

02 String

Programación Orientada a Objetos

Néstor Suat-Rojas. Ing. Msc (c)
nestor.suat@unillanos.edu.co

Escuela de Ingeniería
Facultad de Ciencias Básicas e Ingeniería

Introducción

C++ tiene una forma de representar una secuencia de caracteres como un objeto de clase, llamado `string`. La clase `string` es usado para almacenar textos, las palabras son encerradas con comillas dobles.

```
#include <iostream>
#include <string>

using namespace std;

int main() {
    string mensaje="Hola mundo";
    cout << mensaje;
    return 0;
}
```

String vs Char*

En ejecución los arreglos de caracteres son más rápidos que los `string`. Sin embargo los `string` tiene muchas funciones para manipular cadenas.

String

```
int main() {  
    string mensaje="Hola mundo";  
    cout << mensaje;  
    return 0;  
}
```

Char array

```
int main() {  
    char* mensaje="Hola mundo";  
    cout << mensaje;  
    return 0;  
}
```

String Concatenation

El operador + puede ser usado entre strings para concatenar uno al otro.

Operador +

```
string firstName = "John ";  
string lastName = "Doe";  
string fullName = firstName + lastName;  
cout << fullName;
```

```
string firstName = "John";  
string lastName = "Doe";  
string fullName = firstName + " " + lastName;  
cout << fullName;
```

Función append

```
string firstName = "John ";  
string lastName = "Doe";  
string fullName = firstName.append(lastName);  
cout << fullName;
```

String Concatenation con Números

La primera sobrecarga de operadores que vemos es el operador +, que permite sumar números o concatenar cadenas.

Operador + con números

```
int x = 10;  
int y = 20;  
int z = x + y;
```

Operador + con cadenas

```
string firstName = "John";  
string lastName = "Doe";  
string fullName = firstName + " " + lastName;  
cout << fullName;
```

Concatenando cadenas y números

```
int main() {  
    string x = "10";  
    int y = 20;  
    string z = x+to_string(y);  
}
```

String Length

Para obtener el tamaño de un string podemos usar la función length.

length()

```
string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
cout << "The length of the txt string is: " << txt.length();
```

size()

```
string txt = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";  
cout << "The length of the txt string is: " << txt.size();
```

Accediendo a strings

También se puede acceder a únicamente un carácter de toda la cadena.

Acceder

```
string myString = "Hello";  
cout << myString[1];  
// Outputs e
```

Modificar

```
string myString = "Hello";  
myString[0] = 'J';  
cout << myString;  
// Outputs Jello instead of Hello
```


Entradas de usuario con String

En ocasiones es posible que pidamos al usuario ingresar una cadena de texto.

```
string fullName;  
cout << "Type your full name: ";  
cin >> fullName;  
cout << "Your name is: " << fullName;  
  
// Type your full name: John Doe  
// Your name is: John
```


Funciones con Strings

getline(): Capturar cadenas que el usuario ingrese.

push_back(): Es usada para insertar un carácter al final de la cadena.

pop_back(): Permite borrar el último carácter de un string.

```
int main() {  
    string str;  
  
    getline(& cin, & str); // almacena lo que el usuario ingrese  
    cout<<"El usuario ingreso: "<<str<<endl;  
  
    str.push_back( c: '!'); // Inserta el caracter ! al final  
    cout<<"El string después de insertar '!' es: "<<str<<endl;  
  
    str.pop_back(); // Elimina el último caracter del string  
    cout<<"El string después de eliminar el último carácter es: "<<str<<endl;  
  
    return 0;  
}
```

Funciones con Strings

capacity(): Esta función devuelve la capacidad asignada a la cadena.

resize(): Cambia el tamaño de un string, puede ser incrementado o decrementado.

shrink_to_fit(): Decremento la capacidad asignada según el tamaño mínimo del str.

```
int main() {  
    string str = "Las venas abiertas de america latina";  
  
    str.resize( n: 17); // Redimensiona el tamaño de str.  
    cout<<"Resize: "<<str<<endl;  
  
    cout<<"Capacidad: "<<str.capacity()<<endl; //Imprime la capacidad asignada.  
  
    cout<<"El tamaño de str es: "<<str.length()<<endl;  
    str.shrink_to_fit(); //Cambia la capacidad ajustando al tamaño del string.  
    cout<<"Capacidad nueva: "<<str.capacity();  
  
    return 0;  
}
```



Funciones con Strings

begin(): Devuelve un iterador al comienzo de la cadena.

end(): Devuelve un iterador al final de la cadena.

rbegin(): Devuelve un iterador inverso que apunta al final de la cadena.

rend(): Devuelve un iterador inverso que apunta al principio de la cadena.

```
int main() {  
    string str = "Las venas abiertas de america latina";  
  
    string::iterator it; //Declarando un iterador.  
    string::reverse_iterator iit; //Declarando un iterador inverso  
  
    cout<<"La cadena usando el iterador: ";  
    for(it=str.begin(); it!=str.end(); it++) cout<<*it;  
    cout<<endl;  
  
    cout<<"La cadena inversa usando el iterador inverso";  
    for(iit=str.rbegin(); iit!=str.rend(); iit++) cout<<*iit;  
    cout<<endl;  
  
    return 0;  
}
```

Funciones con Strings

copy(): Copia una subcadena en otra variable. (variable_destino, tamaño copiado, posición donde comienza la copia).

swap(): Intercambia una cadena con otra.

compare(): Permite comparar dos cadenas si son iguales o no.

```
int main() {  
    string str1 = "Las venas abiertas de america latina";  
    string str2 = "Arriba Villavicencio";  
    char ch[80];  
  
    str1.copy(ch, n: 18, pos: 0); //Usando copy(), se copia un substring a un char*  
    cout << "La nueva copia: " << ch << endl;  
  
    str1.swap( &: str2);  
    cout << "String 1: " << str1 << endl;  
    cout << "String 2: " << str2 << endl;  
  
    cout << ((str1.compare(str2) != 0) ? "No son iguales" : "Son iguales");  
  
    return 0;  
}
```

Funciones con Strings

find("string"): Devuelve la posición de la primera ocurrencia de la consulta.

rfind("string"): Devuelve la posición de la última ocurrencia de la consulta.

```
int main() {  
    string str = "El llano es llano, lo demas monatania";  
    string consulta = "llano";  
  
    cout << "Primera ocurrencia palabra 'llano' empieza en la posición: ";  
    cout << str.find(consulta) << endl;  
  
    cout << "Segunda ocurrencia palabra 'llano' empieza en la posición: ";  
    cout << str.rfind(consulta) << endl;  
  
    if(str.find(consulta) != string::npos)  
        cout << "la palabra consultada fue encontrada en en str";  
    else  
        cout << "la palabra consultada NO fue encontrada en en str";  
  
    return 0;  
}
```


Funciones con Strings

insert(pos_to_begin, string_to_insert): Inserta alguna cadena dentro de otra dada una posición.

clear(): Limpia todos los caracteres de un string.

empty(): Pregunta si un string está vacío o no.

```
int main() {  
    string str = "1280 Almas";  
  
    str.insert( pos: 5, s: "***** "); //Inserta *** a la cadena  
    cout<<str<<endl;  
  
    str.clear(); //Limpia la cadena de caracteres.  
    cout<<"Cadena limpia: "<<str<<endl;  
  
    if(str.empty() == 1) cout<<"Cadena está vacía."  
    return 0;  
}
```

Funciones con Strings

replace(): Reemplaza una porción de la cadena.

front(): Trae el primer carácter.

back(): Trae el último carácter.

```
int main() {  
    string str = "1280 Almas";  
  
    str.replace( pos: 5, n1: 3, s: "***"); //Inserta *** a la cadena  
    cout<<str<<endl;  
  
    cout<<"Primer carácter: "<<str.front()<<endl;  
    cout<<"Último carácter: "<<str.back();  
    return 0;  
}
```


Taller (Parte 1)

Resolver los siguientes ejercicios:

1. Escribe un programa que pida al usuario ingresar 5 números enteros (un solo input, cada número separados por coma) y encuentre la suma y el promedio de esos números

Ejemplo:

Introduzca los números: 1,2,3,4,5

Resultado esperado:

La suma de los números es: 15

EL promedio es: 3.0

2. Escribir un programa para determinar si una oración es un palíndromo o no.

Ejemplo:

Ingresa la frase: Anita lava la tina

ES UN PALÍNDROMO

Ingresa la frase: Nadie tiene un propósito

NO ES UN PALÍNDROMO

Bibliografía

- Cplusplus <http://www.cplusplus.com/reference/string/string/>
- Geek for geeks <https://www.geeksforgeeks.org/>
- w3schools https://www.w3schools.com/cpp/cpp_strings.asp

Gracias...