

**Universidad Nacional Autónoma de México**  
**Facultad de Ciencias**  
**Sistemas Operativos**

Méndez Ávila Luis Giovanni

317143980

**Análisis del planificador de Minix**

- **Schedule.c (src/servers/sched)**
  - **do\_start\_schedulling**

Inicia la planificación de un proceso específico. Establece la prioridad máxima del proceso, su quantum de tiempo y otros atributos relacionados con la planificación. Asigna una prioridad y un quantum de tiempo.
  - **do\_stop\_schedulling**

Detiene la planificación para un proceso específico. Libera los recursos asociados con el proceso.
  - **do\_noquantum**

Se invoca cuando un proceso se queda sin quantum. Incrementa la prioridad del proceso en una unidad y vuelve a programarlo para que se ejecute.
  - **do\_nice**

Se utiliza para cambiar el nivel de prioridad de un proceso
  - **schedule\_process**

Se encarga de establecer la prioridad, el quantum de tiempo y la CPU en la que debe ejecutarse el proceso.
  - **balance\_queues**

Se ejecuta periódicamente para equilibrar las colas de planificación
- **Schedproc.h**
  - **struct schedproc:** Esta estructura define la información de planificación para cada proceso en el sistema. Tiene atributos como endpoint, que es el identificador del proceso, parent, que es el identificador del proceso padre, flags para indicar varios estados del proceso, y varios atributos relacionados con la planificación del proceso, como max\_priority, priority, time\_slice, cpu, y cpu\_mask. Estos atributos manejan la prioridad del proceso, el tiempo de ejecución asignado y en qué procesador puede ejecutarse el proceso.
  - **schedproc[NR\_PROCS]:** Esto declara una matriz de estructuras schedproc llamada schedproc que especifica el número máximo de procesos admitidos en el sistema.
- **Schedule.c (src/servers/pm)**
  - **sched\_init**

inicializa la planificación para todos los procesos del sistema. Recorre todas las entradas en la tabla de procesos (mproc)
  - **sched\_start\_user**

Se utiliza para iniciar la planificación de un proceso de usuario específico. Calcula la prioridad máxima del proceso basada en su valor "nice" y elige el padre adecuado para la herencia de la configuración del planificador.
  - **sched\_nice**

Ajusta la prioridad de un proceso basándose en su valor "nice".

**Preguntas:**

- **¿cuales son las 2 principales razones por la cual Andrew llama a Linux obsoleto?**

La primera es que menciona que Linux es un SO monolitico, lo cuál de entrada acusa de ser parte de Sistemas Operativos viejos, que basicamente son un archivo que se ejecuta en modo kernel que contiene la gestión de procesos, memoria, el sistema de archivos y otros. Andrew considera que los SO de microkernel son más eficientes ya que la mayor parte del sistema operativo se ejecuta como procesos separados, principalmente fuera del kernel. Se comunican mediante el paso de mensajes. El trabajo del kernel es manejar el paso de mensajes, el manejo de interrupciones, la gestión de procesos de bajo nivel y posiblemente la entrada y salida.

La segunda razón es la portabilidad ... no entendi todos los números que menciona Andrew pero por lo que terminé entendiendo el tema de la portabilidad se refiere a que se deben diseñar Sistemas Operativos que puedan ser funcionales con arquitecturas más avanzadas y que sigan siendo útiles mediante las computadoras siguen evolucionando... cosa que aparentemente Linux no cumplía en esos años.

- **¿que puntos/predicciones hechos por Andrew siguen siendo correctos en la actualidad?**