

## CLASE VIERNES 08 SEPTIEMBRE ASTROINFORMÁTICA

todo tiene q tener ; en C

la compilación la podemos dividir en 2 partes

1. preprocesado, optimizacion, semantica
2. ensamblado

librerías se descargan en directorios lib

.so son lib

librerías las podemos dividir en 2 tipos: estáticas y dinámicas

estáticas son las que terminan en .a , son el .c compilado transformado a lib .a de modo que cada vez que se use el linker toma el .a y lo mete todo dentro del ejecutable, estáticas genera librerías que no necesitan otras librerías, lo ejecuto y funciona

dinámicas (shared), son las .so o .dll, no son incorporadas en los ejecutables al momento del linkeo, por lo tanto cuando el programa se ejecuta necesariamente llama a la librería en el directorio donde estaba cuando se compiló

¿qué pasa cuando programo simulaciones?

compilación que no corre : seteo de dónde está la lib del sistema

sintaxis de C: preprocesador

preprocesador: no es parte del código, son instrucciones previas al compilado del código, que uno le da al programa para que tome el código fuente y lo tenga en una memoria temporal?

#include : lib propias es común

#define : para definir identificador (no variable pk estas están dentro del código), por ejemplo si pongo PI 3,1415, entonces si en el código pongo PI inmediatamente lo reemplaza por 3,14 ...

no necesita definir el tipo (str,float...)

condicionales

#ifdef, #ifndef, #endif, #else:

no son como condicionales de python, sirven para activar o desactivar bloques del código

los identificadores de procesador puede ir en cualquier parte, no necesariamente en el principio

siempre que uso #ifdef debo terminar antes del return con #endif

si pongo vim asdf.h en vez de .c creo una librería automáticamente

le cambio el main , librería no tiene main, en el int el pongo el nombre de mi librería con "" es una librería de este directorio

si es <> es una .so o .h que viene del sistema

-c solo compila no hace linker  
.o versiones en código máquina  
si compilo por separado con -c creo un .o  
luego si compilo junto -o y pongo \*.o me compila todos los .o que se crearon

#define SAD

se puede compilar y linkear x separado : -c  
dinámicas son las .so , las que causan el problema porque cuando uno compila  
tiene que decirle dónde está, y lo tiene q saber el compilador el linker y el  
programa ???  
el .h no se compila, por eso se crea el .gch  
float y double, float clásico flotante que se restringe a tener 4 bites, entero normal  
sólo admite 20 millones de números.  
C incorpora  
C no definió strings, armaron una librería  
los complicados son string y FILES  
todas las variables pasan por 2 procesos: la declaración y la inicialización  
%d es para meter un entero en el print  
RAM random access memory  
C permite acceder a la RAM sin gastar bites

hackear C, accedo a memoria que no puedo acceder