

CFGM SISTEMAS MICROINFORMÁTICOS Y REDES

Sistemas Operativos Monopuesto

TEMA 03: Fundamentos de Programación I

Estructuras de Selección

Contenido

- Sentencias y composición
- Estructuras de selección
- Sentencia if
- Sentencia if-else
- Sentencias if-else anidadas
- Sentencia switch
- Ejercicios

Sentencias

- Un programa se compone de un conjunto de sentencias (instrucciones).
- Las sentencias puede ser **básicas**:
 - `printf("Hola\n");`
 - `scanf("%d",&horas);`
 - `x = x+23;`
- Para cualquier programa no trivial necesitaremos más de una sentencia básica. Por tanto necesitamos mecanismos para crear sentencias más complicadas a partir de las sentencias básicas. A las sentencias que combinan varias sentencias básicas les llamaremos **sentencias compuestas**.

Mecanismos de composición

- Existen tres mecanismos básicos de crear sentencias compuestas a partir de sentencias básicas:
 - **Secuencia:** “Ve a la panadería. Pide un pan. Págalo. Tráelo a casa. Córtalo en pedazos.”
 - **Selección:** “Si llueve coge el paraguas, si no coge el abanico.”
 - **Repetición:** “Copia 100 veces ‘No hablaré en clase’”

Composición secuencial

- En C la composición secuencial se consigue:
 - Abriendo la secuencia con una llave {
 - Utilizando el punto y coma para unir las sentencias que se desean componer
 - Cerrando la secuencia con otra llave }

{

s_1;

s_2;

s_3;

}

Ejecuta la sentencia s_1, cuando termines, ejecuta la sentencia s_2 y cuando termines ejecuta la sentencia s_3

Estructuras de selección

- En ocasiones es necesario ejecutar un conjunto de sentencias u otras en función de la situación en la que se encuentra el programa.
- En C existen dos sentencias que permiten crear **estructuras de selección**:
 - **if**
 - **switch**

Sentencia if

if (*condición*) *sentencia*

**Si la condición es verdadera ejecuta la sentencia.
Si es falsa no ejecuta nada.**

Ejemplo 1

Programa que lee dos números enteros ($n1$ y $n2$) y en el caso de que el primero sea mayor que el segundo muestra “ $n1$ es mayor que $n2$ ”

Ejemplo 2

Programa que lee dos números enteros ($n1$ y $n2$) y si el primer número es mayor que cero muestra “ $n1$ es mayor que cero”. Además, si el segundo número es mayor que cero muestra “ $n2$ es mayor que cero”

Sentencia if-else

```
if (condición)  
    sentencia_1  
else  
    sentencia_2
```

Si la condición es verdadera ejecuta la sentencia *sentencia_1*. Si es falsa ejecuta la sentencia *sentencia_2*

Ejemplo

Programa que lee dos números enteros ($n1$ y $n2$) y si el primer número es múltiplo del segundo muestra “ $n1$ es múltiplo de $n2$ ”. En caso contrario muestra “ $n1$ no es múltiplo de $n2$ ”.

Múltiples alternativas

- 1 alternativa -> if
- 2 alternativas -> if-else
- ¿Cómo podemos hacer si tenemos más de 2 alternativas?
- Ejemplo: Programa que lee un numero real que representa la nota de una asignatura y muestra si es un suspenso $[0,5)$, un suficiente $[5,6.5)$, un notable $[6.5,8.5)$ o un excelente $[8.5,10]$
- En ese caso podemos utilizar **sentencias if-else anidadas**

Sentencias if-else anidadas

```
if (condicion_1)  
    sentencia_1;  
else if (condicion_2)  
    sentencia_2;  
...  
else if (condicion_n-1)  
    sentencia_n-1;  
else  
    sentencia_n;
```

Si la condición *condicion_1* es verdadera ejecuta la sentencia *sentencia_1*. Si no lo es, analiza la condición *condicion_2* y si es verdadera, ejecuta la sentencia *sentencia_2*

...

si finalmente ninguna de las $n-1$ condiciones son verdaderas, ejecuta la sentencia *sentencia_n*

Ejemplo

Programa que lee un numero real que representa la nota numérica de una asignatura y muestra si es un suspenso $[0,5)$, un suficiente $[5,6.5)$, un notable $[6.5,8.5)$ o un excelente $[8.5,10]$

Alternativas aún más múltiples

- En ocasiones se da el caso de que existen muchas alternativas que dependen de una única variable.
- Ejemplo: Programa que lee un numero entero que representa un mes (1 -> enero, 12->diciembre) y muestra el nombre del mes.
- Podríamos utilizar sentencias if-else anidadas, pero queda demasiado extenso.

Sentencia switch

```
switch (selector) {  
    case etiq_1:  
        sentencia_1;  
    ...  
    case etiq_n:  
        sentencia_n;  
    default:  
        sentencia_n+1;  
}
```

Comprueba si el valor del selector es el de la etiqueta *etiq_1*. Si no lo es comprueba *etiq_2*,...

Si encuentra una etiqueta *etiq_i* que coincide con el valor del selector. A partir de ese momento ejecuta las sentencias desde *sentencia_i* hasta *sentencia_n+1*.

En otro caso ejecuta la sentencia *sentencia_n+1*.

Sentencia switch-break

```
switch (selector) {  
    case etiq_1:  
        sentencia_1;  
        break;  
  
    ...  
    case etiq_n:  
        sentencia_n;  
        break;  
  
    default:  
        sentencia_n+1;  
  
}
```

Comprueba si el valor del selector es el de la etiqueta *etiq_1*. Si no lo es comprueba *etiq_2*,...

Si encuentra una etiqueta *etiq_i* que coincide con el valor del selector ejecuta la sentencia *sentencia_i*.

En otro caso ejecuta la sentencia *sentencia_n+1*

Ejemplo

Programa que lee un numero entero que representa un mes (1 -> enero, 12->diciembre) y muestra el nombre del mes.