

Sistemas Operativos

Ejercicio 1. (1 pts)

1. El SO tiene dos funciones principales. Identifíquelas y explíquelas (3 reng. por cada una)
2. ¿Para que sirven las system calls (llamadas al sistema)? (1 reng)

1)

Gestión de recursos: Administra y asigna los recursos del hardware, como la CPU, la memoria, el almacenamiento y los dispositivos de entrada/salida, asegurando que cada aplicación tenga acceso adecuado a estos recursos sin conflictos.

Interfaz de usuario: Proporciona una interfaz que permite a los usuarios interactuar con el hardware y el software, ya sea a través de una interfaz gráfica (GUI) o de línea de comandos (CLI), facilitando la operación del sistema y la ejecución de programas.

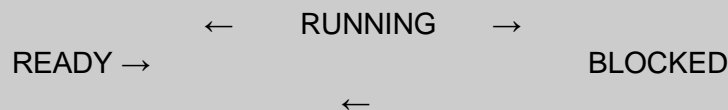
2)

Una *system call* (o llamada al sistema) es una interfaz que permite a los programas de usuario solicitar servicios del sistema operativo. A través de estas llamadas, los programas pueden realizar operaciones como.

Ejercicio 2. (2 pts)

1. Grafique el diagrama de estados de un proceso. Para cada arco, muestre un ejemplo donde se produce esta transición entre estados. (1 reng. por transición)
2. El diagrama que acaba de dar, ¿a que tipo de planificador corresponde? Justifique su respuesta. (2 reng.)
3. ¿Cual es la diferencia entre un programa, un proceso, y un hilo? (2 reng)

1)



Running → Blocked = Por un I/O o consultar el disco.

Blocked → Ready = Terminó el I/O.

Ready → Running = Comienza a ejecutar un programa.

Running → Ready = Desplanificación, context switch.

2)

Es un planificador apropiativo.

3)

Ejercicio 3. (2.5 pts)
 Asuma los siguientes procesos Comprador y Vendedor.

Estado inicial: { saldo_comprador = 200, saldo_vendedor = 0, deuda=0 }	
Comprador While true do If (deuda==0 AND saldo_comprador >= 100) then deuda:=100 saldo_comprador:=saldo_comprador-deuda endif end	Vendedor While true do if deuda>0 then saldo_vendedor:=saldo_vendedor+deuda deuda:=0 endif end

i. El objetivo es que al final de la ejecución, exactamente dos compras se hayan ejecutado, llegando al estado { saldo_vendedor = 200 y saldo_comprador = 0}. Sin embargo, existe una condición de carrera donde se llega a resultados diferentes. Identifique estas situaciones, listando a los valores posibles a los que pueden llegar las variables saldo_comprador y saldo_vendedor.

ii. Sincronice los procesos (por ejemplo, usando semáforos mutex) para prevenir la condición de carrera.

Proceso	Arribo	Uso CPU	Inicio	Fin	M	
A	0	8				
B	3	5				
C	5	7				
D	0	3				
E	9	5				

Figura 1

	C					R			
P0	0	1	1	2	1	1	0	0	
P1	1	0	0	0	0	7	5	0	
P2	1	3	5	4	1	0	0	2	
P3	0	6	3	2	0	0	2	0	
P4	0	0	1	4	0	6	3	2	

A = 1 5 2 0

Figura 2

1. Completar la tabla de planificación de la Figura 1 para las políticas: SJF, SRTN, RR(Q=1), RR(Q=5). Para cada caso haga el diagrama de planificación y compute el tiempo promedio de espera M .

CPU	D	D	D	B	B	B	B	B
READY	A	A	A	A	A	AC	AC	AC

ARRIBO	AD			B		C		
TIEMPO	0	1	2	3	4	5	6	7

C	E	E	E	E	E	A	A	A
AE	A	A	A	A	A			
14	15	16	17	18	19	20	21	22

TABLA FINAL =

Proceso	Arribo	Uso cpu	InicioSJF	Fin SJF	M SJF
A	0	8	20	27	
B	3	5	3	7	
C	5	7	8	14	
D	0	3	0	2	
E	9	5	15	19	

STCF

CPU	D	D	D	B	B	B	B	B
READY	A	A	A	A	A	AC	AC	AC
ARRIBO	AD			B		C		
TIEMPO	0	1	2	3	4	5	6	7

C	C	C	C	C	C	A	A	A
14	15	16	17	18	19	20	21	22

TABLA FINAL =

Proceso	Arribo	Uso cpu	In.STC	Fin STC	M STC
A	0	8	20	27	
B	3	5	3	7	
C	5	7	14	19	
D	0	3	0	2	
E	9	5	9	13	

RR (Q=1)

CPU	A	D	A	D	B	A	D//	B
READY	D							
COLA		A	D	BA	AD	DBC	BCA	CA
ARRIBO	AD			B		C		
TIEMPO	0	1	2	3	4	5	6	7

B	C	E	A	B///	C	E	A//	C
CEA	EAB	ABC	BCE	CEA	EA	AC	CE	E

14	15	16	17	18	19	20	21	22

TABLA FINAL =

Proceso	Arribo	Uso cpu	In.RR	Fin RR	M RR
A	0	8	0	17	
B	3	5	4	18	
C	5	7	8	26	
D	0	3	1	6	
E	9	5	12	25	

RR (Q=2)

CPU	A	A	D	D	A	A	B	B
READY								
COLA	D	D	A	AB	BD	BDC	DCA	DCA
ARRIBO	AD			B		C		
TIEMPO	0	1	2	3	4	5	6	7

B	E	E	C	C	A	A//	B//	E
ECA	CAB	CAB	ABE	ABE	BEC	BEC	EC	C
14	15	16	17	18	19	20	21	22

TABLA FINAL =

Proceso	Arribo	Uso cpu	In.RR	Fin RR	M RR
A	0	8	0	20	
B	3	5	6	21	
C	5	7	9	27	
D	0	3	2	8	
E	9	5	15	26	

Ejercicio 5. (2.5 pts)

- Se tiene un Estado de Asignación de Recursos (EAR) con 5 procesos y 4 clases de recursos con los valores como lo establece la Figura 2. Indique si es seguro, y en caso de serlo muestre la secuencia de ejecución serial que hace finalizar los procesos, indicando el cambio de disponibles (columna 'A') luego de que cada proceso termina. (C=Current, R=Maximum required, A=Available)
- Si se tienen n procesos, y m recursos, y cada proceso puede pedir a lo sumo k recursos,
 - ¿cual es la relación que se tiene que cumplir para que no haya deadlocks?
 - Suponga que $k=m$. De una secuencia de ejecución que prevenga deadlocks.

	C				R			
P0	0	1	1	2	1	1	0	0
P1	1	0	0	0	0	7	5	0
P2	1	3	5	4	1	0	0	2
P3	0	6	3	2	0	0	2	0
P4	0	0	1	4	0	6	3	2
	A = 1 5 2 0							
Figura 2								

Los semáforos no entran.

Ejercicio 6. (Sólo para libres) ¿Cuál es el objetivo del paginado de memoria? ¿Cuál es el objetivo de utilizar segmentación?

El objetivo del paginado de memoria es poder disminuir el consumo de memoria y evitar la fragmentación de la misma dividiendo el espacio de direcciones en páginas de tamaños fijos y mapeando en memoria las páginas válidas.

El objetivo de la segmentación es evitar la fragmentación interna de los procesos debido al enorme hueco entre el heap y el stack, se divide en 3 segmentos lógicos (CODE, STACK, HEAP) con 3 pares de registros base and bounds

Ejercicio 7. Considere un tamaño de página de $4KB = 2^{12}$ bytes y la tabla de paginado de la Fig. 3.

- (a) Determine las direcciones físicas de las direcciones virtuales 46643 y 36818.
(b) Determine la dirección virtual correspondiente a la dirección física 12033.

V	F	P/A	V	F	P/A
0	000	0	8	000	0
1	000	0	9	000	0
2	000	0	10	001	1
3	100	1	11	101	1
4	111	1	12	000	1
5	010	1	13	000	0
6	000	0	14	110	1
7	000	0	15	011	1

Ejercicio 8. Un computador tiene cuatro páginas. El tiempo

a)

Son 4 bits de VPN ya que tenemos 16 entradas en la page table.

$$46643 / 2 = 23321 / 2 = 11660 / 2 = 5830 / 2 = 2915 / 2 = 1457 / 2 = 728 / 2 = 364 / 2 = 182 / 2 =$$

0

1 1 0 0 1 1 0 0

$$91 / 2 = 45 / 2 = 22 / 2 = 11 / 2 = 5 / 2 = 2 / 2 = 1$$

1 1 0 1 1 0

$$46643 = 0b \text{ 1011 011000110011}$$

$$VPN = 2^0 + 2^1 + 2^3 = 11$$

$$PFN = 0101 \ 011000110011$$

$$36818 / 2 = 18409 / 2 = 9204 / 2 = 4602 / 2 = 2301 / 2 = 1150 / 2 = 575 / 2 = 287 / 2 = 143 / 2 =$$

1

0 1 0 0 1 0 1 1

$$71 / 2 = 35 / 2 = 17 / 2 = 8 / 2 = 4 / 2 = 2 / 2 = 1$$

1 1 1 0 0 0

$$36818 = 0b \ 1000 \ 111111010010 \text{ INVALIDO}$$

b)

$$12033 / 2 = 6016 / 2 = 3008 / 2 = 1504 / 2 = 752 / 2 = 376 / 2 = 188 / 2 = 94 / 2 = 47 / 2 = 23 / 2$$

1 0 0 0 0 0 0 1

$$11 / 2 = 5 / 2 = 2 / 2 = 1$$

1 1 0

12033 = 0b 0010 111100000001

VPN correspondientes a 00100 es 5 luego

V.Address = 0101 111100000001

Ejercicio 8. Un computador tiene cuatro páginas. El tiempo de carga, el último momento de accesos, y los bits R y M de cada página son como se especifica en la Fig. 4. Establezca que página remplazarán los siguientes algoritmos: NRU, FIFO, LRU, "Second Chance".

Pág.	T Carga	T Ult Acc	R	M
0	584	1319	1	1
1	1080	1247	0	1
2	659	1278	0	0
3	518	1341	1	0

Figura 4:

NO SE NO ENTIENDO

Ejercicio 9. En un sistema hay n usuarios u_1, \dots, u_n , m procesos p_1, \dots, p_m , y l archivos a_1, \dots, a_l . Suponga que se desea controlar el acceso a los archivos mediante (A) listas de control de acceso (ACL) o bien con (B) listas de capacidades.

2	659	1278	0	0
3	518	1341	1	0

Figura 4:

- (I) Si un sistema tiene n (la cantidad de usuarios) extremadamente grande, pero m y l chico, ¿qué conviene utilizar, ACLs o capacidades? Justifique.
- (II) Suponiendo que los permisos de acceso dependen selectivamente de los usuarios y deben ser a menudo revocados, ¿qué método de control de acceso conviene utilizar? Justifique.
- (III) Suponiendo que los permisos no son usualmente revocados, y que los pedidos de acceso son del tipo "El proceso p_i requiere acceso al archivo a_j ", ¿qué es más eficiente, utilizar ACL o capacidades? Justifique.
- (IV) El acceso de "lectura" ¿a qué propiedad de seguridad se refiere: confidencialidad o integridad?. ¿Y el de "escritura"? ¿Y el de "ejecución"? Justifique.

NO LO TOMAN

Ejercicio 10. Suponga que tiene una partición del disco rígido con capacidad de 128 GB dividida en bloques de 4 KB. ¿Cuántos bloques de disco necesita exactamente para almacenar el bitmap que indica qué bloques libres del disco?

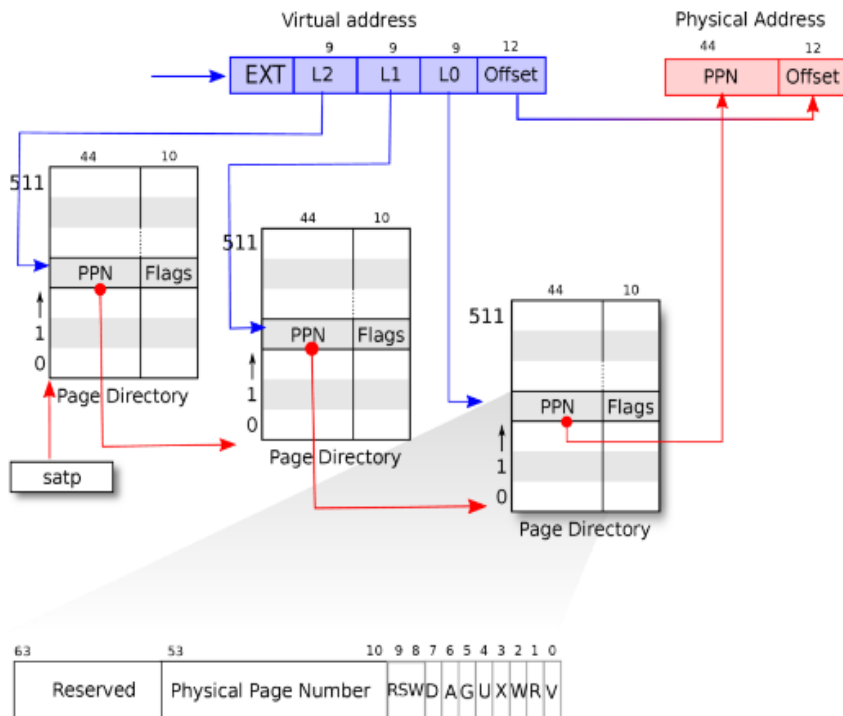
$128\text{GB} / 4\text{KB} = 2^7 * 2^{30} / 2^{12} = 2^{25} = 33554432$ de bloques en donde cada uno tendrá el bitmap indicando si está libre.

Ejercicio 11. La E/S puede efectuarse de dos maneras distintos según se mapee a memoria o no. Describa brevemente cada una. Dé 2 ventajas del mapeo de memoria y 2 desventajas.

NO SE asjda

Ejercicio 4

Tenemos un esquema de paginación RISC-V con páginas de 4 KiB de 3 niveles con formato 9,9,9,12 -> 44,12 como muestra la figura.



Bits de control

V: *válido*

R: se puede leer, *readable*

W: se puede escribir, *writable*

X: se puede ejecutar, *executable*

Supongamos que tenemos el registro de paginación apuntando al marco físico satp=0x0000000FE0.

0x0000000FE0	0x0000000FEA	0x00000AD0BE
-----	-----	-----
0x1FF: 0x0000000000, ----	0x1FF: 0x0000000000, ----	0x1FF: 0x0000000000, ----
⋮	⋮	⋮
0x004: 0x0000000000, ----	0x004: 0x0000000000, ----	0x004: 0x0000000000, ----
0x003: 0x0000000000, ----	0x003: 0x0000000000, ----	0x003: 0x0000D1AB10, XWR-
0x002: 0x0000000FEA, XWRV	0x002: 0x00000AD0BE, XWRV	0x002: 0x0000DECADA, -WRV
0x001: 0x0000000FEA, XWRV	0x001: 0x00000AD0BE, XWRV	0x001: 0x000CAFECAFE, ----
0x000: 0x0000000FEA, XWRV	0x000: 0x00000AD0BE, XWRV	0x000: 0x000000ABAD, X--V

a) Traducir de **virtual a física** las direcciones:

0x0000	0x 000 0000 0000 0000 0000 0000 0000 0000 PD1 = 0 → (0000000FEA) PD2 = 0 → (00000AD0BE) PD3 = 0 → (000000ABAD) OFFSET = 0 → 000000ABAD000
0x1000	0x1000 = 0x 000 0000 0000 0000 0000 0000 0001 0000 0000 0000 PD1 = 0 → (0000000FEA) PD2 = 0 → (00000AD0BE) PD3 = 1 → (00CAFECAFE) INVALIDA
0x2000	0x2000 = 0x 000 0000 0000 0000 0000 0000 0010 0000 0000 0000 PD1 = 0 → (0000000FEA) PD2 = 0 → (00000AD0BE) PD3 = 2 → (0000DECADA) OFFSET = 0 -> 0000DECADA000
0x3000	0x3000 = 0x 000 0000 0000 0000 0000 0000 0011 0000 0000 0000 PD1 = 0 → (0000000FEA) PD2 = 0 → (00000AD0BE) PD3 = 3 → (0000D1AB10) INVALIDA

$8192 / 2^{12} = 2 \rightarrow \text{VPN}$ (contiene el PDI, el PT0 y el PT1)

Sacamos el offset

$8192 - 4096 * 2 = 0 \rightarrow \text{OFFSET}$

extendemos a los 44 bits el VPN

00000000 00000000 000000010
PDI PT0 PT1

indexando la VPN $\rightarrow \text{PFN} = 0x00000DECADA(00....00) = 59.805.376.512$
44 bits 12 bits

sumamos el PFN + OFFSET = $59.805.376.512 + 0 = 59.805.376.512 \rightarrow 0xDECADA000$