

Programación de Sistemas

Taller 4 – gcc y Make

Introducción

En este taller practicarán el uso de gcc y Make, y expandirán el programa desarrollado en el taller 3. Reorganizaremos nuestro programa y expandiremos su funcionalidad. Esto lo harán en grupos de dos personas. Implementaremos dos algoritmos más de cifrado: auto-llave y de contraseña.

El programa originalmente tenía dos modos de ejecución interactivo y no interactivo:

\$./prog

Cifrado cíclico

Ingrese mensaje a cifrar: hola pedro

Ingrese la llave numérica: 3

Mensaje cifrado: krod shgur

Mensaje cifrado en morse: -.- .-. --- -.. /- --. ...- .-.

\$./prog 3 "hola pedro"

Mensaje cifrado: krod shgur

Mensaje cifrado en morse: -.- .-. --- -.. /- --. ...- .-.

Ahora, el modo interactivo, pediremos también el tipo de cifrado con una letra: cíclico (C), auto-llave (A) o contraseña (P)

\$./prog

Tipo de cifrado: C

Cifrado Cíclico

Ingrese mensaje a cifrar: hola pedro

Ingrese la llave numérica: 3

Mensaje cifrado: krod shgur

Mensaje cifrado en morse: -.- .-. --- -.. /- --. ...- .-.

Autor: Eduardo Murillo

En el modo NO interactivo, podremos mandar un tercer argumento (-c, -a o -p) que nos dirá el tipo de cifrado a usar:

\$./prog 3 "hola pedro" -c

Mensaje cifrado: krod shgur

Mensaje cifrado en morse: -.- .-. --- -.. / --. ...- .-.

Parte 1

En esta parte, crearan una nueva rama en su repositorio de git llamada **v2.0**. Agregaran los siguientes archivos y directorios:

Directorios:

1. src
2. lib
3. include
4. obj
5. bin

Archivos:

1. src/main.c
2. src/cifrado.c
3. src/codificacion.c
4. include/cifrado.h
5. include/codificación.h

En cifrado.h declararán las siguientes funciones:

char* cifradoCiclico(char* mensaje, int llave);

char* cifradoAutollave(char* mensaje, char* llave);

char* cifradoContraseña(char* mensaje, char* llave);

En el archivo **cifrado.c** implementarán estas funciones. No olvide llamar al archivo cabecera necesaria. **La descripción de los cifrados está al final de este documento.**

En codificación.h tendrán la siguiente función:

char *claveMorse(char* mensaje);

Autor: Eduardo Murillo

En **codificación.c** implementarán la función `claveMorse`.

En el archivo **main.c** implementará el código de su programa principal y cualquiera otra función auxiliar que necesite (pero que no sea de cifrado ni de codificación). Este hará uso de las funciones definidas en `codificación.h` y `cifrado.h`. Una vez implementado, escriba las sentencias de gcc para:

1. Hacer de `cifrado.c` una **Librería compartida (.so)** y escribirla en la carpeta `lib/`
2. Generar el archivo objeto (.o) de `codificación.c` y escribirla en la carpeta `obj/`
3. Generar el archivo objeto (.o) de `main.c` y escribirla en la carpeta `obj/`
4. Generar el archivo ejecutable **cifrador** en la carpeta `bin/` (enlazando con los .o y .so)

Parte 2

Automatice el proceso de creación del ejecutable usando Make. Cree un archivo llamado **Makefile** en el directorio del proyecto. Su makefile debe estar creado de tal forma que al escribir:

\$ make

Se compilen los archivos y se enlace el ejecutable (usando los comandos de la parte 1).

Cree también una regla **clean** que permita borrar:

1. El ejecutable
2. Los archivos .o
3. Los archivos .so

De tal forma que se borren al hacer:

\$ make clean

Entregable

URL del repositorio de git. Fecha de entrega 26 de Mayo de 2017, 23:59. Tarea Individual

ANEXO

CIFRADO AUTO-LLAVE

Para esto hacemos uso de la llamada TABULA RECTA:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

El cifrado funciona de la siguiente manera: necesitamos una contraseña, y esta la anexamos al texto a cifrar, y luego alineamos el texto a cifrar con el nuevo string generado. Entonces, el par ordenado de cada posición se vuelve la coordenada (fila, columna) de la letra a usar de la tabula recta para cifrar cada letra de nuestro mensaje. Por ejemplo, con la contraseña **MARAZUL**, queremos cifrar el texto **HOLA PEDRO**

TEXTO:	HOLA PEDRO.....
CONTRASEÑA:	MARA ZULHOLAPEDRO
CIFRADO	TOCA OYOYC.....

Entonces el mensaje cifrado sería: **TOCA OYOYC**. Notar que respetamos los espacios.

Autor: Eduardo Murillo

CIFRADO AUTO-LLAVE

En este cifrado usamos también una contraseña, que determina las letras a usar. Por ejemplo, usando la contraseña MARAZUL, el alfabeto se convierte en:

Alfabeto	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Nuevo Alfabeto:	MARAZUL ABCDEFGHIJKLMNPOQR

Entonces, al encriptar un mensaje, todas las letras A se vuelven M, todas las B se vuelven A, todas las C se vuelven R, etc. Por ejemplo para encriptar “**HOLA PEDRO**” con la llave **MARAZUL**:

Sin encriptar:	HOLA PEDRO
Encriptado:	AHEM IZAKH