
Examen de la práctica de PRO2 - Primavera 2023

X76954_es

El objetivo en este examen será modificar algunos aspectos de tu solución de la práctica. Concretamente deberás

1. cambiar uno de los criterios para elegir el procesador en el que colocar un proceso durante la funcionalidad `enviar_procesos_cluster`
2. cambiar el criterio para colocar un proceso en un procesador
3. añadir una funcionalidad nueva que borre ciertos procesadores del cluster e imprima información sobre dicho borrado

Los detalles están descritos con precisión en el documento `enunciado_examen.pdf` que forma parte del archivo `public.tar` que puedes descargar en esta página (icono del gatito). Verás que tendrás que modificar elementos ya existentes en tu práctica, así como añadir funcionalidades nuevas al main y operaciones nuevas a algunas clases. Las novedades en el formato de la entrada y la salida del programa, incluyendo mensajes de error, se deducen del sample ("Public Test Cases").

IMPORTANTE: Este problema del Jutge es el único canal disponible para realizar la entrega del examen de la práctica. Ten en cuenta que

- Dispondrás de 2h 30m para la resolución del examen.
- La nota del examen procederá exclusivamente de la corrección automática de tu último envío. No penaliza el número de intentos, pero recomendamos empezar a enviar soluciones solo cuando funcionen con el sample en vuestra consola. Recordad que se pierde mucho tiempo mientras se evalúa un envío.
- No superar ningún juego de pruebas conlleva un cero en la nota de este examen, al igual que si se aprecian indicios suficientes de copia.

Observación

Recomendamos que hagas una copia "limpia" de tu práctica en un subdirectorio aparte y en esa copia llesves a cabo todos los cambios.

Verifica que todos los módulos compilan sin errores, que el proceso de montaje da un archivo ejecutable correcto y que el `program.exe` pasa el juego de pruebas público suministrado. Comprueba que tu `Makefile` genera el fichero `program.exe` y crea el fichero `.tar` para hacer el envío al Jutge. Asegúrate que entregas el fichero `.tar` con la solución del examen, no la solución original. No es necesario incluir la carpeta con la documentación en Doxygen ni, en su caso, la de la competencia transversal.

El Jutge prueba tus entregas mediante 4 juegos de pruebas que, salvo por los cambios introducidos en este enunciado, tienen la misma sintaxis que los de la práctica y menos requisitos de eficiencia.

- sample: el juego de pruebas público
- privat1: las funcionalidades de la entrega intermedia (incluyendo eficiencia)

- privat2: un poco de todo (sin eficiencia)
- privat3: eficiencia

En un fichero llamado `practica.tar` debes entregar

- Los ficheros `.hh` y `.cc` de las clases y el programa principal. No incluyas ficheros `.o` y similares.
- El fichero `Makefile`, que usaremos para generar y probar el ejecutable

Ten en cuenta las siguientes restricciones:

- El fichero que contiene el programa principal se ha de llamar `program.cc` (recuerda actualizar el `main` con la nueva opción)
- El `Makefile` ha de generar un ejecutable llamado `program.exe`
- Es importante que uses las opciones de compilación del Jutge de PRO2 (véase [Documentation](#) → [Compilers](#) → PRO2 en www.jutge.org)
- No usar la opción `-D_GLIBCXX_DEBUG` o usarla de forma incorrecta podrá ser penalizado

Produce el fichero `practica.tar` con la instrucción Linux

```
tar -cvf practica.tar fitxer1 fitxer2 fitxer3 ...
```

desde el directorio/carpeta donde tengas los ficheros que vas a entregar. Incluye esta instrucción en tu `Makefile`, de forma que el `.tar` se pueda generar ejecutando `make practica.tar`. Con eso reducirás el riesgo de error en sucesivas entregas. El Jutge no acepta `.tar` donde los ficheros estén dentro de carpetas. Recomendamos usar GNU `tar` para reducir el riesgo de que el fichero `practica.tar` sea incompatible con el Jutge.

Ejemplo de entrada 1

<pre> proc1 10 proc11 100 proc111 500 * * proc112 500 * * proc12 100 proc121 500 * * proc122 500 * * 4 1111 11 11111 111 imprimir_estructura_cluster imprimir_area_espera imprimir_procesadores_cluster alta_proceso_procesador pr 99 15 15 app proc12 99 10 15 imprimir_procesador pr modificar_cluster pp p1 100 * * mc proc1 p2 100 </pre>	<pre> * * ipro proc12 mc proc12 p3 100 * * mc proc121 p1 100 p2 100 * * p3 100 * * iec ipc ipro proc12 app proc1 99 10 10 ipro proc1 </pre>
---	---

```
alta_proceso_procesador
proc1 99 5 5
app proc12 101 91 15
app proc12 101 90 15
ipc
baja_proceso_procesador
pr 99
bpp proc12 100
bpp proc12 99
ipro proc12
avanzar_tiempo 7
imprimir_procesador proc12
at 2
ipro proc1
alta_proceso_espera 222 99 10 10

ape 111 99 10 10

ape 111 99 20 15

ape 111 98 1000 20

ape 111 100 10 1

imprimir_prioridad 222
ipri 111
alta_prioridad 1111
ap 2
baja_prioridad 333
bp 111
bp 11
ape 1111 80 10 10
ape 111 80 5 5
iae
enviar_procesos_cluster 1
ipc
iae
epc 100
ipc
iae
iec

pc 1
iec
ipc

configurar_cluster
sndr01
1000
*
*
iec

podar_cluster 20
ipc
iec

cc
abc
200
*
```

```
zsda
550
*
*

iec

iae

cmp proc

compactar_memoria_procesador abc

ipro abc
app abc 40 50 8
app abc 30 50 5
app abc 10 50 10
app abc 20 50 5
app zsda 30 50 15
at 5
compactar_memoria_cluster
ipc
at 4
cmc
ipc
iec

pc 1
ipc
iec

fin
```

Ejemplo de salida 1

```
#imprimir_estructura_cluster
(proc1(proc11(proc111 ) (proc112 )) (proc12(proc121 ) (proc122 )))
#imprimir_area_espera
11
0 0
111
0 0
1111
0 0
11111
0 0
#imprimir_procesadores_cluster
proc1
proc11
proc111
proc112
proc12
proc121
proc122
#alta_proceso_procesador pr 99
error: no existe procesador
#app proc12 99
#imprimir_procesador pr
error: no existe procesador
#modificar_cluster pp
error: no existe procesador
#mc proc1
error: procesador con auxiliares
#ipro proc12
0 99 10 15
#mc proc12
error: procesador con procesos
#mc proc121
#iec
(proc1(proc11(proc111 ) (proc112 )) (proc12(proc121 ) (proc122 )))
#ipc
p1
p2
p3
proc1
proc11
proc111
proc112
proc12
0 99 10 15
proc122
#ipro proc12
0 99 10 15
#app proc1 99
#ipro proc1
0 99 10 10
#alta_proceso_procesador proc1 99
error: ya existe proceso
#app proc12 101
error: no cabe proceso
#app proc12 101
#ipc
p1
p2
```

```
p3
proc1
0 99 10 10
proc11
proc111
proc112
proc12
0 99 10 15
10 101 90 15
proc122
#baja_proceso_procesador pr 99
error: no existe procesador
#bpp proc12 100
error: no existe proceso
#bpp proc12 99
#ipro proc12
10 101 90 15
#avanzar_tiempo 7
#imprimir_procesador proc12
10 101 90 8
#at 2
#ipro proc1
0 99 10 1
#alta_proceso_espera 222 99
error: no existe prioridad
#ape 111 99
#ape 111 99
error: ya existe proceso
#ape 111 98
#ape 111 100
#imprimir_prioridad 222
error: no existe prioridad
#ipri 111
99 10 10
98 1000 20
100 10 1
#alta_prioridad 1111
error: ya existe prioridad
#ap 2
#baja_prioridad 333
error: no existe prioridad
#bp 111
error: prioridad con procesos
#bp 11
#ape 1111 80
#ape 111 80
#iae
111
99 10 10
98 1000 20
100 10 1
80 5 5
0 0
1111
80 10 10
0 0
11111
0 0
2
0 0
```

```

#enviar_procesos_cluster 1
#ipc
p1
p2
p3
procl
0 99 10 1
procl1
procl11
0 99 10 10
procl12
procl2
10 101 90 6
procl22
#iae
111
98 1000 20
100 10 1
80 5 5
1 0
1111
80 10 10
0 0
11111
0 0
2
0 0
#epc 100
#ipc
p1
p2
p3
procl
0 99 10 1
procl1
procl11
0 99 10 10
10 80 10 10
procl12
0 100 10 1
procl2
10 101 90 6
procl22
0 80 5 5
#iae
111
98 1000 20
3 1
1111
1 0
11111
0 0
2
0 0
#iec
(procl(procl1(procl11 ) (procl12 )) (procl122 ))
#pc 1
Poda 6
#iec
(procl (procl2 (procl22 )))
#ipc

```

```

procl
0 99 10 1
procl2
10 101 90 6
procl22
0 80 5 5
#configurar_cluster
#iec
(sndr01 )
#podar_cluster 20
error: se intenta borrar la raiz
#ipc
sndr01
#iec
(sndr01 )
#cc
#iec
(abc (zsda ))
#iae
111
98 1000 20
3 1
1111
1 0
11111
0 0
2
0 0
#cmp proc
error: no existe procesador
#compactar_memoria_procesador abc
#ipro abc
#app abc 40
#app abc 30
#app abc 10
#app abc 20
#app zsda 30
#at 5
#compactar_memoria_cluster
#ipc
abc
0 40 50 3
50 10 50 5
zsda
0 30 50 10
#at 4
#cmc
#ipc
abc
0 10 50 1
zsda
0 30 50 6
#iec
(abc (zsda ))
#pc 1
Poda 6
#ipc
abc
0 10 50 1
zsda
0 30 50 6

```

#iec

| (abc (zsda))

Información del problema

Autoría: PRO2

Generación: 2026-01-25T21:26:19.362Z

© *Jutge.org*, 2006–2026.

<https://jutge.org>