

---

## Práctica de PRO2 - Primavera 2023 - entrega final provisionalX31703\_es

---

Este problema permite hacer entregas de prueba de la práctica completa. Tened en cuenta que:

- no es el canal para hacer la entrega definitiva de la práctica
- el problema del Jutge de la entrega definitiva **puede contener juegos de prueba o condiciones diferentes** de los que aparecen en este problema
- las entregas realizadas en este problema no serán tenidas en cuenta para la nota de la práctica
- en esta entrega no pedimos carpetas generadas mediante `doxygen`, pero los ficheros de las clases pueden llevar todos los comentarios `doxygen` que queráis; de hecho recomendamos que incluyáis las especificaciones de todas las operaciones y que sean lo más definitivas posible

### Entrada

Una secuencia de instrucciones y datos que siguen el formato del enunciado de la práctica y del juego de pruebas público.

### Salida

Una secuencia de resultados que siguen el formato del enunciado de la práctica y del juego de pruebas público.

### Observación

El Jutge prueba vuestras entregas mediante 4 juegos de pruebas:

- `sample`: el juego de pruebas público
- `privat1`: combinación de todos los juegos de pruebas de la entrega intermedia
- `privat2`: un poco de todo, salvo eficiencia
- `privat3`: eficiencia

En un fichero llamado `practica.tar` tenéis que entregar

- Los ficheros `.hh` y `.cc` de las clases y el programa principal
- El fichero `Makefile`, que usaremos para generar y probar el ejecutable

Tened en cuenta las siguientes restricciones:

- El fichero que contiene el programa principal se ha de llamar `program.cc`
- El `Makefile` ha de generar un ejecutable llamado `program.exe`

- Es importante que uséis las opciones de compilación definidas en el menú Documentación → Compilers → PRO2 de [www.jutge.org](http://www.jutge.org) para funcionar en igualdad de condiciones con el Jutge
- No usar la opción `-D_GLIBCXX_DEBUG` o usarla de forma incorrecta podrá ser penalizado

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

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

desde el directorio/carpeta donde tengáis los ficheros que vais a entregar. incluid en vuestro Makefile una regla con esta instrucción, de forma que el `.tar` se pueda generar ejecutando `make practica.tar`. Con eso reduciréis 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

<pre> proc1 10 proc11 100 proc111 500 * * proc1 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 * * ipro proc12 mc proc12 p3 100 * * mc proc121 p1 100 p2 100 * * p3 100 * </pre>	<pre> * iec ipc ipro proc12 app proc1 99 10 10 ipro proc1 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 </pre>
---	---

```

iae
epc 100
ipc
iae
configurar_cluster
sndr01
1000
*
*
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
fin

```

## Ejemplo de salida

```

#imprimir_estructura_cluster
(proc1(proc11(proc111 ) (proc112 )) (proc12(proc121 )
#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(p1(p2 ) (p
#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
procl
0 99 10 10
procl1
procl11
procl12
procl2
0 99 10 15
10 101 90 15
procl22
#baja_proceso_procesador pr 99
error: no existe procesador
#bpp procl2 100
error: no existe proceso
#bpp procl2 99
#ipro procl2
10 101 90 15
#avanzar_tiempo 7
#imprimir_procesador procl2
10 101 90 8
#at 2
#ipro procl
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
0 0
#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
procl12
procl2
0 99 10 10
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
0 80 10 10
p2
p3
procl
0 99 10 1
procl1
0 100 10 1
10 80 5 5
procl11
procl12
procl2
0 99 10 10
10 101 90 6
procl22
#iae
111
98 1000 20
3 1
1111
1 0
11111
0 0
2
0 0
#configurar_cluster
#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
```

## Información del problema

Autor : PRO2

Generación : 2023-05-31 14:46:39

© *Jutge.org*, 2006–2023.

<https://jutge.org>