Programación de Sistemas
Práctica No. 2
El proceso de compilación en C

Objetivo:

El alumno conocerá las diferentes etapas de la compilación de programas en C y generará y analizará los diferentes archivos que se obtienen en cada una de esas etapas.

Introducción

Los pasos de compilación de programas en C son los siguientes:

- Preproceso (expansión de macros, inclusión de ficheros, eliminación de comentarios, ...)
- Conversión a ensamblador.
- Compilación a objeto.
- Enlace (linking) de objetos y bibliotecas.

Actividades a realizar

Trabajando en el IDE (entorno de desarrollo) con un archivo único

1.- Trabajando con el editor de Turbo C, captura en un archivo llamado `tams.c` el código siguiente,

```
/* ********************************************************
 * Archivo: tams.c
 * Despliega los tamaños de cada tipo básico que maneja este
 * compilador.
 * ******************************************************** */
 #include <stdio.h>
 main()
 {
      int i;
      long l;
      short s;
      float f;
      char c;
      double d;

      printf("Tamaño de los tipos básicos en este compilador\n");
      printf("Tipo char = %d byte.\n",sizeof(c));
      printf("Tipo short = %d bytes.\n",sizeof(s));
      printf("Tipo entero = %d bytes.\n",sizeof(i));
      printf("Tipo entero largo = %d bytes.\n",sizeof(l));
      printf("Tipo flotante = %d bytes.\n",sizeof(f));
      printf("Tipo flotante doble = %d bytes.\n",sizeof(d));

      return 0;
 }
```

1.1.- El IDE de Borland tiene las siguientes opciones en el Menú `Compile`: `Compile`, `Make`, `Link` y `Build All`.

1.2.- La opción `Compile` genera sólo el archivo objeto. Ejecútala y revisa el directorio `bin` (desde una ventana de MS-DOS ubícate en el directorio `c:\tc\bin` tecleando `cd c:\tc\bin` y, una vez ahí, teclea `dir tams.*`) para determinar los datos del archivo `tams.obj`:
Tamaño: _____ Fecha y hora de creación: _____

1.3.- La opción `Make` compila y liga para generar el archivo `tams.exe`. Ejecútala y revisa el directorio `bin` para determinar los datos del archivo `tams.exe`:
Tamaño: _____ Fecha y hora de creación: _____

1.4.- En la ventana de MS-DOS, ejecuta el programa tecleando `tams<enter>` y reporta los tamaños de los siguientes tipos de datos:

Tipo	Tamaño (en bytes)
`char`	_____
`short`	_____
`int`	_____
`long int`	_____
`float`	_____
`double`	_____

2.- Trabajando en el IDE (entorno de desarrollo) con un programa dividido en varios archivos

2.1.- En una ventana nueva captura el archivo `m.c`:

```
/* Archivo: m.c
 * Ilustra el manejo de programas multimodulos
 */
extern void a(char *);
int main()
{
    static char texto[]="Hola Mundo desde programa multimodulo!.\n";
    a(texto);
    return 0;
}
```

2.2.- En otra ventana nueva captura el archivo `a.c`:

```
/* Archivo: a.c
 * Ilustra el manejo de programas multimodulos
 */
#include<io.h>
#include<string.h>
void a(char *s)
{
     write(1,s,strlen(s));
}
```

En el IDE, cuando un programa está dividido en varios módulos debe definirse un proyecto para que el entorno sepa que debe compilar y ligar varios archivos para generar el ejecutable.

2.3.- Usando el menú `Project -> Open Project`, genera el proyecto `PROYECT1.PRJ`
Aparece una ventana en la parte inferior del IDE con el título `Project: Proyect1`?
Sí: ___ No: ____

2.4.- Para indicarle al entorno cuáles son los archivos de este proyecto, en el menú `Project -> Add item` agregamos los archivos `a.c` y `m.c`. Al agregarse estos dos archivos, la ventana inferior (con el título `Project: Proyect1`) debe reflejar estos cambios. Aparecen los archivos `a.c` y `m.c` en esta ventana? Sí: ___ No: ____

2.5.- Para indicarle al entorno que compile y lige todos los archivos del proyecto, en el menú `Compile` elegimos la opción `Build All`

Ejecuta la opción anterior y contesta las siguientes preguntas:

Nombre del archivo ejecutable generado: _____
Revisa el directorio `bin` para determinar los datos del archivo ejecutable generado:
Tamaño: _____ Fecha y hora de creación: _____

3.- Trabajando desde la línea de comandos

En este parte de la práctica vamos a verificar los pasos de la compilación en C que, cómo comentamos en la introducción, son: *i*) preproceso, *ii*) Conversión a ensamblador, *iii*) Conversión a lenguaje máquina y *iv*) Ligado de rutinas y librerías. Lo haremos desde la línea de comandos para ilustrar mejor este proceso.

En el compilador C/C++ de Borland, el preproceso lo realiza el programa `cpp.exe`. Su forma de uso es:

```
cpp -oarchivo_salida archivo_fuente
```

Por ejemplo, si queremos preprocesar el archivo `m.c` y guardar la salida del preprocesador en el archivo `m.txt`, tecleariamos:

```
cpp -om.txt m.c
```

En esta parte vamos a invocar el compilador `tcc.exe` desde la línea de comandos. La forma general de la línea de ordenes es:

```
tcc [-opcion1 -opcion2 ... -opcionN] nombre1 nombre2 ... nombreM
```

Donde `opcion`*i* es una opción de compilación o enlace y `nombre`*j* es o un archivo fuente, o un archivo ASM o un archivo OBJ o una librería.

Por ejemplo, suponga que tiene un programa llamado `X.C`. Para compilarlo y ligarlo desde la línea de comandos, deberá teclear

```
tcc x.c
```

Si no hay errores en el programa `X.C`, se compilará y enlazará con las librerias adecuadas para generar el ejecutable `X.EXE`. Esta es la forma más simple de la línea de comandos.

Algunas opciones para el compilador desde la línea de comandos son:

`tcc -c nombre.c`	Sólo genera el archivo OBJ sin ligar.
`tcc -S nombre.c`	Sólo traduce a ensamblador
`tcc nombre1.obj nombre2.obj`	Liga los archivos objeto `nombre1.obj` y `nombre2.obj` para generar el ejecutable `nombre1.exe`.

Por tanto, si sólo quiere generar el archivo objeto de X.C, el comando será `tcc -c X.C`, lo que generará el archivo `X.OBJ`. Si sólo quiero traducirlo a ensamblador, el comando será `tcc -S X.C`, lo que generará el archivo `X.ASM`.

Si quiero completar la compilación (esto es ejecutar el ligador), suponiendo que tengo el archivo objeto, se teclea: `tcc X.OBJ` lo que generará el archivo ejecutbale `X.EXE`.

Si quiero completar la compilación, suponiendo que tengo el archivo en ensamblador, se teclea: `tcc X.ASM` lo que generará el archivo ejecutbale `X.EXE`.

3.1.- Desde una ventana de MS-DOS, ubícate en el directorio `c:\tc\bin` tecleando `cd c:\tc\bin` y, una vez ahí,

"preprocesa" los programs `m.c` y `a.c` con los siguientes comandos:

```
cpp -oa.txt a.c
cpp -om.txt m.c
```

Con el comando `dir ?.txt` determina los datos de los archivos `m.txt` y `a.txt`
Archivo `m.txt` Tamaño: _____ Fecha y hora de creación: _____
Archivo `a.txt` Tamaño: _____ Fecha y hora de creación: _____

Usando el comando `more m.txt`, despliega el contenido del archivo `m.txt` y comenta que sucedió con las líneas de comentarios: _____
_____

Usando el comando `more a.txt`, despliega el contenido del archivo `a.txt` y comenta que sucedió con las líneas de comentarios y con los `#include<..>`: _____
_____
_____
_____
_____
_____

3.2.- Desde un ventana de MS-DOS, ubícate en el directorio `c:\tc\bin` tecleando `cd c:\tc\bin` y, una vez ahí, genera el ensamblador de los programs `m.c` y `a.c` con los siguientes comandos:

```
tcc -S m.c
tcc -S a.c
```

Con el comando `dir *.asm` determina los datos de los archivos `m.asm` y `a.asm`
Archivo `m.asm` Tamaño: _____ Fecha y hora de creación: _____
Archivo `a.asm` Tamaño: _____ Fecha y hora de creación: _____

Usando el comando `more m.asm`, despliega el contenido del archivo `m.asm`, busca las instrucciones `call ...` que encuentres en el código ensamblador y comenta brevemente sobre ellas:
_____
_____
_____
_____
_____

Usando el comando `more a.asm`, despliega el contenido del archivo `a.asm`, busca las instrucciones `call ...` que encuentres en el código ensamblador y comenta brevemente sobre ellas:
_____
_____
_____
_____
_____

3.3.- Vamos a la siguiente etapa de la compilación generando los archivos objeto tecleando:

```
tcc -c m.asm
tcc -c a.asm
```

Con el comando `dir *.obj` determina los datos de los archivos `m.obj` y `a.obj`
Archivo `m.obj` Tamaño: _____ Fecha y hora de creación: _____
Archivo `a.obj` Tamaño: _____ Fecha y hora de creación: _____

3.4.- Vamos a completar el proceso de compilación ejecutando el ligado de los archivos tecleando:

```
tcc a.obj m.obj
```

Con el comando `dir *.exe` determina los datos del archivo `a.exe`
Archivo `a.exe` Tamaño: _____ Fecha y hora de creación: _____

¿Porqué crees que el ejecutable es mucho mayor que la suma de los tamaños de los archivos objeto?
_____
_____

3.5.- Ahora teclea el comando:

```
tcc m.obj a.obj
```

Con el comando `dir *.exe` determina los datos del archivo `m.exe`
Archivo `m.exe` Tamaño: _____ Fecha y hora de creación: _____

Por último, teclea el siguiente comando:

```
tcc -ehola1.exe m.obj a.obj
```

Con el comando `dir *.exe` determina los datos del archivo `hola1.exe`
Archivo `hola1.exe` Tamaño: _____ Fecha y hora de creación: _____

3.6.- Tomando en cuenta las opciones de la línea de comandos para el compilador `tcc.exe`, ¿Cuál crees que debe ser el comando para generar el ejecutable `varios.exe` a partir del archivo fuente `tams.c`?

Comando: _____

Archivo ejecutable generado: _____
Tamaño: _____ Fecha y hora de creación: _____

Ejecútalo y reporta el resultado de su ejecucion: _____
_____
_____

4.- Comantarios y conclusiones
_____
_____
_____
_____
_____
_____
_____
_____
_____