

# Clase #15 de 27

## Variables Externas & Niveles de Lenguaje Parte I

*Sep 6, Lunes*

# Agenda para esta clase

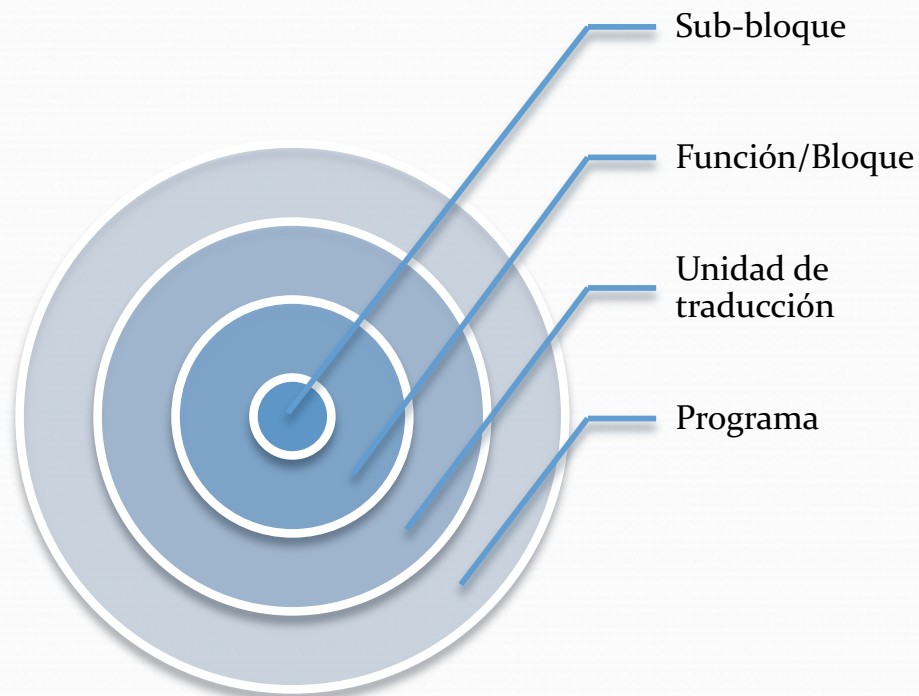
- Parte I
  - Variables Externas y Alcance
- Parte II
  - Niveles de Lenguaje

# Variables Externas y Alcance

K&R 1.10 Variables Externas y Alcance

Última parte de Introducción a ANSI C

# Alcance y Vinculación



- Programa
  - ¿Qué es un programa?
    - Conjunto de Unidades de Traducción (UT, archivos fuente .c)
  - Vinculación entre todas las UT
  - o entre solo algunas UT
  - o entre algunas funciones de algunas UT
- UT
  - Alcance"entre todas las funciones
  - o vinculación entre solo algunas funciones
- Función
  - Alcance en el bloque del cuerpo que la define
- Sub Bloque
  - Alcance en el subbloque.

# Variables Automáticas versus Variables Externas

## Variables Automáticas

- Variable Automática: privada o local
  - Solo la función tiene acceso directo
  - ¿Parámetros?
    - Son variables automáticas
- ¿Por qué automáticas?
  - Reserva y liberación de forma automática
    - Push de argumentos
    - Pop en parámetros
  - Comienzan al invocar la función
  - Desaparecen al terminar
  - No retienen valor entre invocaciones
  - Sin inicialización  $\Rightarrow$  Basura
- Mismo Nombre  $\nRightarrow$  Mismo Objeto
  - Variable i de getline y de copy

## Variables Externas

- Definidas externas a toda función: Públicas o Globales
- Acceso directo, por nombre, desde "todas" las funciones de la unidad de traducción
  - Antes de usar un nombre de variable, debe estar declarado
  - Comparar con el bloque más externo de Pascal
- Alternativa a lista de argumentos para comunicación entre funciones
- Definición y Declaración
  - Definida una sola vez, de forma externa  $\Rightarrow$  Reserva de memoria
  - Declaradas (tipo de dato) en cada función que la usa
    - De forma Explícita con keyword extern
    - De forma Implícita por contexto y orden
- Vinculadas con otras
  - El Linker conecta las variables
  - ¿Entre otras unidades de traducción o en una misma unidad de traducción?
- Son variables estáticas
  - Permanentes  $\Rightarrow$  Retienen el valor
  - Inicialización en cero.



# Duración del Almacenamiento, Tiempo de Vida, Alcance y Vinculación

## Storage Duration, Lifetime, Scope, Linkage

- Duración del Almacenamiento, Tiempo de Vida (Storage Duration, Lifetime)
  - Estática
    - Externa
    - Keyword static
  - Automática
    - Parámetros
    - Bloque
    - Keyword auto
  - "Alocada"
    - Funciones estándar
      - malloc
      - calloc
      - realloc
      - free
  - Thread
    - C99
- Alcance (Scope)
  - Desde la declaración
    - En UT
    - En función
    - En bloque
- Retención de valor entre invocaciones
  - Variable declarada dentro de la función, con la keyword static
  - Alcance: bloque
  - Duración: estática
- Vinculación (Linkage)
  - Funciones y variables
    - Entre varias UT
      - Vinculación Externa
    - Dentro de UT
      - Vinculación Interna
  - Sin vinculación
    - struct, enum, typedef, label

# Ejemplo – Versión especializada de Imprimir la línea más larga (1 de 5)

```
#include <stdio.h>
#define MAXLINE 1000 /* maximum input line size */

/*Analizar semántica*/
int max;              /* maximum length seen so far */
char line[MAXLINE];   /* current input line */
char longest[MAXLINE]; /* longest line saved here */

int getline(void);
void copy(void);
```

- Declaraciones externas de variables
- Prototipos, ya se usaron como declaraciones externas, pueden hacerse privados a las UT con palabra reservada `static`
- Definición de variables y de funciones.

# Ejemplo – Versión especializada de Imprimir la línea más larga (2 de 5)

```
int main(void){  
    int len;  
  
    /* Analizar semántica */  
    extern int max;  
    extern char longest[];  
  
    ...  
}
```

- Declaración externa
  - Fuera de cualquier función
- Declaración con extern
  - usa extern delante
  - Es opcional si existe definición previa
  - Declaraciones con extern en header (.h)
    - Para vincular entre UT
    - Convención y razón del nombre
  - Puede ser una declaración externa o interna
  - Puede ser definición.





# Ejemplo – Versión especializada de Imprimir la línea más larga (3 de 5)

```
/* Uso de externas y de automáticas */
max = 0;
while ((len = getline()) > 0)
    if (len > max) {
        max = len;
        copy();
    }

if (max > 0) /* there was a line */
    printf("%s", longest);

return 0;
}
```

# Ejemplo – Versión especializada de Imprimir la línea más larga (4 de 5)

```
int getline(void){
    int c, i;
    extern char line[]; /*Analizar semántica*/

    for(
        i=0;
        i < MAXLINE - 1 && (c=getchar())!=EOF && c!='\n';
        ++i
    )
        line[i] = c;

    if (c == '\n') {
        line[i] = c;
        ++i;
    }

    line[i] = '\0';
    return i;
}
```

# Ejemplo – Versión especializada de Imprimir la línea más larga (5 de 5)

```
void copy(void){  
    int i;  
  
    /*Analizar semántica*/  
    extern char line[], longest[];  
  
    i = 0;  
    while ((longest[i] = line[i]) != '\0')  
        ++i;  
}
```

# Identificar declaraciones, definiciones, variables automáticas y variables externas

```
#include <stdio.h>
#define MAXLINE 1000

int max;
char line[MAXLINE];
char longest[MAXLINE];

int getline(void);
void copy(void);

int main(void){
    int len;

    extern int max;
    extern char longest[];

    max = 0;
    while ((len = getline()) > 0)
        if (len > max) {
            max = len;
            copy();
        }

    if (max > 0)
        printf("%s", longest);

    return 0;
}
```

```
int getline(void){
    int c, i;
    extern char line[];

    for(
        i=0;
        i < MAXLINE - 1 &&
        (c=getchar())!=EOF && c!='\n';
        ++i
    )
        line[i] = c;

    if (c == '\n') {
        line[i] = c;
        ++i;
    }

    line[i] = '\0';

    return i;
}

void copy(void){
    int i;
    extern char line[], longest[];

    i = 0;
    while((longest[i]=line[i])!='\0')
        ++i;
}
```

# Utilización de variables externas

- Pueden ser modificadas accidentalmente
  - Definir al final y declarar solo en las funciones que las deben usar
  - Un programa C puede estar compuesto por varios ó UT: escribir módulos chicos
- Similares a los atributos de instancia o de clase
  - Accesibles por todos los métodos de esa clase
  - En este caso la clase es análoga a la UT
- Las variables externas siempre están disponibles
  - Ventajas
    - Disponibilidad
    - Lista de argumentos más cortas
    - Simplifica comunicación
    - Objetos únicos: stdin, CPU
  - Desventajas
    - Acoplamiento no obvio
    - Bugs
    - Mantenimiento
- En este caso perdemos la oportunidad de crear dos funciones genéricas: getline y copy
  - ANSI provee: fgets y strcpy.



# Bibliografía

- [K&R1988] 1.10 Variables Externas y Alcance
- [K&R1988] 4.3 Variables Externas
- [K&R1988] 4.4 Reglas de Alcance
- [K&R1988] 4.6 Variables Estáticas

# Ejercicios

- De ejemplos de variables
  - Con alcance de bloque (i.e., local), con duración estática
  - Con alcance de archivo, con vinculación interna
- De ejemplos de identificadores
  - Sin vinculación
  - Con vinculación interna
  - Con vinculación externa
  - Con alcance función
- Explique que es la vinculación de identificadores
- Explique la diferencia entre duración y alcance
- Explique la diferencia entre espacio de nombre y alcance
- De ejemplo de nombre de identificadores léxicamente iguales pero que pertenecen a diferentes espacios de de nombre
- Comparación de Lenguajes: elija un lenguaje diferente a C y a C++ y ejemplifique los siguientes conceptos:
  - Duración
  - Alcance
  - Espacio de nombre

# Términos de la clase #15 Parte I

Definir cada término con la bibliografía

- Variables Externas
  - ¿Qué es un programa?
  - Unidad de Traducción (UT)
  - Vinculación entre varias UT
  - Alcance dentro de una UT
  - Vinculación entre diferentes UT
  - Bloque
  - Sub-bloque
  - Variable automática
  - Variable externa
  - Variable estática
  - Vínculo entre variables
  - Declaración de variables
  - Definición de variables
  - Alcance ó Alcance Léxico
  - Duración del tiempo de vida o lifetime
- Estática
- Automática
- "Alocada"
- Declaración Externa
- Declaración con extern
- Declaración versus Definción
- Stack de invoación
- Incialización de estáticas
- Ocultamiento de información
- Static para encapsular
- Posincrmento ó Incremento (sufijo)
- Predecremento ó Decremento (prefijo)
- Aplicación de variables externas.
- Variable

# ¿Consultas?

**Fin de la clase**