

Clase #06 de 27

Constantes e

Introducción a Streams

Mayo 10, Lunes

Agenda para esta clase

- Constantes simbólicas
- Funciones del Preprocesador – Parte II
- Flujos y copia

Constantes simbólicas

K&R 1.4 Constantes Simbólicas

“Números mágicos”

```
#include <stdio.h>
```

```
int main(void){  
    int fahr;
```

```
    for( fahr = 0; fahr <= 300; fahr = fahr + 20 )  
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
```

```
    return 0;  
}
```

Constantes (o Nombres) Simbólicas

```
#include <stdio.h>
#define LOWER 0 // lower limit of table
#define UPPER 300 // upper limit
#define STEP 20 // step size

int main(void){
    int fahr;

    for( fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP )
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32) );
}
```

C11: Calificador const y Declaración en for

```
int main(void){  
    const int LOWER = 0,    // lower limit of table  
              UPPER = 300, // upper limit  
              STEP  = 20;   // step size  
  
    for (int fahr = LOWER; fahr <= UPPER; fahr = fahr + STEP)  
        printf("%3d %6.1f\n", fahr, (5.0/9.0)*(fahr-32));  
}
```

- Ejercicio 1-5. Modifique el programa para que imprima la tabla en orden inverso, es decir, desde 300 hasta 0 grados.

Funciones del Preprocesador

Parte II

Funciones del Preprocesador, Parte II

- Ya conocidas
 - Incluir archivos encabezado, directiva `#include`
 - Reemplazar comentarios por un espacio
 - Concatenar cadenas adyacentes.
- Nueva
 - Atender directivas `#define` que definen nombres macro y su lista de reemplazo
 - Expandir los nombres macro
- Best Practice
 - Evitar, en los posible, el uso `define`, buscar alternativas más abstractas y con misma eficiencia.
 - http://www.stoustrup.com/bs_faq2.html#macro

Introducción a Fases de Traducción y el rol del Preprocesador: Fases 3, 4, y 6

- Fase 1
 - Mapeo de caracteres y reemplazado de *trigrafos*
- Fase 2
 - Unión de líneas físicas que terminan en \
- Fase 3
 - Descomposición en *tokens de preprocesamiento* y en secuencias de blancos, que incluye comentarios.
 - Reemplazo de comentarios por espacio en blanco.
- Fase 4
 - Ejecución de directivas, expansión de macros
 - La directiva *#include* implica repetir las fases 1 a 4 recursivamente.
 - Borrado de directivas.
- Fase 5
 - Conversión de constantes carácter y literales cadena al set de caracteres de ejecución.
- Fase 6
 - Concatenación de cadena literales adyacentes.
- Fase 7
 - Los blancos que separan *tokens de preprocesamiento* ya no son necesarios,
 - cada token de preprocesamiento se convierte a un *token*,
 - los tokens resultantes se analizan *sintácticamente y semánticamente* como una *unidad de traducción*.
- Fase 8
 - Todas las *referencias* a objetos y funciones *externos* se resuelven.
 - Los *componentes* de biblioteca se vinculan.
 - La salida de este proceso se reúne en una *imagen de programa* que tiene la información necesaria para su ejecución en el *ambiente de ejecución*.

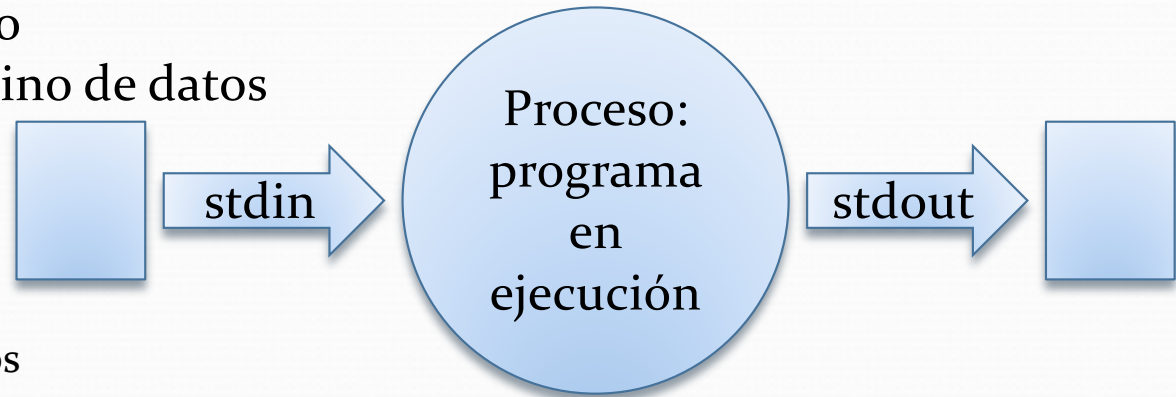
Entrada y Salida con Flujos de Texto Standard

K&R 1.5 Entrada y Salida de a Caracteres
1.5.1 Copia de archivos

Flujos (Streams) y Archivos

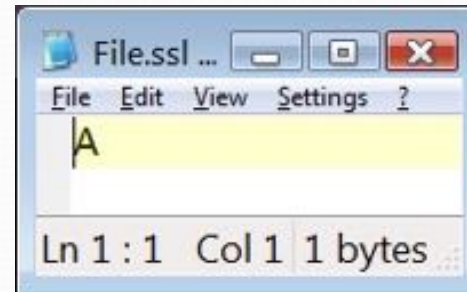
- Proceso como sistema abierto
- Flujo de datos: Origen o destino de datos

- Secuencia de Datos
- Origen
 - Archivo
 - Otro Procesos
 - Teclado u otros dispositivos
- Destino
 - Archivo
 - Otro Procesos
 - Pantalla u otros dispositivo



- Diferencia entre flujo y archivo

- Durante la lectura
 - ¿Cambia el archivo?
 - ¿Cambia el flujo?
- Durante la escritura
 - ¿Cambia el archivo?
 - ¿Cambia el flujo?



A
65
0100 0001

File.ssl
(1 byte)

- ¿Existe diferencia entre un "Archivo Binario" y un "Archivo de Texto"?

Primitivas para operar con Flujos

- Un flujo es una secuencia de bytes
 - Relación entre char y byte
- Primitiva putchar
 - Recibe un char no signado dentro de un int
 - Retorna EOF si no pudo enviar el dato
- Primitiva getchar
 - Retorna un char no signado dentro de un int, ó el int EOF
- EOF es una Señal
 - Valor abstraído
 - Es un int negativo
 - Semántica
 - Fin por finalización
 - Fin por error.

int			
0000 0000	0000 0000	0000 0000	0100 0001
unsigned char	unsigned char	unsigned char	unsigned char



Algoritmo de Lectura para Recorrer un Flujo

- Problema “Implementar la función identidad”: Sale lo que entra, lo que entra sale
- ¿Cuándo finaliza de procesar?
- ¿Cómo se programa en Pascal?
 - ¿Cuál es el diagrama N-S?
- ¿Hay que hacer una lectura previa?
- ¿Cómo se programa en C++?
 - ¿Cuál es diagrama N-S?
- ¿Cómo se programa en C?
 - ¿Cuál es diagrama N-S?.

Copiar entrada a salida; 1era versión

Leer un carácter
Mientras el carácter no
sea el indicador de fin
Mostrar ese carácter
Leer otro carácter

```
#include <stdio.h>

// copy input to output; 1st version
int main(void){
    int c;

    c = getchar();
    while (c != EOF) {
        putchar(c);
        c = getchar();
    }
}
```


Copiar entrada a salida; 2da versión

```
#include <stdio.h>
```

```
// copy input to output; 2nd version
int main(void){
    for (int c; (c = getchar()) != EOF;)
        putchar(c);
}
```

Mientras(haya caracteres)
 Enviar carácter

```
/* c89: */
int main(void){
    int c;

    while ( (c = getchar()) != EOF)
        putchar(c);
}
```

- “*Idiom*”, Expresión idiomática, frase hecha
- Precedencia
- Expresiones equivalentes
 - `c = getchar() != EOF`
 - `c = (getchar() != EOF)`

Ejercicios

- 1-6. Verifique que la expresión `getchar() != EOF` es cero o uno
- 1-7. Escriba un programa que imprima el valor de EOF.

Términos de la clase #6

Definir cada término con la bibliografía

- Constantes simbólicas
 - “Números mágicos”
 - Constantes simbólicas ó Nombres simbólicos
 - Directiva #define
 - Calificador const
- Funciones del Preprocesador Parte II
 - Definición
 - Expansión
 - Best Practice: Evitar defines.
- Flujos y copia
 - Stream (Flujo de datos)
 - Archivo versus Flujo
 - stdin
 - stdout
- "Diferencia" entre Archivo Binario y Archivo de Texto
- Primitivas para operar con flujos
- putchar
- getchar
- Tipos char, signed char, unsigned char
- EOF
- unsigned int
- “Idiom”, Frase hecha, Expresión idiomática
- Idiom para recorrer flujo de a caracteres
- Precedencia (ó Prioridad) de operadores

Tareas para la próxima clase

1. Leer de [K&R1988] **1.5 Entrada y Salida de a Caracteres**
2. Entregar el trabajo “Fases de la Traducción y Errores”.

¿Consultas?

Fin de la clase