

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERIA, ESCUELA DE VACACIONES DE JUNIO

LABORATORIO ARQUITECTURA DE COMPUTADORAS Y ENSAMBLADORES 1

Practica 2

Punto de venta en ensamblador masm x86

Carlos Javier Martínez Polanco

201709282

MANUAL TÉCNICO

1. Herramientas de desarrollo:

- Se utilizó el emulador DOSBox para emular el sistema operativo DOS.
- Se utilizó el gestor Make para compilar y ejecutar el código.
- Se utilizó código ensamblador masm x86 como entorno de desarrollo.
- Se utilizó una computadora con sistema operativo Windows 11 instalado.
- Se utilizó la interrupción DOS número 21 para el manejo de servicios del procesador.
- Se utilizó la directiva Radix 16 para utilizar sistema hexadecimal en la configuración.

2. Variables:

```
clave_capturada      db      09  dup (0)
usuario_capturado    db      08  dup (0)
|_|_|_|_|_|_|_|      db      '$'
espacio_leido         db      00
estado               db      00
buffer_linea          db      0ff dup (0)
tam_liena_leida       db      00
handle_conf           dw      0000
nombre_conf           db      "vacas/pra2/PRA2.CNF", 00
```

Variables para la lectura del archivo de configuración

```

usac      db      "Universidad de San Carlos de Guatemala","$"
curso     db      "Arquitectura de Computadoras y Ensambladores 1","$"
seccion   db      "Seccion N","$"
periodo   db      "Escuela de vacaciones de Junio 2023","$"
nombre    db      "CARLOS JAVIER MARTINEZ POLANCO 201709282","$"
nl         db      0a,"$"
linea_div db      "=====", "$"

```

Variables para el encabezado del proyecto

```

;;MENU PRINCIPAL
titulo_menu db "          MENU DEL SISTEMA", "$"
;;PRODUCTOS
titulo_productos db "(1) OPCIONES DE PRODUCTOS", "$"
subtitulo_productos1 db "(1) INGRESAR PRODUCTOS", "$"
ingresar_prod db "HAS SELECCIONADO INGRESAR PRODUCTO", "$"
subtitulo_productos2 db "(2) ELIMINAR PRODUCTOS", "$"
eliminar_prod db "HAS SELECCIONADO ELIMINAR PRODUCTO", "$"
subtitulo_productos3 db "(3) VER PRODUCTOS", "$"
ver_prod db "HAS SELECCIONADO VER PRODUCTOS", "$"
subtitulo_atras2 db "(2) ATRAS", "$"
subtitulo_atras4 db "(4) ATRAS", "$"
subtitulo_atras5 db "(5) ATRAS", "$"
portada_productos db "          MENU PRODUCTOS", "$"
portada_ventas db "          MENU VENTAS", "$"
portada_herramientas db "          MENU HERRAMIENTAS", "$"
;;VENTAS
titulo_ventas db "(2) OPCIONES DE VENTA", "$"
subtitulo_ventas1 db "(1) INGRESA LOS DATOS DE LA VENTA", "$"
datos_venta db "DATOS VENTA", "$"
;;HERRAMIENTAS
titulo_herramientas db "(3) OPCIONES DE HERRAMIENTAS DEL SISTEMA", "$"
subtitulo_herramientas1 db "(1) GENERAR CATALOGO", "$"
subtitulo_herramientas2 db "(2) REPORTE PRODUCTOS", "$"
subtitulo_herramientas3 db "(3) REPORTE VENTAS", "$"
subtitulo_herramientas4 db "(4) REPORTE PRODUCTOS SIN EXISTENCIA", "$"
:~SALIR

```

Variables para la funcionalidad del menú

```

;;SALIR
titulo_salir db "(4) SALIR DEL SISTEMA", "$"

;;REPORTE CATALOGO
reporte1 db "SE HA GENERADO EL CATALOGO EXITOSAMENTE", "$"
;;REPORTE PRODUCTOS
reporte2 db "SE HA GENERADO EL REPORTE DE PRODUCTOS EXITOSAMENTE", "$"
;;REPORTE VENTAS
reporte3 db "SE HA GENERADO EL REPORTE DE VENTAS EXITOSAMENTE", "$"
;;REPORTE PRODUCTOS SIN EXISTENCIA
reporte4 db "SE HA GENERADO EL REPORTE DE PRODUCTOS SIN EXISTENCIA EXITOSAMENTE", "$"
;;SOLICITAR INGRESO DE DATOS MENU
ingreso_menu db "INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: ", "$"

;;INGRESO DE DATOS INCORRECTO
ingreso_menu_error db "OPCION DEL MENU INGRESADA ERRONEA ", "$"

;;MENSAJE PRESIONAR ENTER
msg_enter db "-> Presione Enter para continuar...", "$"

;;NOMBRE DEL ARCHIVO DE LAS CREDENCIALES
archivo_credenciales db "vacas/pra2/PRA2.CNF", 00

;;BUFFER          20 = 32bytes  tam_cadena = 00
buffer_entrada  db 20, 00
                db 20 dup (0)

;;MANEJADOR O PUNTERO DEL ARCHIVO
handle_creds dw 0000

;;ARCHIVO CREDENCIALES ERROR
error_archivo_credenciales db "ERROR, EL ARCHIVO DE CREDENCIALES NO EXISTE", "$"

;;ARCHIVO PROD ERROR
crear_archivo_prod db "SE HA CREADO EL ARCHIVO PROD.BIN EXITOSAMENTE", "$"
error_crear_archivo_prod db "ERROR, EL ARCHIVO NO HA SIDO CREADO", "$"

```

Variables para mensajes de éxito, error un buffer de entrada para capturar texto

Y el manejador del archivo de configuración.

```

;;ESTRUCTURA DATOS CREDENCIALES
usu db "cpolanco" ;;8 bytes
cla db "201709282" ;; 9bytes

;;Estructura para leer credenciales
|      ;;tamaño en hex, valor
encabezado_cred db 0e, "[credenciales]" ;;14 bytes contando caracteres especiales
usuario_cred db 07, "usuario" ;;7 bytes contando caracteres especiales
clave_cred db 05, "clave" ;;5 bytes contando caracteres especiales
igual_cred db 01, "="
comillas_cred db 01, "'"

success_login db "CREDENCIALES CORRECTAS", "$"

error_usu db "EL USUARIO INGRESADO ES INCORRECTO", "$"
error_clave db "LA CLAVE INGRESADA ES INCORRECTA", "$"

validate_user db "VALIDANDO USUARIO", "$"
validate_key db "VALIDANDO CLAVE", "$"

```

Variables para el reconocimiento de credenciales en el archivo de configuración

```

;;SOLICITAR DATOS DEL PRODUCTO
prompt_cod db "INGRESA EL CODIGO DEL PRODUCTO: ", "$"
prompt_des db "INGRESA LA DESCRIPCION DEL PRODUCTO: ", "$"
prompt_precio db "INGRESA EL PRECIO DEL PRODUCTO: ", "$"
prompt_unidades db "INGRESA LAS UNIDADES DISPONIBLES DEL PRODUCTO: ", "$"
error_prompt_cod db "ERROR AL INGRESAR EL CODIGO DEL PRODUCTO", "$"
error_prompt_des db "ERROR AL INGRESAR LA DESCRIPCION DEL PRODUCTO", "$"
error_prompt_precio db "ERROR AL INGRESAR EL PRECIO DEL PRODUCTO", "$"
error_prompt_unidades db "ERROR AL INGRESAR LAS UNIDADES DEL PRODUCTO", "$"

;; "ESTRUCTURA PRODUCTO" tamaño total de 40 bytes 28H
cod_prod db 05 dup (0) ;;tamaño 4 bytes
des_prod db 21 dup (0) ;;tamaño 32 bytes
price_prod db 05 dup (0) ;;tamaño 2 bytes
units_prod db 05 dup (0) ;;tamaño 2 bytes
| | | db '$'
;; archivo productos MANEJADOR Y NOMBRE
archivo_prods db "vacas/pra2/PROD.BIN",00
handle_prods dw 0000

;;archivo catalg.htm MANEJADOR Y NOMBRE
;;
nombre_rep1 db "vacas/pra2/CATALG.HTM",00
handle_reps dw 0000

;;buffer del numero para la conversion
numero db 05 dup (30)
dia db ?
mes db ?
anio dw ?
cadenaFecha db 11 dup('$') ; Variable para almacenar la fecha en formato de cadena

```

Estructura de productos para la lectura de productos, manejadores para los archivos de productos y el reporte de catálogo.

```

;;; temps
cod_prod_temp    db    05 dup (0)
puntero_temp     dw    0000
ceros            db    2b dup (0)

;;;DATOS PARA EL REPORTE HTM
tam_encabezado_html db    0c
encabezado_html    db    "<html><body>"
tam_inicializacion_tabla db    3e
inicializacion_tabla db    '<table border="1"><tr><td>codigo</td><td>descripcion</td></tr>'
tam_cierre_tabla   db    8
cierre_tabla       db    "</table>"
tam_footer_html    db    0e
footer_html        db    "</body></html>"
td_html            db    "<td>"
tdc_html           db    "</td>"
tr_html            db    "<tr>"
trc_html           db    "</tr>"
p_html             db    "<p>"
pc_html            db    "</p>"
tam_abrir_p         db    3
tam_cierre_p        db    4

;;;ESTRUCTURA PARA LA FECHA DE LOS REPORTEES
rep_tit db "REPORTE GENERADO LA FECHA: ";;28 bytes
rep_day db 2 dup(0);; 2
rep_sep1 db "/" ;;1
rep_mmont db 2 dup(0);;2
rep_sep2 db "/";;1
rep_year db 2 dup(0);;2
rep_sep3 db " HORA: ";;8
rep_hora db 2 dup(0);;2
rep_sep4 db ":";;1
rep_minu db 2 dup(0);;2

rep_aux_fecha dw 0000

```

Estructura para generar el reporte de catálogo, el cual crea una tabla HTML

Estructura para obtener la fecha y hora

3. Procesos

- MAIN:

Configuración del segmento de datos: Mueve el valor de la etiqueta "@data" al registro AX y luego lo asigna al registro DS. Esta configuración es necesaria para acceder a los datos definidos en el programa.

Imprime una nueva línea en la pantalla.

Imprime un mensaje almacenado en la etiqueta "usac" que contiene datos personales.

Imprime una nueva línea.

Imprime un mensaje almacenado en la etiqueta "curso".

Imprime una nueva línea.

Imprime un mensaje almacenado en la etiqueta "seccion".

Imprime una nueva línea.

Imprime un mensaje almacenado en la etiqueta "periodo".

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime un mensaje almacenado en la etiqueta "nombre".

Imprime una nueva línea.

Imprime una línea divisoria almacenada en la etiqueta "linea_div".

Imprime una nueva línea.

Salto a "menu_principal": Al final del proc "main", se realiza un salto a la etiqueta "menu_principal" para continuar con la ejecución del programa desde ese punto.

- MENU_PRINCIPAL:

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "titulo_menu" que corresponde al título del menú principal.

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "titulo_productos" que corresponde al título de la sección de productos del menú.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "titulo_ventas" que corresponde al título de la sección de ventas del menú.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "titulo_herramientas" que corresponde al título de la sección de herramientas del menú.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "titulo_salir" que corresponde al título de la opción de salir del menú.

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "ingreso_menu" que solicita al usuario ingresar una opción del menú.

Ingreso por teclado:

Espera la entrada del usuario a través de la función de interrupción 21h, servicio 01h.

Compara el valor ingresado (AL) con diferentes opciones numéricas:

Si es igual a 31, salta a la etiqueta "menu_productos".

Si es igual a 32, salta a la etiqueta "menu_venta".

Si es igual a 33, salta a la etiqueta "menu_herramientas".

Si es igual a 34, salta a la etiqueta "fin".

Impresión de mensaje de error: Si el valor ingresado no coincide con ninguna opción válida, se imprime un mensaje de error y se realiza un salto de vuelta al proc "menu_principal".

- MENU_PRODUCTOS:

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "portada_productos" que corresponde al título de la sección de productos.

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "subtitulo_productos1" que corresponde a la opción de ingresar un producto.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_productos2" que corresponde a la opción de eliminar un producto.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_productos3" que corresponde a la opción de ver los productos.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_atras4" que corresponde a la opción de regresar al menú principal.

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "ingreso_menu" que solicita al usuario ingresar una opción del menú.

Ingreso por teclado:

Espera la entrada del usuario a través de la función de interrupción 21h, servicio 01h.

Compara el valor ingresado (AL) con diferentes opciones numéricas:

Si es igual a 31, salta a la etiqueta "ingresar_producto".

Si es igual a 32, salta a la etiqueta "eliminar_producto".

Si es igual a 33, salta a la etiqueta "ver_productos".

Si es igual a 34, salta a la etiqueta "menu_principal".

Impresión de mensaje de error: Si el valor ingresado no coincide con ninguna opción válida, se imprime un mensaje de error y se realiza un salto de vuelta al proc "menu_productos".

- MENU_VENTA:

Impresión de mensajes:

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "portada_ventas" que corresponde al título de la sección de ventas.

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "subtitulo_ventas1" que corresponde a la opción de realizar una venta.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_atras2" que corresponde a la opción de regresar al menú principal.

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "ingreso_menu" que solicita al usuario ingresar una opción del menú.

Ingreso por teclado:

Espera la entrada del usuario a través de la función de interrupción 21h, servicio 01h.

Compara el valor ingresado (AL) con diferentes opciones numéricas:

Si es igual a 31, salta a la etiqueta "realizar_venta".

Si es igual a 32, salta a la etiqueta "menu_principal".

Impresión de mensaje de error: Si el valor ingresado no coincide con ninguna opción válida, se imprime un mensaje de error y se realiza un salto de vuelta al proc "menu_venta".

- MENU_HERRAMIENTAS

Impresión de mensajes:

Imprime dos nuevas líneas.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "portada_herramientas" que corresponde al título de la sección de herramientas.

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "subtitulo_herramientas1" que corresponde a la opción de generar un catálogo.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_herramientas2" que corresponde a la opción de generar un reporte de productos.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_herramientas3" que corresponde a la opción de generar un reporte de ventas.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_herramientas4" que corresponde a la opción de generar un reporte de productos sin existencia.

Imprime una nueva línea.

Imprime el mensaje almacenado en la etiqueta "subtitulo_atras5" que corresponde a la opción de regresar al menú principal.

Imprime una nueva línea.

Imprime dos nuevas líneas.

Imprime el mensaje almacenado en la etiqueta "ingreso_menu" que solicita al usuario ingresar una opción del menú.

Ingreso por teclado:

Espera la entrada del usuario a través de la función de interrupción 21h, servicio 01h.

Compara el valor ingresado (AL) con diferentes opciones numéricas:

Si es igual a 31, salta a la etiqueta "reporte_catalogo".

Si es igual a 32, salta a la etiqueta "reporte_productos".

Si es igual a 33, salta a la etiqueta "reporte_ventas".

Si es igual a 34, salta a la etiqueta "reporte_productos_null".

Si es igual a 35, salta a la etiqueta "menu_principal".

Impresión de mensaje de error: Si el valor ingresado no coincide con ninguna opción válida, se imprime un mensaje de error y se realiza un salto de vuelta al proc "menu_herramientas".

- **INGRESAR_PRODUCTO:**

El programa comienza imprimiendo un título y una nueva línea en la pantalla.

Luego, intenta abrir un archivo llamado "archivo_prods" utilizando la interrupción 21h del sistema operativo. Si el archivo no existe, pasa a la etiqueta "crear_archivo_producto" para crearlo.

Si el archivo se abre correctamente o se crea exitosamente, se guarda el identificador del archivo en la variable "handle_prods".

A continuación, el programa solicita al usuario ingresar el código del producto. Muestra un mensaje de prompt y utiliza la función de entrada del sistema operativo para permitir que el usuario ingrese el código.

Después de recibir la entrada del usuario, se valida el tamaño de la entrada para asegurarse de que no esté vacía.

Si la entrada del código es válida, el programa continúa ejecutando otras funciones relacionadas con la gestión de productos.

- **ELIMINAR_PRODUCTOS:**

Imprime una nueva línea en la pantalla.

Muestra un título relacionado con la eliminación de productos.

Imprime otra nueva línea en la pantalla.

Luego, solicita al usuario ingresar un código de producto.

Verifica que la entrada no esté vacía y tenga un tamaño adecuado.

Si el tamaño es válido, copia el código ingresado en una estructura temporal.

Abre el archivo "archivo_prods" en modo lectura/escritura.

Inicia un ciclo para buscar el código del producto dentro del archivo.

Si encuentra el código, marca la ubicación para eliminar el producto.

Escribe ceros en el espacio del producto eliminado.

Finaliza la eliminación y cierra el archivo.

Salta a la etiqueta "menu_productos" para regresar al menú principal.

- VER_PRODUCTOS:

Imprime una nueva línea en la pantalla.

Muestra un título relacionado con la visualización de productos.

Imprime dos nuevas líneas en la pantalla.

Abre el archivo "archivo_prods" en modo lectura.

Si hay un error al abrir el archivo, muestra un mensaje de error y salta a la etiqueta "fin".

Almacena el identificador del archivo en la variable "handle_prods".

Inicia un ciclo para leer los datos de los productos en el archivo.

Lee el código del producto y el número de precio.

Si se encuentra el final del archivo, salta a la etiqueta "fin_mostrar".

Verifica si el código del producto es válido.

Llama a una función llamada "imprimir_estructura" para mostrar los datos del producto.

Salta de vuelta al ciclo para leer el siguiente producto.

Finaliza el ciclo y salta a la etiqueta "menu_productos".

Si hay un error al leer el archivo, muestra un mensaje de error y salta a la etiqueta "fin".

- REPORTE_CATALOGO:

Elimina el archivo "nombre_rep1" si existe.

Crea un nuevo archivo con el nombre "nombre_rep1".

Almacena el identificador del archivo en la variable "handle_reps".

Escribe el encabezado HTML en el archivo.

Escribe la inicialización de la tabla HTML en el archivo.

Abre el archivo "archivo_prods" en modo lectura.

Si hay un error al abrir el archivo, se muestra un mensaje de error y salta a la etiqueta "fin_mostrar_rep1".

Almacena el identificador del archivo en la variable "handle_prods".

Inicia un ciclo para leer los datos de los productos en el archivo.

Lee el código del producto y el número de precio.

Si se lee un total de 0 bytes, se ha alcanzado el final del archivo y salta a la etiqueta "fin_mostrar_rep1".

Verifica si el código del producto es válido.

Llama a una función llamada "imprimir_estructura_html" para mostrar los datos del producto en formato HTML.

Salta de vuelta al ciclo para leer el siguiente producto.

Finaliza el ciclo y realiza las siguientes acciones:

Escribe el cierre de la tabla HTML en el archivo.

Obtiene la fecha y hora actual del sistema.

Convierte la fecha y hora en cadenas de caracteres.

Escribe el título del reporte en el archivo.

Escribe el cierre de la etiqueta <p> en el archivo.

Escribe el footer HTML en el archivo.

Cierra el archivo.

Muestra un mensaje de título relacionado con el reporte de catálogo.

Salta de vuelta al menú de herramientas.

- IMPRIMIR_ESTRUCTURA:

Mueve el offset de la variable "cod_prod" al registro DI, que se utilizará como puntero para recorrer la cadena.

Inicia un ciclo para buscar el final de la cadena "cod_prod".

Compara el valor actual en [DI] con 0 (carácter nulo) para determinar si se ha llegado al final de la cadena.

Si se ha llegado al final de la cadena, salta a la etiqueta "poner_dolar_1".

Incrementa el puntero DI y vuelve al ciclo para seguir buscando el final de la cadena.

Etiqueta "poner_dolar_1":

Mueve el valor ASCII del símbolo de dólar (\$) al registro AL.

Guarda el valor del registro AL en [DI], reemplazando el carácter nulo por el símbolo de dólar.

Imprime el contenido de la cadena "cod_prod" en pantalla utilizando la función DOS con el servicio AH=09h.

Imprime una nueva línea utilizando la función DOS con el servicio AH=09h.

Imprime el contenido de la cadena "des_prod" en pantalla.

Imprime una nueva línea.

Imprime otra nueva línea.

Retorna de la función.

- IMPRIMIR ESTRUCTURA_HTML

Mueve el valor del handle del archivo de reporte al registro BX.

Escribe las etiquetas HTML <tr> y <td> en el archivo utilizando la función DOS con el servicio AH=40h y los parámetros adecuados.

Mueve el offset de la cadena "cod_prod" al registro DX y establece el valor de SI en 0 para contar los caracteres escritos.

Inicia un ciclo para escribir el código del producto en el archivo.

Mueve el offset de la cadena "cod_prod" al registro DI para obtener el carácter actual.

Mueve el valor del carácter al registro AL.

Compara el valor actual en AL con 0 (carácter nulo) para determinar si se ha llegado al final de la cadena.

Si se ha llegado al final de la cadena o se han escrito 6 caracteres, salta a la etiqueta "escribir_desc".

Mueve el valor 1 al registro CX para indicar que se escribirá un solo carácter.

Escribe el carácter actual en el archivo utilizando la función DOS con el servicio AH=40h.

Incrementa el offset de la cadena "cod_prod" y el valor de SI.

Vuelve al ciclo para seguir escribiendo el código del producto.

Etiqueta "escribir_desc":

Escribe las etiquetas HTML </td> y <td> en el archivo.

Mueve el offset de la cadena "des_prod" al registro DX y establece el valor de SI en 0 para contar los caracteres escritos.

Inicia un ciclo para escribir la descripción del producto en el archivo.

Mueve el offset de la cadena "des_prod" al registro DI para obtener el carácter actual.

Mueve el valor del carácter al registro AL.

Compara el valor actual en AL con 0 (carácter nulo) para determinar si se ha llegado al final de la cadena.

Si se ha llegado al final de la cadena o se han escrito 38 caracteres, salta a la etiqueta "cerrar_tags".

Mueve el valor 1 al registro CX para indicar que se escribirá un solo carácter.

Escribe el carácter actual en el archivo utilizando la función DOS con el servicio AH=40h.

Incrementa el offset de la cadena "des_prod" y el valor de SI.

Vuelve al ciclo para seguir escribiendo la descripción del producto.

Etiqueta "cerrar_tags":

Escribe las etiquetas HTML `</td>` y `</tr>` en el archivo.

Retorna de la función.

- ESPERAR_ENTER:

Imprime una nueva línea en la pantalla utilizando la función DOS con el servicio AH=09h y el offset de la cadena "nl" en DX. Esto genera un salto de línea.

Imprime el mensaje "Presione Enter para continuar" en la pantalla utilizando la función DOS con el servicio AH=09h y el offset de la cadena "msg_enter" en DX.

Imprime otra nueva línea en la pantalla para separar el mensaje de la entrada del usuario.

Lee un carácter desde el teclado utilizando la función DOS con el servicio AH=08h. El carácter leído se guarda en el registro AL.

Compara el valor leído en AL con 0Dh (código ASCII para la tecla Enter).

Si los valores no son iguales, salta de vuelta al paso 4 para leer otro carácter desde el teclado.

Si los valores son iguales, retorna de la función.

- CADENA_A_NUM:

Inicializa el registro AX con el valor 0000, que será utilizado para almacenar el resultado final.

Inicializa el registro CX con el valor 0005, que se utiliza como contador para controlar el bucle de conversión.

Entra en un bucle etiquetado como "seguir_convirtiendo" que se repetirá 5 veces.

Lee el valor de un carácter de la cadena, apuntado por el registro DI.

Compara el valor leído con 00h para verificar si se ha llegado al final de la cadena. Si es igual, se salta a la etiqueta "retorno_cadenaAnum".

Resta 30h al valor leído para obtener el valor numérico del carácter. El resultado se guarda en el registro BL.

Multiplica el contenido de AX por 10 (utilizando la instrucción MUL) y guarda el resultado en DX:AX.

Limpia el registro BH para evitar que afecte al resultado final.

Suma el contenido de BX al registro AX para agregar el valor numérico del carácter convertido.

Incrementa el registro DI para apuntar al siguiente carácter en la cadena.

Decrementa el registro CX en 1 para contar la iteración actual del bucle.

Si el contador CX no ha llegado a cero, salta de nuevo a la etiqueta "seguir_convirtiendo" para continuar la conversión con el siguiente carácter.

Si el contador CX ha llegado a cero, se llega a la etiqueta "retorno_cadenaAnum" y la subrutina retorna.

- NUM_A_CADENA:

Inicializa el registro CX con el valor 0005, que representa la longitud de la cadena resultante.

Inicializa el registro DI con la dirección de memoria de la cadena "numero", donde se almacenará el resultado.

Entra en un bucle etiquetado como "ciclo_poner30s" que se repetirá 5 veces.

Mueve el valor 30h (carácter '0') al byte apuntado por DI, que inicializa todos los caracteres de la cadena con '0'.

Incrementa DI para apuntar al siguiente byte en la cadena.

Decrementa el contador CX en 1 para contar la iteración actual del bucle.

Si el contador CX no ha llegado a cero, salta de nuevo a la etiqueta "ciclo_poner30s" para continuar poniendo '0' en los bytes restantes de la cadena.

Después del bucle, tenemos una cadena de caracteres inicializada con "00000".

Vuelve a inicializar el contador CX con el valor almacenado en AX, que representa el número a convertir.

Mueve DI a la última posición de la cadena, apuntando al byte más significativo.

Entra en un bucle etiquetado como "ciclo_convertirAcadena" que se repetirá CX veces.

Lee el valor de un byte de la cadena apuntado por DI.

Incrementa el valor del byte en 1 para convertirlo al siguiente dígito numérico.

Mueve el valor modificado de vuelta al byte de la cadena.

Compara el valor del byte con 3Ah (carácter ':'). Si son iguales, salta a la etiqueta "aumentar_siguiente_digito_primera_vez".

Decrementa DI para apuntar al siguiente byte en la cadena.

Decrementa el contador CX en 1 para contar la iteración actual del bucle.

Si el contador CX no ha llegado a cero, salta de nuevo a la etiqueta "ciclo_convertirAcadena" para continuar la conversión con el siguiente dígito.

Si el contador CX ha llegado a cero, se llega a la etiqueta "retorno_convertirAcadena" y la subrutina retorna.

- CADENAS_IGUALES:

Entra en un bucle etiquetado como "ciclo_cadenas_iguales" que se repetirá hasta que se cumpla una de las siguientes condiciones:

- a. Se encuentra un byte en la cadena apuntada por SI que es diferente al byte correspondiente en la cadena apuntada por DI.
- b. Se ha recorrido toda la longitud de las cadenas sin encontrar diferencias.

En cada iteración del bucle:

- a. Mueve el valor de un byte de la cadena apuntada por SI al registro AL.
- b. Compara el byte en la cadena apuntada por DI con el valor en AL.
- c. Si los bytes son diferentes, salta a la etiqueta "no_son_iguales".
- d. Incrementa DI y SI para apuntar al siguiente byte en las cadenas.
- e. Decrementa el contador CX en 1 para contar la iteración actual del bucle.

Si se ha llegado al final del bucle sin encontrar diferencias entre las cadenas, se llega a la etiqueta ";;;;;" y se mueve el valor 0xFF al registro DL como indicador de que las cadenas son iguales.

Si se encontró una diferencia entre las cadenas, se llega a la etiqueta "no_son_iguales" y se mueve el valor 0x00 al registro DL como indicador de que las cadenas no son iguales.

La subrutina retorna, y el valor en DL indica si las cadenas son iguales (0xFF) o no (0x00).

- MEMSET:

Entra en un bucle etiquetado como "ciclo_memset" que se repetirá hasta que se haya llenado toda la longitud de la cadena.

En cada iteración del bucle:

- a. Mueve el valor 0x00 al registro AL.
- b. Mueve el valor en AL al byte apuntado por DI en la cadena.
- c. Incrementa DI para apuntar al siguiente byte en la cadena.
- d. Decrementa el contador CX en 1 para contar la iteración actual del bucle.

Una vez que se ha llenado toda la cadena con el valor 0x00, la subrutina retorna.

- VALIDAR_ACCESO:

Abre el archivo de configuración especificado en "nombre_conf" utilizando la interrupción 21h, función 3Dh.

Guarda el identificador del archivo en [handle_conf].

Entra en un bucle etiquetado como "ciclo_lineaXlinea" que se repetirá hasta que se haya analizado todo el archivo de configuración.

En cada iteración del bucle:

- a. Inicializa el buffer de línea "buffer_linea" y el contador de tamaño de línea "tam_liena_leida".
- b. Utiliza la interrupción 21h, función 3Fh, para leer una línea del archivo.
- c. Si la línea está vacía (CX=0), se sale del bucle y se finaliza el análisis del archivo.
- d. Verifica si la cadena actual corresponde al estado "credenciales". Si es así, se ejecuta el código para verificar las credenciales.
- e. Verifica si la cadena actual corresponde al estado "usuario" o "clave". Si es así, se ejecuta el código para obtener el valor correspondiente.
- f. Si ninguna de las condiciones anteriores se cumple, se salta a la etiqueta "retorno_exitoso" indicando un acceso exitoso.

El código para verificar las credenciales:

- a. Compara la cadena en "buffer_linea" con el encabezado de las credenciales en "encabezado_cred" utilizando la subrutina "cadenas_iguales".
- b. Si las cadenas son iguales (DL=0xFF), se salta a la etiqueta "si_hay_creds".
- c. Si las cadenas no son iguales, se salta a la etiqueta "retorno_fallido" indicando un acceso fallido.

El código para obtener el valor de usuario o clave:

- a. Compara la cadena en "buffer_linea" con el campo correspondiente (usuario o clave) utilizando la subrutina "cadenas_iguales".
- b. Si las cadenas son iguales (DL=0xFF), se salta a la etiqueta correspondiente para obtener el valor.
- c. Si las cadenas no son iguales, se salta a la etiqueta "retorno_fallido" indicando un acceso fallido.

Para obtener el valor de usuario:

- a. Se busca el carácter igual ('=') en la línea para ubicar el comienzo del valor de usuario.
- b. Se compara la cadena desde ese punto con el campo "igual_cred" utilizando la subrutina "cadenas_iguales".

- c. Si las cadenas son iguales (DL=0xFF), se captura el valor de usuario y se salta a la etiqueta "ciclo_lineaXlinea".
- d. Si las cadenas no son iguales, se salta a la etiqueta "retorno_fallido" indicando un acceso fallido.

Para obtener el valor de clave:

- a. Se busca el carácter igual ('=') en la línea para ubicar el comienzo del valor de clave.
- b. Se compara la cadena desde ese punto con el campo "igual_cred" utilizando la subrutina "cadenas_iguales".
- c. Si las cadenas son iguales (DL=0xFF), se captura el valor de clave y se salta a la etiqueta "ciclo_lineaXlinea".
- d. Si las cadenas no son iguales, se salta a la etiqueta "retorno_fallido" indicando un acceso fallido.

Si se ha llegado al final del bucle sin encontrar diferencias en las cadenas de acceso, se llega a la etiqueta "retorno_exitoso" indicando un acceso exitoso.

Las etiquetas "retorno_fallido" y "retorno_exitoso" establecen el valor del registro DL según el resultado de la validación y retornan de la subrutina.

- FIN:

Finaliza la ejecución del sistema.

4. MACROS

- CONVERTIR_NUMERO_CADENA:

Se definen las etiquetas locales para uso interno dentro del macro: L_DIV, L_ALM, L_CERO y L_FIN.

Se inicializan los registros necesarios para realizar la conversión:

El valor numérico se carga en el registro AX.

La dirección de memoria de la cadena de destino se carga en el registro SI.

Los registros DX, CX y BX se inicializan con valores de 0 y 10, que representa la base decimal.

Se verifica si el valor numérico (AX) es igual a 0. En caso afirmativo, se salta a la etiqueta L_CERO para tratar el caso especial de valor cero.

Etiqueta L_DIV: Se inicia un bucle para dividir sucesivamente el valor numérico (AX) por la base (10) y almacenar los residuos en la pila para su posterior procesamiento.

Dentro del bucle L_DIV:

Se verifica si el valor numérico (AX) es igual a 0. En caso afirmativo, se salta a la etiqueta L_ALM para comenzar a reconstruir la cadena de caracteres.

Se divide el valor numérico (AX) por la base (10) utilizando la instrucción "div BX". El residuo se almacena en el registro DX.

Se empuja el residuo (DX) en la pila para su posterior procesamiento.

Se incrementa el contador (CX) para realizar un seguimiento del número de dígitos convertidos.

Se borra el registro DX para prepararlo para la siguiente división.

Etiqueta L_ALM: Se ejecuta después de que el valor numérico se haya convertido en residuos almacenados en la pila. Aquí se reconstruye la cadena de caracteres a partir de los residuos almacenados en la pila.

Dentro de la etiqueta L_ALM:

Se saca el residuo (DX) de la pila.

Se agrega el valor 30H (equivalente al carácter '0' en ASCII) al residuo (DX) para convertirlo en el dígito ASCII correspondiente.

El dígito ASCII convertido se almacena en la dirección de memoria apuntada por SI.

Se incrementa SI para apuntar a la siguiente posición en la cadena de caracteres.

Se decrementa el contador (CX) para realizar un seguimiento de los dígitos restantes por convertir.

Se repite este proceso hasta que todos los dígitos hayan sido convertidos.

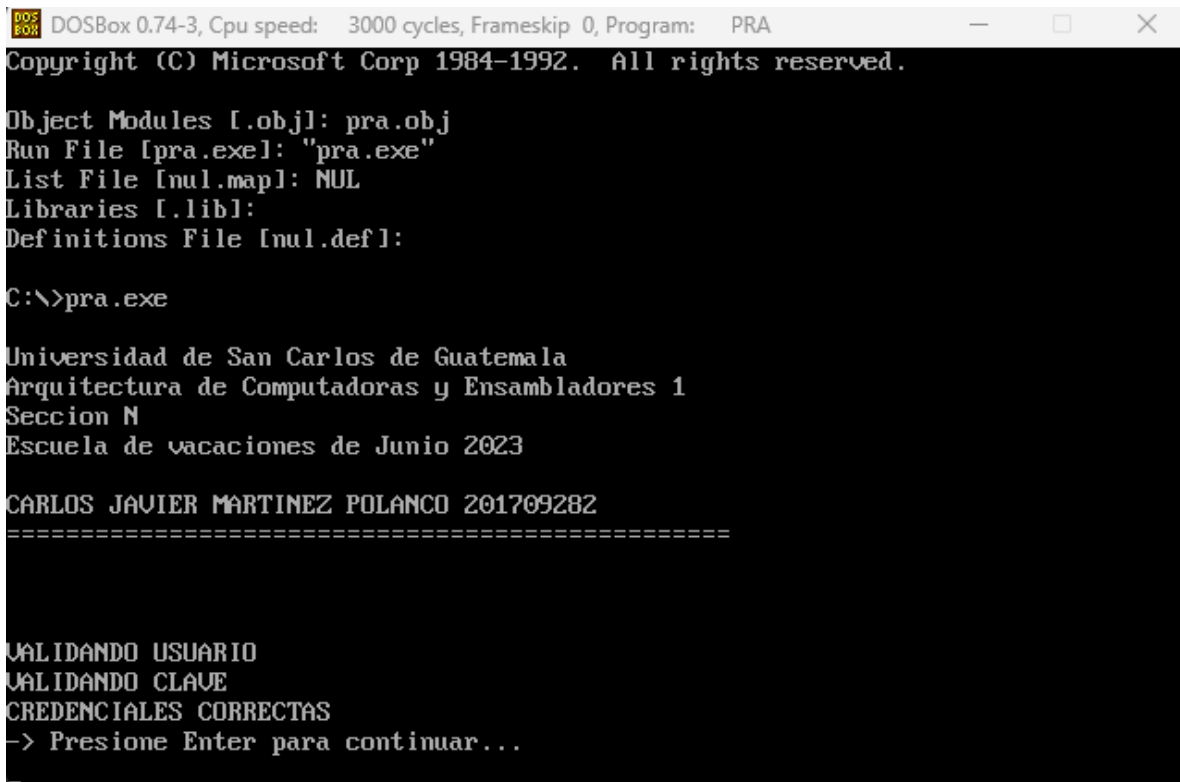
Después de completar la reconstrucción de la cadena de caracteres, se salta a la etiqueta L_FIN para finalizar el proceso de conversión.

Etiqueta L_CERO: Se trata el caso especial en el que el valor numérico es 0. En este caso, se asigna el carácter '0' (30H) directamente a la dirección de memoria apuntada por SI.

Etiqueta L_FIN: Finalización del macro.

MANUAL DE USUARIO

- Acceso al sistema y encabezado



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Program: PRA
Copyright (C) Microsoft Corp 1984-1992. All rights reserved.

Object Modules [.obj]: pra.obj
Run File [pra.exe]: "pra.exe"
List File [nul.map]: NUL
Libraries [.lib]:
Definitions File [nul.def]:

C:\>pra.exe

Universidad de San Carlos de Guatemala
Arquitectura de Computadoras y Ensambladores 1
Seccion N
Escuela de vacaciones de Junio 2023

CARLOS JAVIER MARTINEZ POLANCO 201709282
=====

VALIDANDO USUARIO
VALIDANDO CLAVE
CREDENCIALES CORRECTAS
-> Presione Enter para continuar...
```

- Menú principal

```
-----  
                                MENU DEL SISTEMA  
  
(1) OPCIONES DE PRODUCTOS  
(2) OPCIONES DE VENTA  
(3) OPCIONES DE HERRAMIENTAS DEL SISTEMA  
(4) SALIR DEL SISTEMA  
  
INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR:
```

- Menú productos

```
                                MENU PRODUCTOS  
  
(1) INGRESAR PRODUCTOS  
(2) ELIMINAR PRODUCTOS  
(3) VER PRODUCTOS  
(4) ATRAS  
  
INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: _
```

- Ingresar producto

```
INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 1
HAS SELECCIONADO INGRESAR PRODUCTO

INGRESA EL CODIGO DEL PRODUCTO: 1111

INGRESA LA DESCRIPCION DEL PRODUCTO: producto8A

INGRESA EL PRECIO DEL PRODUCTO: 25

INGRESA LAS UNIDADES DISPONIBLES DEL PRODUCTO: 1_
```

- Ver productos

```
1111
prod1
2222
gfdsgd
3333
gfdsgds
4444
dfgsdds
1111
producto8A
```

MENU PRODUCTOS

- (1) INGRESAR PRODUCTOS
- (2) ELIMINAR PRODUCTOS
- (3) VER PRODUCTOS
- (4) ATRAS

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR:

- Eliminar producto

```
INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 2
HAS SELECCIONADO ELIMINAR PRODUCTO
INGRESA EL CODIGO DEL PRODUCTO: 3333
```

- Menú ventas

```
          MENU VENTAS

(1) INGRESA LOS DATOS DE LA VENTA
(2) ATRAS

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR:
```

- Menú herramientas

```
          MENU HERRAMIENTAS

(1) GENERAR CATALOGO
(2) REPORTE PRODUCTOS
(3) REPORTE VENTAS
(4) REPORTE PRODUCTOS SIN EXISTENCIA
(5) ATRAS

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: _
```

- Generar reporte catalogo

```

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 3

MENU HERRAMIENTAS

(1) GENERAR CATALOGO
(2) REPORTE PRODUCTOS
(3) REPORTE VENTAS
(4) REPORTE PRODUCTOS SIN EXISTENCIA
(5) ATRAS

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 1
SE HA GENERADO EL CATALOGO EXITOSAMENTE

```

- Reporte de catalogo de productos

← → ↻ ⓘ Archivo | C:/ensamblador/vacas/pr2/CATALG.HTM

YouTube 12.2. Conectarse a l... Eazel search Conexión a Base de... Python - Código si... Laser Pointer Rem

codigo	descripcion
1111	prod1
2222	gfdsgd
3333\$g	gfdsgds
4444\$d	dfgsdds\$
1111	producto8A

REPORTE GENERADO LA FECHA: 23/6/23 HORA: 23:11

- Navegación y finalización del programa

```

                                MENU HERRAMIENTAS

(1) GENERAR CATALOGO
(2) REPORTE PRODUCTOS
(3) REPORTE VENTAS
(4) REPORTE PRODUCTOS SIN EXISTENCIA
(5) ATRAS

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 5

                                MENU DEL SISTEMA

(1) OPCIONES DE PRODUCTOS
(2) OPCIONES DE VENTA
(3) OPCIONES DE HERRAMIENTAS DEL SISTEMA
(4) SALIR DEL SISTEMA

INGRESA LA OPCION DEL MENU QUE DESEAS UTILIZAR: 4
C:\>
```