

UPMurphi

UPMurphi es un planificador basado en BFS desarrollado por Giuseppe Della Penna de la Università degli Studi dell'Aquila, de Italia.

Para instalarlo, tenemos una guía en su repositorio de [github](#), pero lo explico brevemente a continuación:

1. Instalamos las dependencias:

```
sudo apt-get install build-essential flex bison libc6-dev-i386 gcc-multilib g++-multilib
```

2. Entramos en el directorio “src” y compilamos con “make”

```
cd src
```

```
make
```

Si no hemos tenido ningún error, ya deberíamos tener el compilador funcionando. Para comprobarlo podemos usarlo con los programas de prueba que tenemos en la carpeta “ex” dentro del propio repositorio.

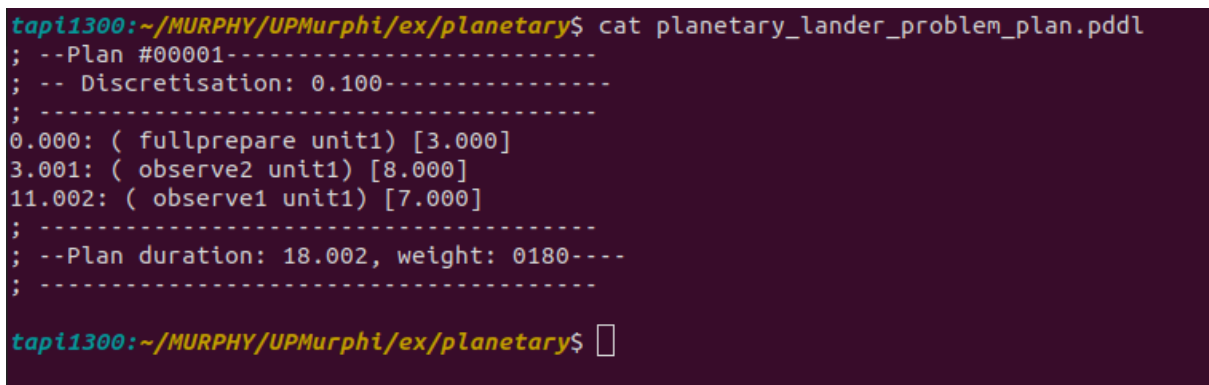
Entramos por ejemplo en el ejemplo “planetary” y ejecutamos el siguiente comando:

```
../../bin/upmc planetary_lander.pddl planetary_lander_problem.pddl
```

Este comando discretiza el problema y crea una serie de archivos que vamos a utilizar en el siguiente paso para crear el plan. El más importante de todos ellos es un archivo ejecutable llamado “planetary_lander_planner”. Ejecutamos dicho ejecutable de la siguiente forma:

```
./planetary_lander_planner
```

Cuando termine de ejecutarse, nos creará un archivo pddl con el mismo nombre que el problema añadiendo “_plan”, en nuestro caso será “planetary_lander_problem_plan.pddl”, que si lo leemos nos dirá el plan obtenido.



```
tapi1300:~/MURPHY/UPMurphi/ex/planetary$ cat planetary_lander_problem_plan.pddl
; --Plan #00001-----
; -- Discretisation: 0.100-----
; -----
0.000: ( fullprepare unit1) [3.000]
3.001: ( observe2 unit1) [8.000]
11.002: ( observe1 unit1) [7.000]
; -----
; --Plan duration: 18.002, weight: 0180---
; -----
tapi1300:~/MURPHY/UPMurphi/ex/planetary$
```

Como hemos podido comprobar, la búsqueda del problema es algo lenta por el método que se utiliza.

También tiene limitaciones en su uso, ya que requiere que el dominio utilice tipos (“:typing”), y la única métrica que puede usar es “:minimize total-time”.

Este planificador, de los requerimientos que hemos dado en clase, no tiene implementados los siguientes:

- :numeric-fluents
- :ucpop
- :true-negation
- :open-world

- :expression-evaluation
- :safety-constraints
- :subgoal-through-axioms
- :domain-axioms
- :dag-expansions
- :foreach-expansions
- :action-expansions

Referencias:

<https://github.com/gdellapenna/UPMurphi>

<http://people.disim.univaq.it/dellapenna/content.php?page=upmurphi>