



Instituto Politécnico Nacional
Escuela Superior de Cómputo
ESCOM



Ingeniería en Inteligencia Artificial

Cómputo paralelo

Práctica 1

El entorno de GNU-Linux

INTEGRANTES

Bustillos Cruz Jonatan
Delgado Lucero Cristian Isaac
Frem Cortés José Angel
Luna Gonzales Gabriel Alexis

Ciudad de México a 17 de febrero del 2026

Índice general

| | |
|--|----------|
| 1. Objetivo | 1 |
| 2. Marco Teórico | 2 |
| 2.1. Máquinas virtuales | 2 |
| 2.2. Distribuciones de GNU-Linux | 2 |
| 2.3. Requisitos de instalación del sistema operativo GNU-Linux | 2 |
| 2.4. Estadísticas de uso del sistema operativo GNU-Linux en el mundo | 2 |
| 2.5. Escritorios GNOME y KDE | 2 |
| 2.6. Entornos Command Line Interface y Graphical User Interface | 2 |
| 2.7. Terminal de GNU-Linux | 2 |
| 2.8. Tipos de usuario en GNU-Linux | 2 |
| 2.9. Rutas relativas y absolutas | 2 |
| 2.10. Redireccionamiento | 2 |
| 2.11. Clasificación de los comandos en GNU-Linux | 2 |
| 2.12. Variables de entorno | 2 |
| 2.13. Comandos de GNU-Linux mostrados en la tabla 1 | 2 |
| 3. Instalación de GNU-Linux | 3 |
| 3.1. Instalación - Bustillos Cruz Jonatan | 3 |
| 3.2. Instalación - Delgado Lucero Cristian Isaac | 3 |
| 3.3. Instalación - Frem Cortés José Angel | 3 |
| 3.4. Instalación - Luna Gonzales Gabriel Alexis | 3 |
| 4. Entorno de GNU-Linux | 4 |
| 4.1. Comandos - Bustillos Cruz Jonatan | 5 |
| 4.1.1. Comando ps | 5 |
| 4.1.2. Comando more | 5 |
| 4.1.3. Comando time | 5 |
| 4.1.4. Comando du | 5 |
| 4.1.5. Comando ps -fea | 5 |
| 4.1.6. Comando less | 5 |
| 4.1.7. Comando uname | 5 |
| 4.1.8. Comando pstree | 5 |
| 4.1.9. Comando man | 5 |
| 4.1.10. Comando mkdir | 5 |
| 4.2. Comandos - Delgado Lucero Cristian Isaac | 5 |
| 4.2.1. Comando cal | 5 |
| 4.2.2. Comando clear | 5 |

| | |
|---|-----------|
| 4.2.3. Comando apt | 5 |
| 4.2.4. Comando rm | 5 |
| 4.2.5. Comando date | 5 |
| 4.2.6. Comando ifconfig | 5 |
| 4.2.7. Comando exit | 5 |
| 4.2.8. Comando mv | 5 |
| 4.2.9. Comando echo | 5 |
| 4.2.10. Comando df | 5 |
| 4.3. Comandos - Frem Cortés José Angel | 5 |
| 4.3.1. Comando w | 5 |
| 4.3.2. Comando kill -l -9 | 6 |
| 4.3.3. Comando cat | 6 |
| 4.3.4. Comando pico | 7 |
| 4.3.5. Comando who | 7 |
| 4.3.6. Comando trap -l | 7 |
| 4.3.7. Comando fg | 8 |
| 4.3.8. Comando nano | 8 |
| 4.3.9. Comando bash | 9 |
| 4.4. Comandos - Luna Gonzales Gabriel Alexis | 10 |
| 4.4.1. Comando pwd | 10 |
| 4.4.2. Comando cd | 10 |
| 4.4.3. Comando vi | 10 |
| 4.4.4. Comando wc | 10 |
| 4.4.5. Comando su | 10 |
| 4.4.6. Comando ls | 10 |
| 4.4.7. Comando apt-get | 10 |
| 4.4.8. Comando sudo | 10 |
| 4.4.9. Comando ls -la | 10 |
| 5. Ejercicios Prácticos | 11 |
| 5.1. Rutas absolutas y relativas | 11 |
| 5.2. Borrado de archivos con comodines | 11 |
| 5.3. Redireccionamiento con > y » | 11 |
| 5.4. Instalación de software por línea de comandos | 11 |
| 5.5. Instalación de software usando el entorno gráfico | 11 |
| 5.6. Mover y copiar archivos en la jerarquía de directorios | 11 |
| 5.7. Gestión de usuarios | 11 |
| 5.8. Hola Mundo en Lenguaje C usando nano | 11 |
| 5.9. Hola Mundo en Lenguaje C usando gedit | 11 |
| 6. Conclusiones | 12 |
| 6.1. Bustillos Cruz Jonatan | 12 |
| 6.2. Delgado Lucero Cristian Isaac | 12 |
| 6.3. Frem Cortés José Angel | 12 |
| 6.4. Luna Gonzales Gabriel Alexis | 12 |

Índice de figuras

| | |
|---|---|
| 4.1. Resultado en terminal del comando w. | 6 |
| 4.2. Resultado en terminal del comando kill -l -9. | 6 |
| 4.3. Resultado en terminal del comando cat. | 7 |
| 4.4. Resultado en terminal del comando pico. | 7 |
| 4.5. Resultado en terminal del comando who. | 7 |
| 4.6. Resultado en terminal del comando trap -l. | 8 |
| 4.7. Resultado en terminal del comando fg. | 8 |
| 4.8. Resultado en terminal del comando nano. | 8 |
| 4.9. Se crea un archivo de texto por el uso del comando nano. | 9 |
| 4.10. Resultado en terminal del comando bash. | 9 |

Índice de tablas

Capítulo 1

Objetivo

Instalar el Sistema Operativo GNU-Linux en una computadora portátil o de escritorio a través de una máquina virtual y/o en una partición física para tener una plataforma de desarrollo en la materia de Cómputo Paralelo en donde se puedan ejecutar comandos de administración y desarrollo de programas en Lenguaje C y otros lenguajes.

Capítulo 2

Marco Teórico

- 2.1. Máquinas virtuales**
- 2.2. Distribuciones de GNU-Linux**
- 2.3. Requisitos de instalación del sistema operativo GNU-Linux**
- 2.4. Estadísticas de uso del sistema operativo GNU-Linux en el mundo**
- 2.5. Escritorios GNOME y KDE**
- 2.6. Entornos Command Line Interface y Graphical User Interface**
- 2.7. Terminal de GNU-Linux**
- 2.8. Tipos de usuario en GNU-Linux**
- 2.9. Rutas relativas y absolutas**
- 2.10. Redireccionamiento**
- 2.11. Clasificación de los comandos en GNU-Linux**
- 2.12. Variables de entorno**
- 2.13. Comandos de GNU-Linux mostrados en la tabla 1**

Capítulo 3

Instalación de GNU-Linux

Nota: Esta fase de la práctica se realiza en equipo, sin embargo, como todos los integrantes realizarán actividades en el sistema operativo, todos deben instalar el sistema operativo y colocar resultados en el archivo del reporte de práctica.

- 3.1. Instalación - Bustillos Cruz Jonatan**
- 3.2. Instalación - Delgado Lucero Cristian Isaac**
- 3.3. Instalación - Frem Cortés José Angel**
- 3.4. Instalación - Luna Gonzales Gabriel Alexis**

Capítulo 4

Entorno de GNU-Linux

Nota: A partir de este momento, todo el trabajo es en equipo. Ejecuta el programa terminal y capture la pantalla. Ejecuta en la terminal los comandos de la Tabla 1 explicando su funcionamiento.

4.1. Comandos - Bustillos Cruz Jonatan

- 4.1.1. Comando ps
- 4.1.2. Comando more
- 4.1.3. Comando time
- 4.1.4. Comando du
- 4.1.5. Comando ps -fea
- 4.1.6. Comando less
- 4.1.7. Comando uname
- 4.1.8. Comando pstree
- 4.1.9. Comando man
- 4.1.10. Comando mkdir

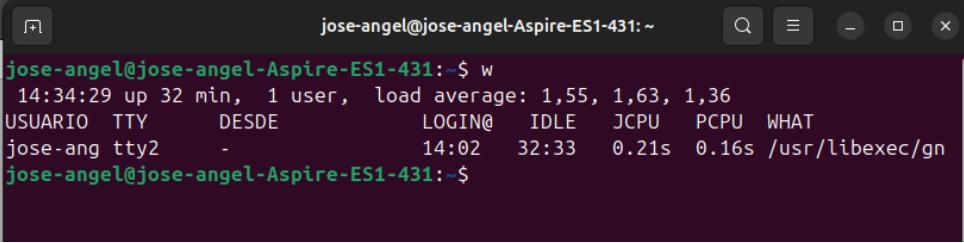
4.2. Comandos - Delgado Lucero Cristian Isaac

- 4.2.1. Comando cal
- 4.2.2. Comando clear
- 4.2.3. Comando apt
- 4.2.4. Comando rm
- 4.2.5. Comando date
- 4.2.6. Comando ifconfig
- 4.2.7. Comando exit
- 4.2.8. Comando mv
- 4.2.9. Comando echo
- 4.2.10. Comando df

4.3. Comandos - Frem Cortés José Angel

4.3.1. Comando w

Este comando proporciona un resumen detallado del estado del sistema y de los usuarios con sesiones activas. Un ejemplo de su uso se puede apreciar en la siguiente imagen:



```
jose-angel@jose-angel-Aspire-ES1-431:~$ w
14:34:29 up 32 min, 1 user, load average: 1,55, 1,63, 1,36
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
jose-ang tty2 - 14:02 32:33 0.21s 0.16s /usr/libexec/gn
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.1: Resultado en terminal del comando w.

4.3.2. Comando kill -l -9

El flag -l sirve para listar todas las señales que puedes enviar. Te muestra una lista numerada (como SIGKILL, SIGTERM, etc.).

El -9 envía la señal SIGKILL, que fuerza el cierre de un proceso de forma inmediata. Se usa cuando un programa se trabó y no quiere cerrar por las buenas.

Un ejemplo de su uso se puede apreciar en la siguiente imagen:



```
jose-angel@jose-angel-Aspire-ES1-431:~$ kill -l -9
bash: kill: (-9) - No existe el proceso
jose-angel@jose-angel-Aspire-ES1-431:~$ kill -l -9
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
 11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGNALRM    15) SIGTERM
 16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
 21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU    25) SIGXFSZ
 26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO      30) SIGPWR
 31) SIGSYS      34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
 38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
 43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
 48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
 53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
 58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
 63) SIGRTMAX-1  64) SIGRTMAX
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.2: Resultado en terminal del comando kill -l -9.

4.3.3. Comando cat

Sirve para ver el contenido de un archivo de texto rápido sin abrir un editor. Un ejemplo de su uso se puede apreciar en la siguiente imagen:



```
jose-angel@jose-angel-Aspire-ES1-431:~$ cat tips.txt
-----
TIPS DE LINUX:
1) Editor de texto: Bloc de notas en Ubuntu.

2) Para abrir la terminal en Ubuntu rápidamente, utiliza el atajo de teclado Ctrl + Alt + T.

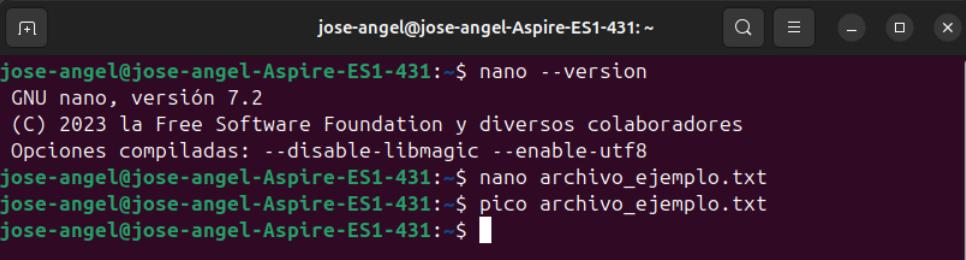
3) Arroba @: Alt Gr + Q.
-----

jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.3: Resultado en terminal del comando cat.

4.3.4. Comando pico

Es un editor de texto súper sencillo (estilo Bloc de Notas). Un ejemplo de su uso se puede apreciar en la siguiente imagen (aunque se recomienda mejor ver el resultado del comando nano):

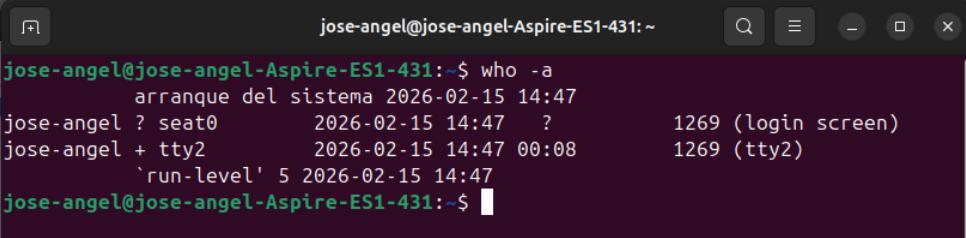


```
jose-angel@jose-angel-Aspire-ES1-431:~$ nano --version
GNU nano, versión 7.2
(C) 2023 la Free Software Foundation y diversos colaboradores
Opciones compiladas: --disable-libmagic --enable-utf8
jose-angel@jose-angel-Aspire-ES1-431:~$ nano archivo_ejemplo.txt
jose-angel@jose-angel-Aspire-ES1-431:~$ pico archivo_ejemplo.txt
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.4: Resultado en terminal del comando pico.

4.3.5. Comando who

Se utiliza para listar exclusivamente a los usuarios que han iniciado sesión en el sistema en tiempo real. Un ejemplo de su uso se puede apreciar en la siguiente imagen:



```
jose-angel@jose-angel-Aspire-ES1-431:~$ who -a
arranque del sistema 2026-02-15 14:47
jose-angel ? seat0 2026-02-15 14:47 ? 1269 (login screen)
jose-angel + tty2 2026-02-15 14:47 00:08 1269 (tty2)
`run-level' 5 2026-02-15 14:47
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.5: Resultado en terminal del comando who.

4.3.6. Comando trap -l

Su funcionamiento lo vi similar al comando de kill. Un ejemplo de su uso se puede apreciar en la siguiente imagen:

```
jose-angel@jose-angel-Aspire-ES1-431:~$ trap -l
 1) SIGHUP      2) SIGINT      3) SIGQUIT      4) SIGILL      5) SIGTRAP
 6) SIGABRT     7) SIGBUS      8) SIGFPE       9) SIGKILL     10) SIGUSR1
11) SIGSEGV     12) SIGUSR2     13) SIGPIPE     14) SIGNALRM    15) SIGTERM
16) SIGSTKFLT   17) SIGCHLD     18) SIGCONT     19) SIGSTOP     20) SIGTSTP
21) SIGTTIN     22) SIGTTOU     23) SIGURG      24) SIGXCPU     25) SIGXFSZ
26) SIGVTALRM   27) SIGPROF     28) SIGWINCH    29) SIGIO       30) SIGPWR
31) SIGSYS     34) SIGRTMIN    35) SIGRTMIN+1  36) SIGRTMIN+2  37) SIGRTMIN+3
38) SIGRTMIN+4  39) SIGRTMIN+5  40) SIGRTMIN+6  41) SIGRTMIN+7  42) SIGRTMIN+8
43) SIGRTMIN+9  44) SIGRTMIN+10 45) SIGRTMIN+11 46) SIGRTMIN+12 47) SIGRTMIN+13
48) SIGRTMIN+14 49) SIGRTMIN+15 50) SIGRTMAX-14 51) SIGRTMAX-13 52) SIGRTMAX-12
53) SIGRTMAX-11 54) SIGRTMAX-10 55) SIGRTMAX-9  56) SIGRTMAX-8  57) SIGRTMAX-7
58) SIGRTMAX-6  59) SIGRTMAX-5  60) SIGRTMAX-4  61) SIGRTMAX-3  62) SIGRTMAX-2
63) SIGRTMAX-1  64) SIGRTMAX
```

Figura 4.6: Resultado en terminal del comando trap -l.

4.3.7. Comando fg

Sirve para traer de vuelta un proceso que se haya pausado con Ctrl + Z en primer plano y seguir trabajando con el. Un ejemplo de su uso se puede apreciar en la siguiente imagen:

```
jose-angel@jose-angel-Aspire-ES1-431:~$ fg
bash: fg: actual: no existe ese trabajo
jose-angel@jose-angel-Aspire-ES1-431:~$ fg %1
bash: fg: %1: no existe ese trabajo
```

Figura 4.7: Resultado en terminal del comando fg.

4.3.8. Comando nano

Es un editor de texto súper sencillo (estilo Bloc de Notas). Es básicamente lo mismo que pico. nano es la versión moderna y mejorada de pico.

Un ejemplo de su uso se puede apreciar en la siguiente imagen:



The screenshot shows a terminal window with the following content:

```
jose-angel@jose-angel-Aspire-ES1-431:~$ nano --version
GNU nano, versión 7.2
(C) 2023 la Free Software Foundation y diversos colaboradores
Opciones compiladas: --disable-libmagic --enable-utf8
jose-angel@jose-angel-Aspire-ES1-431:~$ nano archivo_ejemplo.txt
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.8: Resultado en terminal del comando nano.

Si el archivo existe, se abre y se puede trabajar en el. De lo contrario, se crea como nuevo; por ejemplo, ver la siguiente figura.

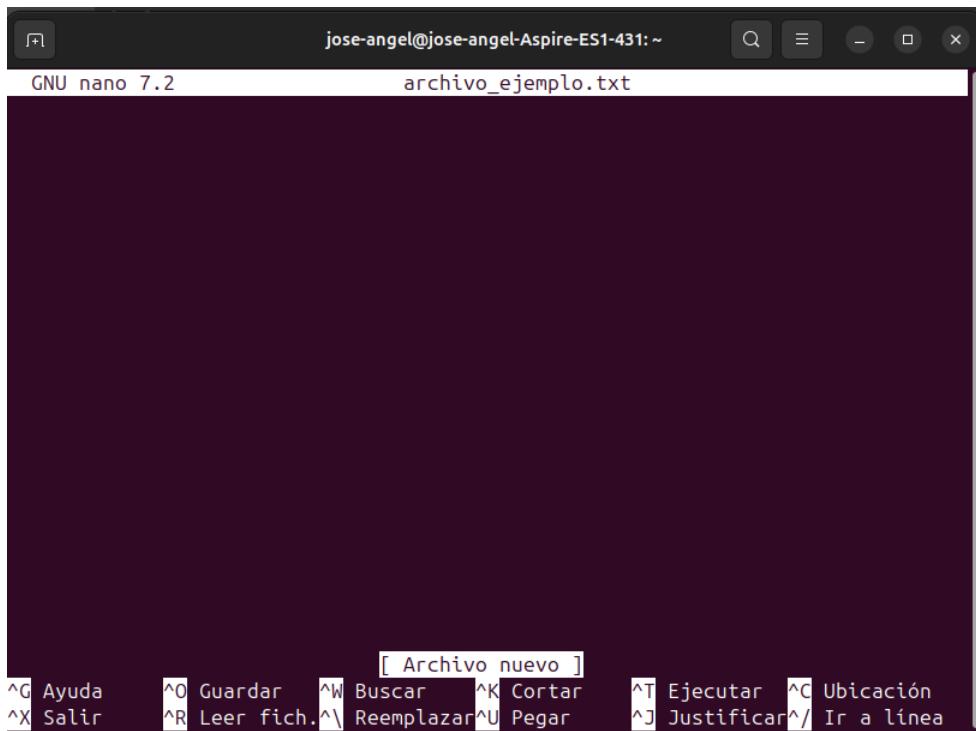


Figura 4.9: Se crea un archivo de texto por el uso del comando nano.

4.3.9. Comando bash

Es el intérprete de comandos por defecto en casi todo Linux. Aunque realmente no vi que hiciera gran cosa este comando.

Un ejemplo de su uso se puede apreciar en la siguiente imagen:

```
jose-angel@jose-angel-Aspire-ES1-431:~$ bash
jose-angel@jose-angel-Aspire-ES1-431:~$ bash nombre_script.sh
bash: nombre_script.sh: No existe el archivo o el directorio
jose-angel@jose-angel-Aspire-ES1-431:~$
```

Figura 4.10: Resultado en terminal del comando bash.

4.4. Comandos - Luna Gonzales Gabriel Alexis

- 4.4.1. Comando pwd
- 4.4.2. Comando cd
- 4.4.3. Comando vi
- 4.4.4. Comando wc
- 4.4.5. Comando su
- 4.4.6. Comando ls
- 4.4.7. Comando apt-get
- 4.4.8. Comando sudo
- 4.4.9. Comando ls -la

Capítulo 5

Ejercicios Prácticos

- 5.1. Rutas absolutas y relativas
- 5.2. Borrado de archivos con comodines
- 5.3. Redireccionamiento con > y »
- 5.4. Instalación de software por línea de comandos
- 5.5. Instalación de software usando el entorno gráfico
- 5.6. Mover y copiar archivos en la jerarquía de directorios
- 5.7. Gestión de usuarios
- 5.8. Hola Mundo en Lenguaje C usando nano
- 5.9. Hola Mundo en Lenguaje C usando gedit

Capítulo 6

Conclusiones

- 6.1. Bustillos Cruz Jonatan**
- 6.2. Delgado Lucero Cristian Isaac**
- 6.3. Frem Cortés José Angel**
- 6.4. Luna Gonzales Gabriel Alexis**