



F.I.U.B.A.

75-08 Sistemas Operativos

Final 20/12/2007

Prof. Lic. Ing. Osvaldo Clúa

Nota: Las notas se publicarán por cartelera y en la página


<http://www.fi.uba.ar/materias/7508/Teoria-2007/Final-Dic-07-Feb-08.html>

encriptadas con la *passwd*

1. Ver y resolver en la hoja adjunta.
2. Describa con ejemplos la implementación de las distintas opciones de resolución de llamados a biblioteca según se hagan en tiempo de compilación, carga o ejecución. (Puede usar Linux o Windows para los ejemplos)
3. Un programa fue diseñado para usar varios *threads*. ¿Cómo puede ser (si está bien diseñado) su performance respecto a una versión con un único thread? (Mejora, empeora, no varía,... justifique)
 - a) Si el sistema implementa *threads* a nivel de usuario.
 - b) Si el sistema implementa *threads* a nivel del Kernel.
4. Discuta como asignaría prioridades a programas
 - a) con mucho uso de I/O,
 - b) con mucho uso de CPU y
 - c) programas que interactúan mucho con el usuario.

Si su sistema admite *swapping* y Ud pudiera indicar cuales pueden ser *swappeados*, a cual de ellos marcaría como *swappeable* y porqué.

5. Para cada afirmación. Indique si es verdadera o falsa y justifique:
 - a) Un proceso en estado de blocked puede pasar a estado de wait/ready.
 - b) Los threads no comparten el espacio de direcciones.
 - c) Los procesos no comparten el espacio de direcciones.
 - d) Cada proceso trata sus propias interrupciones.

 F.I.U.B.A.	75-08 Sistemas Operativos Final 20/12/2007 Prof. Lic. Ing. Osvaldo Clúa		1	
			2	
			3	
	Padrón: _____ Apellido y Nombres: _____		4	
	_____		5	

1 Usando como ejemplo el código del programa *exec01.cc* que sigue,?

```

1. #include <iostream>
2. using namespace std;
3. int main(){
4.     cout<<"Programa exec01"<<endl;
5.     // setenv(VAR,VALOR,OVERWRITE) da a la variable de
6.     // ambiente VAR el valor VALOR. Si overwrite !=0
7.     // la modifica si existiera.
8.     setenv("VAR2","VE_MAIN",1);
9.     execl("./prog","./prog","paruno",NULL);
10.    cout <<"Programa exec01,VAR1="<<
11.        getenv("VAR1")<<endl;
12.    cout <<"Programa exec01,VAR2="<<
13.        getenv("VAR2")<<endl;
14. }
```

Donde el programa *prog.cc* es:

```

1 #include <iostream>
2 using namespace std;
3 int main(int argc,char* argv[]){
4 // char* getenv(VARIABLE) obtiene el valor de VARIABLE del ambiente
5 cout <<"Este es prog1 argumento="<<argv[1]<<" VAR1="<<
6     getenv("VAR1")<<" VAR2="<<getenv("VAR2");
7 cout<<endl;
8     setenv("VAR2","VE_PROG",1);
9 }
```

y se lo invoca con las siguientes líneas de *shell*:

```

1. $ export VAR1= uno
2. VAR2= dos
3. $ ./execl
4. $ echo shell, Var1=$VAR1
5. $ echo shell, Var2=$VAR2
```

indicar en el siguiente formulario que imprime.

a) Impresión *exec01*

Línea 4: *programa exec01*

Línea 10:

Línea 12:

b) Impresión *prog*

Línea 5

c) Impresión *shell*

Línea 4

Línea 5