COMANDOS UTILES PARA CMD LINUX EN CODESPACES:

NOTAS:
cuando se menciona archivo se tiene que escribir el nombre del archivo con su extensión de archivo que es , por ejemplo: touch P6.asm , touch archivo.txt, etc.

en le caso de donde dice carpeta , solo se pone el nombre de la carpeta en cuestión por ejemplo: rm -r LIB, etc.

Gestión de Archivos y Directorios

ls # Lista archivos y carpetas

ls -l # Lista con detalles

pwd # Muestra el directorio actual

cd ruta # Cambia de directorio

mkdir nombre_carpeta # Crea una carpeta

touch archivo # Crea archivo vacío

cat archivo # Muestra contenido del archivo

head archivo # Muestra primeras líneas

tail archivo # Muestra últimas líneas

less archivo # Muestra archivo con scroll

stat archivo # Detalles del archivo

rm archivo # Elimina un archivo

rm -r carpeta # Elimina carpeta y su contenido

rm -f archivo # Fuerza eliminación sin confirmar

rm -i archivo # Pide confirmación antes de borrar

cp archivo destino # Copia archivo

cp -r carpeta destino # Copia carpeta

mv archivo destino # Mueve archivo

mv archivo nuevo_nombre # Renombra archivo

🔎 **Búsqueda y Visualización**

find . -name archivo # Busca archivo por nombre

```
grep "palabra" archivo        # Busca palabra dentro de un archivo
wc archivo                    # Cuenta líneas, palabras, caracteres
file archivo                  # Muestra tipo de archivo
diff archivo1 archivo2        # Compara diferencias entre archivos
```

🔒 Permisos y Propiedad

```
chmod 755 archivo             # Cambia permisos de archivo
chown usuario archivo         # Cambia el propietario del archivo
```

📝 Edición de Archivos

```
nano archivo                  # Editor de texto en terminal
```

🗜️ Compresión y Archivos Tar/Gzip

```
tar -cvf archivo.tar carpeta  # Crear archivo .tar
tar -xvf archivo.tar          # Extraer archivo .tar
gzip archivo                  # Comprimir archivo con gzip
gunzip archivo.gz             # Descomprimir gzip
```

⚙ Instalación de Herramientas (APT)

```
sudo apt-get update
sudo apt-get install nasm
sudo apt-get install build-essential
sudo apt-get install nano
sudo apt-get install binutils
sudo apt-get install gdb

sudo apt install libc6-dev-i386   # Librerías de desarrollo 32-bit
sudo apt install make             # Instalador de make
sudo apt install clang            # Compilador alternativo
sudo apt install strace ltrace    # Trazado de llamadas a sistema/lib
```

🧱 Compilación y Ejecución en Ensamblador NASM (x86 y x64)

0. 🧰 Preparación del entorno

sudo apt-get update # Actualiza los repositorios de paquetes

sudo apt-get install nasm # Instala el ensamblador NASM

sudo apt-get update && sudo apt-get install -y nasm **#actualizar paquetes y instalar nasm**

sudo apt-get install gcc-multilib # (opcional) Instala soporte para gcc de 32 bits si se usará en sistemas de 64 bits

#installa paquetes necesarios dependiendo de los archivos en repositorios:

```
sudo apt-get update && \
if ls *.asm &>/dev/null && ! ls *.c &>/dev/null; then
  # Solo ASM
  for pkg in nasm; do
    dpkg -s $pkg &>/dev/null && echo "[OK] $pkg ya está instalado" || sudo apt-get install -y $pkg
  done
elif ls *.asm &>/dev/null && ls *.c &>/dev/null; then
  # ASM + C
  for pkg in nasm gcc-multilib libc6-dev-i386; do
    dpkg -s $pkg &>/dev/null && echo "[OK] $pkg ya está instalado" || sudo apt-get install -y $pkg
  done
else
  echo "[INFO] No se detectaron archivos .asm (ni .c opcionales)."
fi
```

---

1. 🛠️ Ensamblar Código en NASM

nasm -f elf32 programa.asm -o programa.o # Ensamblar en formato ELF 32-bit

nasm -f elf64 programa.asm -o programa.o # Ensamblar en formato ELF 64-bit

nasm -f elf p4.asm # Ensamblar p4.asm (por defecto formato de 32 bits)

nasm -g -F dwarf -f elf32 archivo.asm -o archivo.o # Ensamblar con info de depuración

---

2. 🔗 Enlazar con ld (Linker)

ld -m elf_i386 -s -o programa programa.o # Enlace simple sin bibliotecas

ld -m elf_i386 -o programa programa.o libpc_io.a # Enlace con biblioteca estática

ld -m elf_i386 -o programa programa.o ./LIB/libpc_iox.a ./LIB/pbin.o # Enlace con múltiples bibliotecas

ld -m elf_i386 -dynamic-linker /lib/ld-linux.so.2 -lc -o archivo archivo.o # Enlace con libc (link dinámico)

ld -m elf_i386 -o P4 p4.o ./LIB/libpc_io.a # Enlace con biblioteca estática (caso P4)

ld -m elf_i386 -o P5 P5.o libpc_iox.a # Enlace con biblioteca externa (caso P5)

---

3. 🔧 Enlazar con gcc (para funciones de C o mezcla de ASM + C)

🖥️ En sistemas de 32 bits:

gcc -m32 -nostartfiles programa.o -o programa # Enlace sin runtime estándar

gcc -m32 programa.o -o programa -lc # Enlace con libc estándar

🧩 Ensamblar y compilar ASM + C (Máquina de 32 bits):

nasm -f elf archivoASM.asm # Ensamblar ASM

gcc -c archivoC.c # Compilar C

gcc archivoASM.o archivoC.o -o nombre_ejecutable # Enlace final

🧩 Ensamblar y compilar ASM + C (Máquina de 64 bits):

sudo apt-get install gcc-multilib # Instalar soporte 32 bits (si falta)

sudo apt-get install libc6-dev-i386 # Instala librerías de desarrollo en 32

nasm -f elf archivoASM.asm # Ensamblar

nasm -f elf32 ./P10/P10.asm -o ./P10/P10.o # Ensamblar en carpeta

gcc -m32 -c archivoC.c # Compilar C en modo 32 bits

gcc -m32 archivoASM.o archivoC.o -o nombre_ejecutable # Enlace final

gcc -m32 ./P10/Test_Fun.c ./P10/P10.o -o ./P10/P10 #enlazar archivo c con archivo nasm.o

---

4. ▶ Ejecutar el binario

./programa # Ejecutar programa general

cd p4 && ./P4 # Ejecutar desde su carpeta

./P5 # Ejecutar P5

./P10/P10 #Ejecutar dentro de carpeta

---

5. 🧹 Extras y limpieza

strip archivo # Eliminar símbolos para reducir tamaño

rm archivo.o # Eliminar archivos objeto

rm programa # Eliminar binario generado

rm -f *.o *.out *.bin #Eliniminar binario y objeto a la vez

🔎 Verificación y Análisis del Binario

file programa # Verifica el tipo de archivo

readelf -h programa # Muestra encabezado ELF

ldd programa # Muestra dependencias del ejecutable

objdump -d programa.o # Desensamblar archivo objeto

objdump -d programa # Desensamblar ejecutable

hexdump -C programa # Ver contenido hexadecimal

nm archivo.o # Muestra símbolos del objeto

strings archivo # Extrae texto legible

readelf -s archivo # Tabla de símbolos

readelf -l archivo # Segmentos de carga

objdump -M intel -d archivo # Desensamblado en sintaxis Intel

---

📂 Limpieza de Archivos Generados

```
rm programa.o programa     # Elimina objeto y binario

rm -f *.o *.out *.bin programa   #Elimina archivos objeto y bin y ejecutables)
make clean                  # Si usas Makefile
```

🐞 Depuración con GDB

```
gdb ./programa
(gdb) break _start        # Punto de ruptura en inicio
(gdb) run
(gdb) info registers      # Ver registros
(gdb) stepi               # Ejecuta instrucción a instrucción
(gdb) quit
```

📦 Gestión de Paquetes Python (pip)

```
pip install nombre_del_paquete
pip install --upgrade pip
pip uninstall nombre_del_paquete
pip list
pip install -r requirements.txt

pip freeze > requirements.txt     # Genera lista de paquetes instalados
pip show nombre_paquete           # Muestra detalles del paquete
```

🧠 Otros Comandos Útiles

```
df -h                  # Espacio en disco
free -h                # Uso de memoria RAM
ps aux                 # Lista de procesos
history                # Historial de comandos

history | grep comando     # Filtra historial por palabras clave
history -n                 # Muestra historial de la sesión actual
```

```
!n                    # Ejecuta el comando número n
!comando              # Ejecuta el último comando que empieza con "comando"
history > historial.txt    #Exportar historial a un archivo:
history -c                 # Borrar el historial actual de la sesión
history -c && history -w     #Borrar historial de la sesión y también del archivo
rm ~/.bash_history     #Eliminar directamente el archivo .bash_history
alias lsl="ls -l"          # Crear alias personalizado
top                    # Monitor de procesos interactivo
htop                   # Versión mejorada (si está instalada)
uptime                 # Tiempo encendido del sistema
whoami                 # Muestra el usuario actual
uname -a               # Información del sistema
env                    # Variables de entorno
```

#elimina y crea programas a la vez:

```
rm -f programa1 programa2 programa3 && touch ejemplo.asm ejemplo.o ejemplo.out
```

#compila , enlaza, ejecuta , ensambla y asegura archivo asm :

```
cd P5 && archivo=$(ls *.asm | head -n1 | sed 's/\.asm//') && nasm -f elf "$archivo.asm" &&
ld -m elf_i386 -o "$archivo" "$archivo.o" ../LIB/*.a && ./"$archivo"
```

#comado multiuso de compilación:

```
cd P10 && base=$(ls *.asm | head -n1 | sed 's/\.asm//') && \
nasm -f elf32 "$base.asm" -o "$base.o" && \
if ls *.c &>/dev/null; then \
  gcc -m32 -c *.c && \
  gcc -m32 "$base.o" *.o -o "$base"; \
elif ls ../LIB/*.a &>/dev/null; then \
  ld -m elf_i386 -o "$base" "$base.o" ../LIB/*.a; \
else \
  ld -m elf_i386 -s -o "$base" "$base.o"; \
```

fi && ./"$base"

🧰 **Git y GitHub :**

para respaldar tus cambios en caso de que no pueda (borrar todos commits y crea un nueva rama):

 # Clona tu repositorio y accede a él

git clone https://github.com/uabcingeniero58/LAB-OC-2025

cd LAB-OC-2025

 # Crea una nueva rama sin historial

git checkout --orphan nueva-rama

 # Agrega y haz commit de los cambios

git add .

git commit -m "commit inicial"

 # Reemplaza la rama principal correctamente

git branch -D main

git branch -m nueva-rama main

 # Sube los cambios y sobrescribe historial (corrigiendo el guion largo incorrecto)

git push origin main --force

Github extra:

git config --global user.name "Tu Nombre"

git config --global user.email "correo@example.com"

git status

git log --oneline --graph --all

git stash # Guarda cambios temporales

git stash pop # Recupera cambios

🕹️OCIO:

```
bc                          # Calculadora interactiva (salir con Ctrl+D)
cal                         # Calendario actual
cal 2025                    # Calendario del año 2025
cal -j                      # Calendario juliano
cal 4 2025                  # Calendario de abril de 2025
curl -L ascii.live/forrest  # Run, Forrest, run!
curl -L ascii.live/parrot   # Parrot animacion
curl -L ascii.live/clock    # Clock animation
curl -L ascii.live/can-you-hear-me  # Voice message animation
curl -L ascii.live/donut    # Donut animacion
curl ifconfig.me            # Mostrar IP pública (si tiene acceso a red)
curl wttr.in/Tijuana        # Clima actual para una ciudad
curl -L wttr.in/Moon        # fase lunar actual en ASCII

curl -L git.io/ricex        # imagen estática ascii de arroz
curl -L git.io/unix         # imagen estática ascii de Unix logo
curl -L git.io/taco         # imagen estática ascii de Taco
curl -L git.io/pizzza       # imagen estática ascii de Pizza
curl -L git.io/pancakes     # imagen estática ascii de Pancakes
curl -L git.io/poptart      # imagen estática ascii de Pop-Tart
curl -L git.io/waffles      # imagen estática ascii de Waffles
curl -L git.io/burger       # imagen estática ascii de hamburguesa
curl -L git.io/rice         # imagen estática ascii de arroz (alternative)
curl -L git.io/vburger      # imagen estática ascii de Veggie burger
curl -L git.io/pizzzza      # imagen estática ascii de Another pizza version
curl -L git.io/coffee       # imagen estática ascii de Coffee cup

curl -L http://artscene.textfiles.com/asciiart/phonepig.hum  #funny middle finger
```

curl http://artscene.textfiles.com/asciiart/unicorn # imagen estetica de unicorino

curl -L --output - http://artscene.textfiles.com/asciiart/angela.art # art ascii forbidden

date   # Mostrar hora actual

echo $((RANDOM % 20 + 1))   # Simular dados RPG (d20)

echo $((RANDOM % 6 + 1))   # Tira un dado virtual (1 al 6)

echo $RANDOM   # Generador de números aleatorios

echo "Hola 👋 Mundo 🌍"   # Muestra un saludo animado

echo "Hola mundo" | tr 'A-Z' 'a-z'   # Convertir texto a minúsculas

echo "hola mundo" | tr 'a-z' 'A-Z'   # Convertir texto a mayúsculas

echo "Hola mundo" | wc   # Contar líneas, palabras y caracteres

echo "palabra" | rev   # Imprime palabra al revés en terminal

for i in {1..10}; do echo "5 x $i = $((5*i))"; done   # Imprime una tabla de multiplicar del 5

head /dev/urandom | tr -dc A-Za-z0-9 | head -c 12   # Generar clave aleatoria

man ascii | col -b | less   # Muestra una tabla ASCII

shuf -n1 -e piedra papel tijeras   # Piedra, papel o tijeras aleatorio

shuf -n1 -e sí no   # Simula una ruleta de sí/no

sleep 5 && echo "¡Listo!"   # Cuenta atrás de 5 segundos

telnet towel.blinkenlights.nl   # Ver animación de Star Wars (requiere IPv4)

time read   # Cronómetro básico (Ctrl+C para detener)

uname -a   # Mostrar información del sistema

uptime   # Mostrar cuánto tiempo ha estado encendido el sistema

watch -n 1 date   # Ver calendario y hora en tiempo real

yes hola   # Repite una palabra infinitamente

-m32 -O0 -masm=intel   # solo para compiler explorer (pagina externa) (-O0)

sudo apt install boxes # libreria para Dibuja cajas ASCII alrededor de texto.

echo "Linux divertido" | boxes -d cat   #Dibuja cajas ASCII alrededor de texto.

sudo apt install aview   # libreria imageness en ASCII directamente en la terminal.

asciiview imagen.png # Muestra imágenes en ASCII directamente en la terminal.