



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
ESCUELA DE INGENIERÍA  
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2233 - Programación Avanzada  
2° semestre 2015

# Actividad 16

## Threading avanzado - Workers v2.0!

### Problema

El programa de la AC15 funcionó bien, pero algo no fue completamente satisfactorio: es incómodo que mientras el usuario escribe un comando, eventualmente un worker puede terminar e imprimir su resultado, separando el input del usuario en dos partes. Además, cuando se calcula varianza, se abre nuevamente el archivo para el cual ya se le había calculado el promedio, lo cual incrementa el tiempo que demora en terminar. Por esto, se le pide que haga algunas modificaciones a su programa.

### Requerimientos

Los cambios son los siguientes:

- Cuando un worker termina, **no imprime** su resultado, sino que lo guarda en una lista de outputs. Un output es una tupla de la forma (**nombre**, **comando**, **resultado**, **tiempo**), donde:
  - **nombre** es el nombre con el cual se identifica al thread que terminó (por ejemplo, 'Thread-1').
  - **comando** es un string que indica la instrucción y la estrella.
  - **resultado** es 'DONE' si la instrucción fue **open**, y el valor numérico del promedio/varianza si la instrucción fue **mean** o **var**.
  - **tiempo** es la fecha y hora en que terminó de ejecutarse el comando. Puede importar **datetime**.
- Para evitar que un mismo archivo de texto se abra nuevamente al calcular varianza, los comandos posibles ingresados por el usuario ahora son los siguientes (**nota:** solo los primeros tres comandos requieren que se cree un worker para su ejecución):
  - **open estrella:** carga el archivo que corresponde a la estrella, es decir, guarda un conjunto de magnitudes de brillos en un diccionario de estrellas ya cargadas (indexado por los nombres de las estrellas).
  - **mean estrella:** calcula el promedio de los datos de una estrella ya cargada.
  - **var estrella:** calcula la varianza de los datos de una estrella ya cargada.
  - **status:** imprime la lista de outputs con los resultados que hay hasta el momento.
  - **exit:** indica que ya no se ingresarán más comandos.

- A las razones por las cuales no se puede ejecutar un comando, se agrega que no se puede ejecutar **mean estrella** o **var estrella** si los datos de la estrella no han terminado de ser cargados al diccionario.
- Cuando se ingresa **exit**, se imprime la lista con los outputs que hay hasta el momento y luego se imprime cuáles comandos no terminaron de ejecutarse.

## To - DO

Se le entrega una solución de la AC15 como main. Usted puede basarse en ella, o (si confía lo suficiente en su solución) trabajar en base a una solución propia. El puntaje se distribuye como sigue:

- (2.0 pts) Modificar correctamente la clase **Worker**.
- (2.5 pts) Modificar correctamente el programa principal.
- (1.5 pts) Debes preocuparte de que la lista de outputs y cualquier otra variable que pueda ser modificada por los workers sean thread-safe.

## Tips

- Recuerda que para conseguir la información de las estrellas debes correr **getdata.py**. Antes de hacer **git push**, borra estos archivos de tu carpeta. **NO subas estos archivos a tu repositorio. Habrá descuento de puntaje.**

## Ejemplo

```
Ingrese siguiente comando:
var Sirius
[DENIED] La estrella Sirius aun no ha sido cargada al sistema
      o no ha terminado de cargarse
Ingrese siguiente comando:
open Siriius
Ingrese siguiente comando:
status
Lista de outputs: [
]
Ingrese siguiente comando:
mean Sirius
[DENIED] La estrella Sirius aun no ha sido cargada al sistema
      o no ha terminado de cargarse
Ingrese siguiente comando:
open AlphaCentauri
Creando Worker para: open AlphaCentauri
Ingrese siguiente comando:
status
Lista de outputs: [
  ("Thread-1", "open Sirius", "DONE", "2015-09-27 15:39:33")
]
Ingrese siguiente comando:
mean Sirius
Creando Worker para: mean Sirius
Ingrese siguiente comando:
exit
Lista de outputs: [
  ("Thread-1", "open Sirius", "DONE", "2015-09-27 15:39:33"),
  ("Thread-3", "mean Sirius", -1.4698, "2015-09-27 15:39:53")
]
No alcanzo a terminar: Thread-2 / open AlphaCentauri
```

## Mensaje motivacional

*Un programador tenía un problema y decidió usar threads.  
Ahora tiene dos, tres y cuatro problemas.  
Y todos al mismo tiempo.*