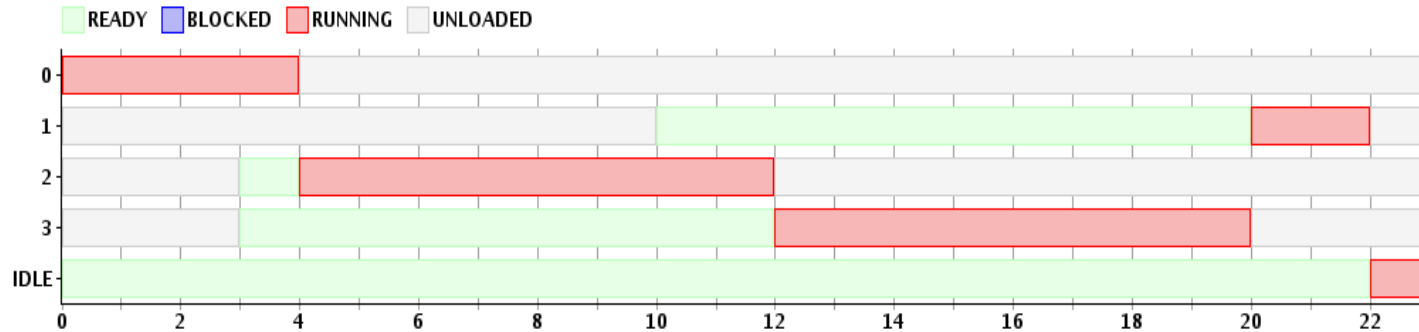


# Presentación TP1 - Scheduling

Sistemas Operativos. DC - FCEN - UBA  
1er cuatrimestre de 2015

# Scheduler

Módulo del sistema operativo que se encarga de la remoción, selección y reemplazo del proceso en ejecución.



# Entendiendo el simulador (Lotes)

- Tareas Aperiódicas
  - Tipo (TaskCPU, TaskIO, ....)
  - Parámetros
  - Release time
- Tareas Periódicas &Ar,p,t
  - Tipo fijo: TaskCPU(t)
  - Release time
  - Nombre familia: A
  - Cantidad repeticiones: r
  - Tiempo entre repeticiones: p

TaskCPU 3  
@10:  
TaskCPU 1  
@3:  
\*2 TaskCPU 7

&A4,10,2  
TaskCPU 3  
&B1,5,2

# Entendiendo el simulador (Tareas)

- Función dentro de tasks.cpp
- `uso_CPU(t)`
- `uso_IO(t)`
- `return`

```
void TaskIO(int pid, vector<int> params) {  
    // Uso el CPU params[0] clicos.  
    uso_CPU(pid, params[0]);  
  
    // Uso IO params[1]+1 ciclos.  
    uso_IO(pid, params[1]);  
  
    // Uso1 ciclo  
    return;  
}
```

# Entendiendo el simulador

`./simusched <lote.tsk> <num_cores> <costo_cs> <costo_mi> <sched> [<params_sched>]`

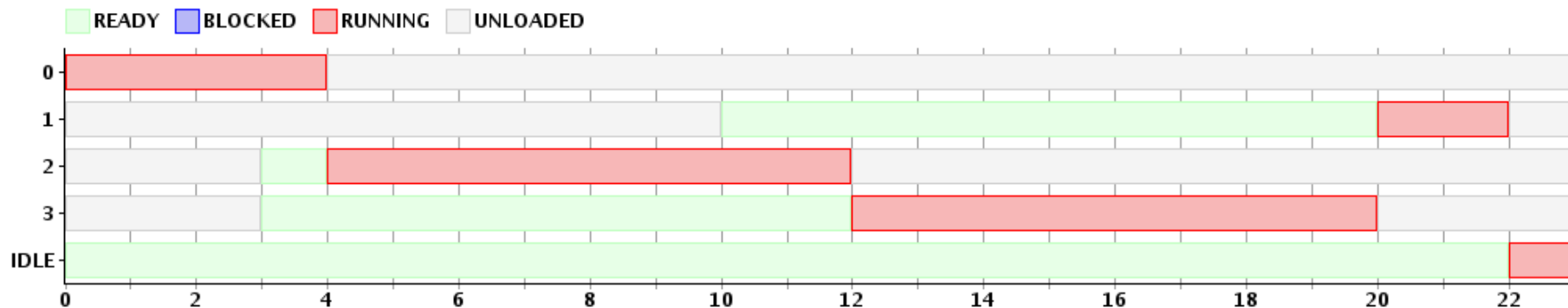
- `<lote.tsk>` es el archivo que especifica el lote de tareas a simular.
- `<num_cores>` es la cantidad de núcleos de procesamiento.
- `<costo_cs>` es el costo de cambiar de contexto.
- `<costo_mi>` es el costo de cambiar un proceso de núcleo
- `<sched>` es el nombre de la clase de scheduler a utilizar

# Entendiendo el simulador

- **LOAD 0 1.** En el ciclo 0, la tarea 1 está ready
- **CPU 33 1 0.** En el ciclo 33, se está ejecutando la tarea 1 en el core 0
- **BLOCK 44 1.** En el ciclo 44, la tarea 1 está bloqueada
- **UNBLOCK 65 1.** En el ciclo 65, se desbloqueó la tarea 1
- **EXIT 66 1 0.** En el ciclo 66, la tarea 1 terminó en el core 0
- **CPU 112 -1 0.** En el ciclo 112, el core 0 está idle.
- **CONTEXT CPU 0 32.** Cambio de contexto en el core 0, en el ciclo 32

# Graficación de simulaciones

```
/simusched lote.tsk 1 1 5 SchedFCFS | ./graphsched.py >> imagen.png
```



Pueden tener que instalar las siguientes librerías

```
sudo apt-get install python-matplotlib libfreetype6-dev ttf-freefont
```

```
sudo pip install Pillow
```

<http://www.dc.uba.ar/materias/so/2015/c1/descargas/tps/tp1/view>



# Funciones globales para Real Time

&Ar,p,t

**char tipo(int pid)**

Devuelve la familia del proceso pid

Es decir, **A**

**unsigned int period (int pid)**

Devuelve el período declarado por las tareas de esa familia

Es decir, **p**

**unsigned int declared\_cputime (int pid)**

Devuelve el tiempo de CPU que las tareas de la familia declaran consumir

Es decir, **t** ¿Es t el tiempo total de ejecución de una tarea?

**unsigned int total\_tasks()**

Número total de tareas, sólo permitido para SchedFixed

# Recomendaciones



# El ejercicio fue sencillo

## MALA IDEA

El código es lo suficientemente sencillo como para no mencionarlo en el informe ni agregar comentarios.

Asumo que el corrector va a entender por qué hice lo que hice.

Para cada nuevo nombre de variable, elijo una letra del alfabeto no utilizada.

## BUENA IDEA

Menciono el ejercicio en el informe y las decisiones de implementación

Agrego comentarios en el código

Los nombres de mis variables son declarativos

# Elijo una métrica para evaluar mi algoritmo

## MALA IDEA

Como la métrica fue mencionada en la teórica no necesito explicarla.

Mis lotes de pruebas muestran un buen rango de números enteros de mi métrica. Excelente!

## BUENA IDEA

Explico la métrica: su definición y la motivación detrás de esa definición.

Mis lotes de prueba muestran los casos donde mi algoritmo es bueno/malo para esa métrica según su semántica.

# El nuevo algoritmo supera al anterior

## MALA IDEA

Duplicando la cantidad de núcleos, bajando el costo de cambio de contexto a la mitad y la duración de las tareas a un tercio, mi algoritmo termina la tareas en menor tiempo. Definitivamente es un éxito

## BUENA IDEA

Alterno un parámetro por vez entre experimentos para comparar los resultados.

Cuando tenga sentido, y pueda explicar ese sentido, alterno más de un parámetro.

# El nuevo algoritmo supera al anterior (2da parte)

## MALA IDEA

Evidentemente, es mejor para todos los casos. Mi mamá dice eso de mí

No menciono los casos donde no funciona bien por si me bajan puntos por programar cosas ineficientes.  
Total, seguro no se dan cuentan.

## BUENA IDEA

Explico bajo qué métrica y bajo qué contextos mi nuevo algoritmo es mejor que el anterior.

Si hay casos donde el anterior es mejor, también los menciono.

# Investigamos soluciones de terceros

## MALA IDEA

Lo dejo en su idioma original para dejar en claro que no es mío.

Encontré la definición en el rincón del vago, pongo el link como referencia.

## BUENA IDEA

Cito la publicación de la cual me basé.

Utilizo publicaciones científicas o libros especializados.

Reescribo con mis palabras la idea.

# Realizamos muchos experimentos

## MALA IDEA

Seguro que 1000 gráficos son mejores que 10. Hagamos ese script en python.

*"Se ve claramente que cumple lo que queríamos demostrar"*

## BUENA IDEA

Los gráficos responden las preguntas que planteo (y realmente las planteo de manera escrita)

Escribo cuáles son las cuestiones relevantes que el gráfico muestra, sus parámetros, sus valores y su relación con las hipótesis.



## MALA IDEA

En promedio, los datos están en el valor esperado.

“¿Dónde se cobra el ingreso per cápita? A más de un muerto de hambre le gustaría saberlo” <sup>1</sup>

1. Galeano, E. *El libro de los abrazos*. Siglo XXI, 2000

## BUENA IDEA

El promedio no es una medida estadística muy útil por sí sola.

Puede mejorar nuestro análisis si incluimos la desviación estándar para saber cuán dispersos están nuestros datos de ese promedio. Idealmente, lo que buscamos es estimar la distribución de nuestros datos.

## MALA IDEA

Nos dividimos los ejercicios, cada uno escribe su parte. Alguien copia y pega el último día cada ejercicio y arma una linda carátula.

## BUENA IDEA

El informe es un texto integral. Debe tener una introducción general, los detalles relevantes de implementación, los resultados y la discusión de estos para cada ejercicio, y una conclusión general que recopile las ideas planteadas a lo largo de todo el trabajo.