Practico VM - SistOp

Lautaro Bachmann

Contents

16) **7**

1

Como resolver

En el stack se almacenan las variables normales

En el heap se almacena el contenido dinamico

Las variables globales se almacenan en un segmento especial.

En la vida real los programas tienen mas segmentos que code, stack, heap

Resolucion

variable	Stack	Heap	Comentario
i	X		
s	X		
b	X	X	b se guarda en el stack pero lo que apunta se encuentra en el heap

2)

a)

gets está deprecado porque es inseguro con respecto a memoria

b)

malloc solo está alocando una cantidad de bytes sin tener en cuenta cuantos bytes ocupa cada char del string

c)

strdup aloca memoria, por lo cual se pisa el valor de d ## d) No se tiene en cuenta el tamaño de lo que se está alocando

3)

a)

Falso. malloc pertenece a una libreria

b)

Falso. Puede que el espacio disponible sea suficiente y no haga falta llamar a una syscall

c)

Verdadero. malloc llama a brk o sbrk

d)

Verdadero.

e)

Falso. No depende del tamaño de memoria que se va a alocar

4)

0 -> 4096 5 -> 4096 + 5 128 -> 4096 + 128 8 -> 4096 + 8 10 -> 4096 + 10 256 -> 4096 + 256 => segmentation fault 13 -> 4096 + 13

5) (INC)

6)

Estatica es cuando se parchea el codigo con las direcciones adecuadas. Se realiza unicamente con software y no ofrece ningun tipo de proteccion.

Dinamico es cuando se usan registros base and bounds, hay apoyo de software y hay proteccion.

7)

a)

Verdadero. De no ser así seria una posible vulnerabilidad de la seguridad del SO

b)

Falso. Puede haber una infinita cantidad de procesos (segun los recursos lo permitan), pero no infinita cantidad de registros

8)

Paginas de 4KiB

espacio de direcciones = 65536 bytes = 64 KiB -> 16 paginas codigo = 32 KiB -> 8 paginas heap = 16.001 KiB -> 5 paginas stack = 15.49 KiB -> 4 paginas

paginas necesarias = 8 + 5 + 4 = 17

Como el programa necesit de 17 paginas y el espacio de direcciones solo tiene espacio para 16, sabemos que el programa no cabe

Paginas de 512B

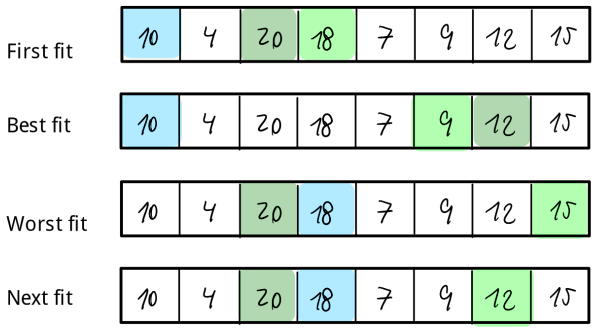
espacio de direcciones = 65536 bytes = 64 KiB -> 128 paginas codigo = 32 KiB -> 64 paginas heap = 16.001 KiB -> 33 paginas stack = 15.49 KiB -> 31 paginas

paginas necesarias = 64 + 33 + 31 = 128

Como la cantidad de paginas necesarias es igual a la cantidad de paginas disponibles sabemos que el programa cabe

9)

Ejercicio 9. Suponga un sistema de memoria contiguo con la siguiente secuencia de tamaños de huecos: 10 KiB, 4 KiB, 20 KiB, 18 KiB, 7 KiB, 9 KiB, 12 KiB, 15 KiB.
Para la siguiente **secuencia de solicitudes** de segmentos de memoria: 12 KiB, 10 KiB, 9 KiB.
¿Cuáles huecos se toman para las distintas políticas?



10)

(10 + 120) * 0.95 + (10 + 120 + 120) * 0.05 = 136ns

11)

Howto

Pensar que valor hace falta para que cada item agregado al array esté en una página distinta

Pensar la pregunta del (a) como un ∀

12)

Howto

Dividir direccion por tamaño de pagina, el cociente es la VPN y el resto es el offset

13)

14)

15)

16)