Página Principal / Mis cursos / SistOp20 / Prácticos / Parcial 1: Virtualización		
Comenzado el	Tuesday, 20 de October de 2020, 14:00	
Estado	Finalizado	
Finalizado en	Tuesday, 20 de October de 2020, 15:59	
Tiempo empleado	1 hora 59 minutos	
Calificación	6,25 de 10,00 (63 %)	
Pregunta 1		
Finalizado		
Sin calificar		

El fork es:

Seleccione una:

- a. Ricky.
- b. Una syscall de UNIX que crea un proceso con la misma imagen de memoria pero en un espacio disjunto.
- c. Tenedor.

Respuesta correcta

La respuesta correcta es: Una syscall de UNIX que crea un proceso con la misma imagen de memoria pero en un espacio disjunto.

```
Pregunta 2
Sin contestar
Puntúa como 2,00
```

Se cambió ligeramente el trapframe donde se mantiene el estado de los registros dentro del process control block de **xv6** (1ra columna). Dada la función mcd() (2da columna), se la compila a i386 con -00 (3ra columna) y con -01 (4ta columna).

La opción -00 significa sin optimizaciones.

La opción -01 significa con optimizaciones básicas.

Para las dos versiones en ensablador, diga si el context switch funciona siempre, a veces o nunca.

```
struct trapframe { | int mcd(int a, int b)
                                             mcd:
                                                                         mcd:
  uint edi;
                    {
                                                                              movl 4(%esp), %edx
                                                 jmp L2
                      while(a!=b) {
                                             L4:
                                                                              mov1 8(%esp), %eax
  uint esi;
  uint ebp;
                         if (a<b)
                                                 movl 4(%esp), %eax
                                                                              cmpl %eax, %edx
  uint oesp;
                          b = b - a;
                                                 cmpl 8(%esp), %eax
                                                                              jne L5
                                                                         L2:
 uint ebx;
                         else
                                                 jge L3
 uint ecx;
                                                 movl 4(%esp), %eax
                           a = a - b;
                                                                              ret
  uint eax;
                      }
                                                 subl %eax, 8(%esp)
                                                                         L3:
  uint gs;
                      return a;
                                                 jmp L2
                                                                              subl %eax, %edx
                    }
                                             L3:
  uint fs;
                                                                         L4:
  uint es;
                                                 movl 8(%esp), %eax
                                                                              cmpl %eax, %edx
 uint ds;
                                                 subl %eax, 4(%esp)
                                                                              je L2
 uint trapno;
                                             L2:
                                                                         L5:
 uint err;
                                                 movl 4(%esp), %eax
                                                                              cmpl %eax, %edx
                                                 cmpl 8(%esp), %eax
  uint eip;
                                                                              jge L3
  uint cs;
                                                 jne L4
                                                                              subl %edx, %eax
  uint eflags;
                                                 movl 4(%esp), %eax
                                                                              jmp L4
  uint esp;
                                                 ret
  uint ss;
}
```

Seleccione una o más de una:

- a. Con -00 (3ra columna): Siempre.
- b. Con -oo (3ra columna): A veces.
- c. Con -00 (3ra columna): Nunca.
- d. Con -01 (4ta columna): Siempre.
- e. Con -01 (4ta columna): A veces.
- f. Con -01 (4ta columna): Nunca.

Respuesta incorrecta.

Las respuestas correctas son: Con -00 (3ra columna): Siempre., Con -01 (4ta columna): A veces.

Pregunta **3**

Correcta

Puntúa 1,00 sobre 1,00

Para los siguientes procesos CPU-bound completar la secuencia temporal de como se planifican los procesos en un monoprocesador para una política de planificación RR con *quanto* de 2.

Proceso	$T_{arrival}$	T_{CPU}
A	0	4
В	1	2

Ejemplo de respuesta: AACCAAABB.

Respuesta:	AABBAA	~
------------	--------	----------

La respuesta correcta es: AABBAA

Pregunta 4

Correcta

Puntúa 1,00 sobre 1,00

Para los siguientes procesos CPU-bound completar la secuencia temporal de como se planifican los procesos en un monoprocesador para una política de planificación STCF.

Proceso	$T_{arrival}$	T_{CPU}
A	0	4
В	1	2
$^{\mathrm{C}}$	3	1
D	3	4

Ejemplo de la respuesta: ABBBADDD.

Respuesta:	ABBCAAADDDD	~
------------	-------------	---

La respuesta correcta es: ABBCAAADDDD

2021	Parcial 1: Virtualización: Revisión del intento
Pregunta 5	
Correcta	
Puntúa 0,50 sobre 0,50	
siguiente secuencia de acc	assembler i386 usando memoria segmentada se produce la esos a la memoria física: 519, 524, 2044, 527, 529, SegFault, 532. o de segmentación base para el segmento de código. ✓
Pregunta 6	
Correcta	
Puntúa 0,50 sobre 0,50	
siguiente secuencia de acc	assembler i386 usando memoria segmentada se produce la esos a la memoria física: 519, 524, 2044, 527, 529, SegFault, 532. o de segmentación base para el segmento de heap .
Respuesta: 1024	✓

La respuesta correcta es: 1024

Pregunta 7		
Sin contestar		
Puntúa como 0,50		

Al ejecutar este programa *assembler* i386 usando **memoria segmentada** se produce la siguiente secuencia de accesos a la memoria física: 519, 524, 2044, 527, 529, SegFault, 532. Deducir el valor del registro de segmentación **límite** para el **segmento de heap**.

	7: movl \$1020,%ebx	
1	2: movl (%ebx),%eax	
1	3: addl \$4, %ebx	
1	7: movl (%ebx),%eax	
2): retq	

Respuesta:

La respuesta correcta es: 1024

Pregunta **8**Incorrecta
Puntúa 0,00 sobre 0,50

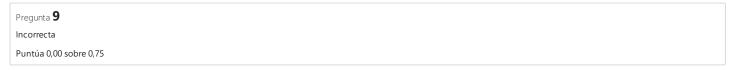
Considere un tamaño de página de 8 KiB (8192 bytes) y la tabla de páginas lineal de abajo. Determine que dirección física que se corresponde a la dirección virtual 24579. Expresar en decimal.

Consejo: no pasar a binario, dividir por el pagesize.

```
VPN | PFN | ¿Presente?
15 | 000 | 0
14
   | 111 | 1
13
   | 000 | 0
12
   | 000 | 0
11
   | 001 | 1
10
   | 101 | 1
   | 000 | 0
   | 100 | 1
   | 000 | 0
   | 000 | 0
   | 011 | 1
   | 000 | 0
3
   | 100 | 1
   | 000 | 1
2
   | 000 | 0
   | 110 | 1
```

Respuesta: 3

La respuesta correcta es: 32771



Considere un tamaño de página de 8 KiB (8192 bytes) y la tabla de páginas lineal de abajo. Determine **alguna** dirección virtual que se corresponde a la dirección física 32772.

Expresar en decimal.

Consejo: no pasar a binario, dividir por el pagesize.

```
VPN | PFN | ¿Presente?
15 | 000 | 0
14 | 111 | 1
13 | 000 | 0
12 | 000 | 0
11 | 001 | 1
10 | 101 | 1
9 | 000 | 0
8 | 100 | 1
7 | 000 | 0
 6 | 000 | 0
   | 011 | 1
   | 000 | 0
   | 100 | 1
 2
   | 000 | 1
 1 | 000 | 0
   | 110 | 1
```

Respuesta:	4	×
------------	---	---

La respuesta correcta es: 24580

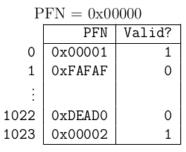
Pregunta 10

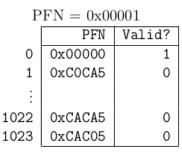
Correcta

Puntúa 0,75 sobre 0,75

Para un esquema de paginación de i386 (10,10,12), con PDBR=CR3=0x00000 y dada la siguiente configuración de memoria, transformar de virtual a física la dirección 0xFFFFF0C0.

Expresar en hexadecimal y con todos los dígitos. Ejemplo: 0x0023ABCD.





Respuesta:

0x0000C0C0

La respuesta correcta es: 0x0000C0C0

Pregunta 11

Correcta

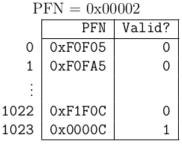
Puntúa 0,75 sobre 0,75

Para un esquema de paginación de i386 (10,10,12), con PDBR=CR3=0x00000 y dada la siguiente configuración de memoria, transformar de virtual a física la dirección **0xFFFFFACA**.

Expresar en hexadecimal y con todos los dígitos. Ejemplo: 0x0023ABCD.

PFN = 0x000000			
	PFN	Valid?	
0	0x00001	1	
1	OxFAFAF	0	
:			
1022	0xDEAD0	0	
1023	0x00002	1	

PFN = 0x00001		
	PFN	Valid?
0	0x00000	1
1	0xC0CA5	0
:		
1022	0xCACA5	0
1023	0xCAC05	0



Respuesta:

0x0000CACA

La respuesta correcta es: 0x0000CACA

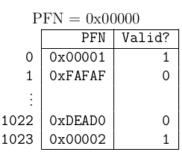
Pregunta 12

Correcta

Puntúa 0,75 sobre 0,75

Para un esquema de paginación de i386 (10,10,12), con PDBR=CR3=0x00000 y dada la siguiente configuración de memoria, transformar de virtual a física la dirección 0x00400BA4.

Expresar en hexadecimal y con todos los dígitos. Ejemplo: 0x0023ABCD.



PFN = 0x00001			
	PFN	Valid?	
0	0x00000	1	
1	0xC0CA5	0	
:			
1022	0xCACA5	0	
1023	0xCAC05	0	

PFN = 0x00002		
	PFN	Valid?
0	0xF0F05	0
1	0xF0FA5	0
:		
1022	0xF1F0C	0
1023	0x0000C	1

Respuesta: Pagfault

La respuesta correcta es: fault

Pregunta 13

Correcta

Puntúa 1,00 sobre 1,00

Para un esquema de paginación de i386 (10,10,12), con PDBR=CR3=0x00000 y dada la siguiente configuración de memoria, transformar de física a virtual la dirección 0x0000CAAA.

Expresar en hexadecimal y con todos los dígitos. Ejemplo: 0x0023ABCD.

$$\begin{array}{c|cccc} PFN = 0x00000 \\ \hline & & PFN & Valid? \\ 0 & 0x00001 & 1 \\ 1 & 0xFAFAF & 0 \\ \vdots & & & \\ 1022 & 0xDEAD0 & 0 \\ 1023 & 0x00002 & 1 \\ \end{array}$$

Respuesta: 0xFFFFAAA

La respuesta correcta es: FFFFAAA

▼ Entrenamiento de Virtualización

Concurrencia ►