

# TP1 - Scheduling

Sistemas operativos

Verano - 2010

¿Qué es una Tarea? (desde el punto de vista del TP)

¿Qué es una Tarea? (desde el punto de vista del TP)

Es un programa que se va a ejecutar

## ¿Qué es una Tarea? (desde el punto de vista del TP)

Es un programa que se va a ejecutar

Antes de simular conocemos:

- En que momento estará lista para ser ejecutada
- Cuanto tiempo de procesamiento utiliza

## ¿Qué es una Tarea? (desde el punto de vista del TP)

Es un programa que se va a ejecutar

### Antes de simular conocemos:

- En que momento estará lista para ser ejecutada
- Cuanto tiempo de procesamiento utiliza

### La simulación nos permitirá conocer:

- En que momento finalizó la tarea
- Cuanto tiempo de espera tuvo

# Un Simulador de Schedulers

- En el *prompt*, tipear: `./simulator`
- Respuesta:  
Usage: simulator <<scheduler\_name>> <<task\_set\_name>>  
          <<quantum>>

---

Resultados analisis del scheduling producido:

tiempo de espera promedio: -1

---

URL to generate graph:

```
http://chart.apis.google.com/chart?chs=
1000x200&chbh=a&chxt=x&chd=t:11|33|22|11|11|
11&cht=bhs&chco=E8B9AF,AB9BB7,788E9D,AFB856,
E8B9AF,D5B77E&chm=tIDLE_TASK,000000,0,0,20|tP1,
000000,1,0,20|tP2,000000,2,0,20|tP3,000000,3,0,
20|tIDLE_TASK,000000,4,0,20|tP4,000000,5,0,
20&chxl=0:|0|4|8|12|16|20|24|28|32|36|40|44
```

- FCFS, SJF: Implementados por la cátedra
- RR, MFQ: Cáscaras para que completen Uds.



- ts1, ts2, ts3, ts4, ts5, ts6: Son los *tasksets de la cátedra*
- fcfs\_1, fcfs\_2: Para jugar con FCFS
- sjf\_1, sjf\_2: Idem con SJF
- test\_1: Un taskset cuyos tasks llegan luego del instante inicial
- ej\_1: Lo deben escribir Uds.

# Qué es un Task

Campos readonly

- name: Nombre del Task
- ptime (Processing time): time necessary to execute task on the processor without interruption
- rtime (Released time): Momento en el que la tarea esta lista para su ejecución

Estos campos no cambian a lo largo de la vida del Task.

# Qué es un Task

## Campos para la simulación

- ttime (Task time): Tiempo de procesador usado por el task. Por defecto vale 0.
- ftime (Finalization Time): Tiempo de finalizacion de la tarea. Por defecto vale -1.
- wtime (Waiting Time): Tiempo de espera de la tarea. Por defecto vale 0.

Estos campos son modificados producto de la simulación.

- Alberga un conjunto de Tareas (Tasks)
- Las tareas no pueden tener un nombre repetido
- Ejemplo:

```
TaskSet taskset1 = new TaskSet("taskset1");  
taskset1.add ( new Task("P1", 15, 5 ) );  
taskset1.add ( new Task("P2", 10, 5 ) );  
taskset1.add ( new Task("P3", 05, 6 ) );
```

Es la clase abstracta que representa un planificador. Todos los schedulers extienden esta clase abstracta.

<code>void init(TaskSet taskset)</code>	Prepara al scheduler para la planificación del TaskSet
<code>String next()</code>	Avanza una unidad de tiempo. Retorna la tarea que está usando el procesador o bien <i>null</i> para marcar el fin de la planificación

# FCFS

init(TaskSet taskSet)

FOR EACH tarea IN TaskSet

agregar a la lista de tareas preparadas de acuerdo a su  
orden de llegada y su rtime

```
IF no hay mas tasks THEN Terminar()
current_time++
IF (prox_tarea.rtime > current_time)
    RETURN IDLE_TASK
IF (prox_tarea.ttime == prox_tarea.ptime) THEN
    prox_tarea.ftime := current_time
    IF no hay mas tareas THEN Terminar()
    ELSE
        prox_tarea := siguiente tarea preparada()
        IF prox_tarea.rtime > current_time
            RETURN IDLE_TASK
    ENDIF
prox_tarea.ttime++
RETURN prox_tarea.name
```

Permite calcular estadísticas de la planificación. Esta clase es invocada una vez finalizada la planificación.

```
/**  
 * Retorna el waiting time promedio de  
 * una tarea.  
 */  
public int get_avg_wtime() {  
    return -1;  
}
```



# Enunciado Trabajo Práctico

## Parte 1: Entrenamiento en el Simulador

- Definir un TaskSet que produzca una planificación particular en FCFS y en SJF
- Completar la clase SchedulerAnalyzer para que calcule el waiting time promedio de un TaskSet ya planificado

# Enunciado Trabajo Práctico

## Parte 2: Nonpreemptive Scheduling

- Usando los algoritmos provistos por la cátedra (FCFS y SJF) estudiar el comportamiento sobre los taskSet  $ts_1, \dots, ts_6$ .
  - Ejemplo: ¿Cómo la planificación del TaskSet ... el algoritmo ...? ¿Por qué?
- Calcular el waiting time promedio de cada tarea sobre cada algoritmo.

# Enunciado Trabajo Práctico

## Parte 3: Preemptive Scheduling

- Implementar el algoritmo de RR y estudiar su comportamiento sobre los tasksets usando distintos quantum.
- Implementar el algoritmo de MFQ
- Modificar el algoritmo de MFQ para que simule el bloqueo de procesos por operaciones de E/S