



**FIUBA**

# 75-08 Sistemas Operativos 2004 Threads

Ing. Osvaldo Clúa

Esta clase completa la clase anterior (procesos). Además Ud. debe bajar un archivo con código ejemplo.

1. ¿Cómo nace un proceso en un sistema operativo batch y en uno de *TimeSharing* como UNIX?
2. ¿Cuál es la función de *exec()* y *fork()* como *system call* de UNIX?
3. Explique brevemente que hacen los siguientes *system calls*: *getpid()*; *wait()*; *waitpid()*.
4. ¿Qué es un proceso *daemon*?
5. Explique usando los ejemplos del código, la forma de acceso al ambiente y a otras variables de la *u\_area*.
6. Explique el uso de las rutinas de terminación como la vista en *sali.cc*.
7. ¿Qué son las *threads*? Describa sus variantes de implementación.
8. ¿Qué campos de los mencionados en 3 pasan al *Thread Control Block*?
9. ¿Qué opciones hay para la interacción del *scheduler* con las *threads*.
10. Deduzca el modelo usado por *pthreads* en UNIX y el modelo usado por PERL (observe los ejemplos del código).
11. Analice el ejemplo que permite el acceso simultáneo a variables por parte de las *threads* y el similar hecho con *fork*.
12. ¿Cómo se inserta el Modelo de Objetos en esta evolución?
13. Ensaye una justificación tecnológica del uso de multiprogramación y multithreading.
14. ¿Cómo impacta el multiprocesamiento en las operaciones de *scheduling*, *dispatching* y *conmutación de contexto*?
15. ¿Cómo es la técnica de programación *seudoconversacional*, *transaccional* o usando *autómatas finitos* (son distintos nombres de la misma técnica).

Todos los códigos hechos en C/C++ pueden hacerse con *Perl* los resultados son distintos solo en el caso del uso de *threads*. Parece una buena forma de repasar para el final hacerlos ....