat.h>
#include<errno.h>
#include<stdio.h>
#include<stdlib.h>
#include<fcntl.h>
#include<limits.h>
#include<string.h>
#include<math.h>

$
```

Muestra las primeras 5 líneas del archivo programa.c

```
$ head -5 programa.c
#include<sys/types.h>
#include<sys/stat.h>
#include<errno.h>
#include<stdio.h>
#include<stdlib.h>

$
```

Muestra los primeros 50 bytes del archivo programa.c

```
$ head -c 50 programa.c
#include<sys/types.h>
#include<sys/stat.h>
#include$
```

Muestra la ayuda de la orden head

```
$ head --help
Usage: head [OPTION]... [FILE]...
Print the first 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.
With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=[-]K      print the first K bytes of each file;
                        with the leading '-', print all but the last
                        K bytes of each file
  -n, --lines=[-]K      print the first K lines instead of the first 10;
                        with the leading '-', print all but the last
                        K lines of each file
  -q, --quiet, --silent  never print headers giving file names
  -v, --verbose          always print headers giving file names
      --help            display this help and exit
      --version         output version information and exit
```

Muestra la versión de la orden head

```
$ head --version
head (GNU coreutils) 8.13
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie and Jim Meyering.
$
```

Mando **tail**

sintaxis:

tail *[-[ncqv] nombrearchivo(s)*

La orden **tail** se utiliza para visualizar las **n** últimas líneas de un archivo, en ocasiones es muy útil ya que algunas veces solo nos interesa conocer la cantidad de líneas o simplemente visualizar en que termina el contenido de nuestro archivo. Por defecto, **tail** muestra las 10 últimas líneas si no le indicamos la cantidad, para lo cual se escribiría lo siguiente:

tail nombrearchivo

Opciones de la orden **tail**

tail -n archivo Muestra las primeras n líneas del archivo. Donde n representa el numero de lineas que se desea visualizar.

tail -c n archivo	Muestra los últimos n caracteres(bytes) del archivo.
tail --help	Muestra ayuda sobre como utilizar la orden <i>tail</i>
tail --version	Muestra la versión que esta siendo utilizada de la orden tail

Ejemplos:

Muestra las últimas 10 líneas del archivo programa.c

```
$ tail programa.c
    strcpy(aux, "");
    // for(l=0;l<=9;l++)
    // {
    //     strcat(temp,u[l]);
    printf("aux= %s\n",temp);
    // }
} //if l=1
//aux=temp;
printf("aux: %s\n",aux);
} //funcion
$
```

Muestra las últimas 15 líneas del archivo programa.c

```
$ tail -15 programa.c
    //strcat(temp,'\0');
    // printf("u[%d]=%s\n",t,u[t]);
    // strcat(u[t],'\0');

    aux=u[t];
    strcpy(aux, "");
    // for(l=0;l<=9;l++)
    // {
    //     strcat(temp,u[l]);
    printf("aux= %s\n",temp);
    // }
} //if l=1
//aux=temp;
printf("aux: %s\n",aux);
} //funcion
$
```

Muestra los últimos 100 bytes del archivo programa.c

```
$ tail -c 100 programa.c
aux= %s\n",temp);
    // }
} //if l=1
//aux=temp;
printf("aux: %s\n",aux);
} //funcion
$
```

Muestra la ayuda de la orden tail

```
$ tail --help
Usage: tail [OPTION]... [FILE]...
Print the last 10 lines of each FILE to standard output.
With more than one FILE, precede each with a header giving the file name.
With no FILE, or when FILE is -, read standard input.

Mandatory arguments to long options are mandatory for short options too.
  -c, --bytes=K          output the last K bytes; alternatively, use -c +K
                        to output bytes starting with the Kth of each file
  -f, --follow[={name|descriptor}]
                        output appended data as the file grows;
                        -f, --follow, and --follow=descriptor are
                        equivalent
  -F                      same as --follow=name --retry
  -n, --lines=K          output the last K lines, instead of the last 10;
                        or use -n +K to output lines starting with the Kth
                        --max-unchanged-stats=N
```

Muestra la versión de la orden tail

```
$ tail --version
tail (GNU coreutils) 8.13
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by Paul Rubin, David MacKenzie, Ian Lance Taylor,
and Jim Meyering.
$
```

Mando **wc**

sintaxis:

wc *[/lwc]* nombreadarchivo(s)

La orden **wc** sirve para contar líneas, palabras y caracteres de un archivo.

Opciones de la orden wc

wc -l nombreadarchivo	Muestra la cantidad de líneas que contiene el archivo.
wc -w nombreadarchivo	Muestra la cantidad de palabras que contiene el archivo.
wc -c nombreadarchivo	Muestra la cantidad de caracteres que tiene el archivo.
wc -L nombreadarchivo	Muestra la longitud de la línea mas larga que contiene el archivo.
wc --help	Proporciona ayuda sobre la orden wc

En caso de que no se le proporcione ninguna opción a la orden wc tomará por defecto *lwc*, esto quiere decir que mostrará en ese orden el número de líneas, palabras y caracteres.

Ejemplos:

```
$ wc lista.txt
 81  540 3951 lista.txt
$
```

En este ejemplo no se le indicó a **wc** alguna opción por lo que muestra que tiene **81 líneas, 540 palabras y 3951 caracteres**.

```
$ wc -l lista.txt
81 lista.txt
$
```

En este caso **wc** indica que **lista.txt** tiene 81 líneas de texto.

```
$ wc -w lista.txt
540 lista.txt
$
```

Ahora se muestra la cantidad de palabras que contiene el archivo **lista.txt**

```
$ wc -c lista.txt
3951 lista.txt
$
```

En este ejemplo **wc** nos indica la cantidad de caracteres que contiene el archivo **lista.txt**

```
$ wc -L lista.txt
82 lista.txt
$
```

Por último **wc** nos muestra la longitud que tiene la línea mas larga que contiene el archivo **lista.txt**

El mando **cut**

sintaxis:

cut -f n [-dchar] archivo(s)

La orden **cut** se utiliza para cortar y pasar a la salida estándar las columnas o campos del archivo especificado.

La opción **-f** se utiliza para indicar los campos o columnas que se desean cortar.

La opción **-d** sirve para especificarle a la orden **cut** el delimitador(carácter de separación) que se tiene entre los campos en caso de que el archivo lo contenga. Por defecto, este delimitador es el espacio en blanco y no es necesario indicarlo.

Para especificar las columnas o campos que deseamos cortar se utiliza una lista. Una lista es una secuencia de números para indicarle a **cut** que campos o columnas se quieren cortar. Hay varios formatos para esta lista:

cut -f número archivo

Solo selecciona el campo que se indique en número.

cut -f inicio-fin archivo

Selecciona el rango de las columnas, desde el valor inicial al final.

cut -f col1,col2,col3 archivo

Selecciona las columnas que se especifican separadas por comas.

cut -d caracter -f número archivo

Indica el carácter que se usará como delimitador.

Para comprender el uso de la orden **cut**, veremos algunos ejemplos utilizando el archivo llamado Personas el cual tiene el siguiente contenido:

```
$ cat Personas
101:Natalia:Morado
123:Cristina:Azul
134:Eva:Cafe
145:Angelica:Rosa
156:Laura:Verde
$
```

Como se ve el contenido de este archivo tiene un delimitador(separador) entre las columnas(campos) que contiene, por lo que hay que indicárselo a la orden cut.

```
$ cut -d: -f1 Personas
101
123
134
145
156
$
```

En este ejemplo solo muestra la columna 1, tomando como delimitador los dos puntos :

```
$ cut -d: -f1,3 Personas
101:Morado
123:Azul
134:Cafe
145:Rosa
156:Verde
$
```

En este otro ejemplo le indicamos a la orden cut que corte las columnas 1 y 3, tomando como delimitador los 2 puntos :

```
$ cut -d: -f1-3 Personas
101:Natalia:Morado
123:Cristina:Azul
134:Eva:Cafe
145:Angelica:Rosa
156:Laura:Verde
$
```

Por último le indicamos a cut que corte las columnas de la 1 a la 3 y las envía a la salida estándar.

Ahora veamos un ejemplo utilizando el archivo lista.txt donde el contenido del archivo no contiene delimitador(separador) entre columnas.

Primero visualicemos el contenido del archivo lista.txt

```
$ cat lista.txt
No.      NOMBRE      MATRICULA      CARRERA
1        AGUILAR ALVAREZ LESLIE KARIM 1/67020 INDUSTRIAL
2        ARCEO ESCALANTE ALEJANDRO 1/67150 LSC
3        ESTRADA ROMO PAOLA LIZETH 1/67752 COMPUTACION
4        GARCIA GUZMAN ANA MARGARITA 1/67891 ELECTONICA
5        GONZALEZ VILLALOBOS ILIAN ELIZABETH 1/68024 CIVIL
6        GUTIERREZ GARCIA GABRIELA 1/68054 ELECTRONICA
7        INIGUEZ FLORES MARIA ELENA 1/68196 TOPOGRAFO
8        JIMENEZ OLIVAREZ VICTOR IVAN 1/68224 LSC
9        MARIN SANTANA MARIANA CARLOTA 1/68436 LSC
10       NUNEZ GARCIA JOSE TRINIDAD 1/64561 TOPOGRAFO
11       AGUIRRE MORALES DENSE CRISTINA 1/67040 MECANICO
12       DAVALOS HERMOSILLO FERNANDO 1/65282 COMPUTACION
13       INZUNZA ACOSTA ADAN 1/68189 INDUSTRIAL
14       MEDINA ABRICA ALEJANDRO 1/68525 MECANICO
```

Ahora se mostrara la columna 1 del archivo lista.txt

```
$ cut -f1 lista.txt
No.
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

En el siguiente ejemplo se muestran las columnas de la 1 a la 2 del archivo lista.txt

```
$ cut -f1-2 lista.txt
No.      NOMBRE
1        AGUILAR
2        ARCEO
3        ESTRADA
4        GARCIA
5        GONZALEZ
6        GUTIERREZ
7        INIGUEZ
8        JIMENEZ
9        MARIN
10       NUNEZ
11       AGUIRRE
12       DAVALOS
13       INZUNZA
14       MEDINA
```

4. PROCEDIMIENTO (DESCRIPCIÓN)

A) EQUIPO NECESARIO

Computadoras con Linux instalado

MATERIAL DE APOYO

Práctica impresa

B) DESARROLLO DE LA PRÁCTICA

1. Copie los archivos papeleria.txt, lista.txt y peliculas.txt que se encuentran en */externos/home/clases/compartido* a su directorio de trabajo.
2. Muestre las 2 primeras líneas del archivo papeleria.txt
3. Muestre las 10 primeras líneas del archivo lista.txt
4. Revise las últimas 3 líneas del archivo lista.txt
5. Muestre las 4 últimas líneas del archivo peliculas.txt
6. Practique con el archivo papeleria.txt cada una de las opciones de head
7. En el directorio donde trabaja normalmente cree un subdirectorio llamado Practica8
8. Copie los archivos lista.txt y peliculas.txt al directorio Practica8
9. Muestre las 10 últimas líneas del archivo lista.txt
10. Muestre las 15 primeras líneas del archivo papeleria.txt
11. Muestre la primer columna del archivo lista.txt
12. Cuántas palabras tiene el archivo peliculas.txt?
13. Cuántos caracteres tiene el archivo papeleria.txt?
14. Cuántas líneas tiene el archivo lista.txt ?
15. En un solo paso muestre cuántas líneas, palabras y caracteres tiene el archivo peliculas.txt
16. Cuál es el tamaño de la línea mas grande del archivo lista.txt?
17. Desplegar la columna 1 y 3 del archivo lista.txt
18. Muestre las columnas de la 2 a la 4 del archivo lista.txt
19. Muestre los 50 últimos bytes del archivo papeleria.txt
20. Muestre únicamente el nombre de las películas que se encuentran en el archivo peliculas.txt

C) CÁLCULOS (SI APLICA) Y REPORTE

El alumno deberá mostrar a su profesor los resultados obtenidos de su práctica así como presentar un examen cuando su profesor se lo indique.

5. RESULTADOS Y CONCLUSIONES

El alumno aprenderá a utilizar los mandos head, tail, y cut para poder visualizar secciones de archivos, así como el mando wc para obtener la cantidad de líneas, palabras y caracteres de un archivo(s).

6. ANEXOS

No aplica

7. REFERENCIAS