

LARtk 5

Jorge Almeida

Departamento de Engenharia Mecânica
Universidade de Aveiro
almeida.j@ua.pt

9 de Março de 2015

O que é?

- Repositório com o material produzido por membros do LAR ao longo dos anos.
- Composto por *packages* ROS formato *catkin*.
- Estrutura em árvore, dividida em *meta-packages* por funcionalidades.
- Formato Git (<http://git-scm.com/>). Em caso de problemas no acesso enviar email para almeida.j@ua.pt.
- Versão atual 5^a, primeira em suporte Git.

Estrutura do repositório

hardware/

... (documentação)

src/

applications/

bases/

demos/

hmi/

navigation/

perception/

planning/

sensors/

utils/

Descrição

applications conjuntos de *launch files* para lançar uma aplicação final

bases controlo de *hardware*

demos demonstrações simples

hmi *human machine interface*

navigation módulos para navegação autónoma

perception detecção de pedestres, objetos, estrada, etc.

planning planeamento de trajetórias

sensors comunicação com sensores

utils utilidades genéricas

Como utilizar

1 Criar a pasta `workincopies` na vossa pasta `home`

2 Clonar para a pasta `workincopies` o repositório:

- `git clone ssh://USER@lars.mec.ua.pt/home/repositories/lar5 lar5`
- **atenção ao `lar5` final**

3 Iniciar o *catkin workspace* na pasta `lar5`

- em `~/workincopies/lar5/src`, **fazer:** `catkin_init_workspace`

4 Compilar o LARtk

- em `~/workincopies/lar5`, **fazer:** `catkin_make`
- atualizar no `.bashrc` a leitura do ficheiro `setup.bash` para o novo `~/workincopies/lar5/devel/setup.bash`

5 *Packages* com dependências específicas não iram compilar. Caso algum *package* não compilado seja necessário, leia os *warnings* do `cmake`

Para utilizar o Git no início de trabalho é necessário criar um *branch* próprio, para isso siga as instruções

1 Atualizar o código local

```
git pull
```

2 Criar *branch* de trabalho

```
git checkout -b NOVOBRANCH
```

3 Adicionar novo *branch* ao servidor remoto (origin)

```
git push --set-upstream origin NOVOBRANCH
```

4 Trabalhar normalmente sobre o novo *branch*

```
git add FICHEIRO  
git commit FICHEIRO -m "MENSAGEM"  
git push
```

A divisão em *branches* apresenta algumas vantagens:

- evita perturbação do trabalho por parte dos colegas, dado que trabalham em *branches* diferentes.
- permite o funcionamento ininterrupto do *branch* `master`.

Em situação futura será necessário integrar o trabalho efetuado num *branch* no `master`. Neste caso terá que se efetuar um *merge*. Esta operação será apenas efetuada pela pessoa responsável pelo repositório.

Mais informação sobre Git consultar:

- <http://git-scm.com/book/pt-br/v1>
- <https://www.atlassian.com/git/tutorials/comparing-workflows/feature-branch-workflow>

Regras

- Não são permitidas letras maiúsculas nos nomes dos *packages*.
- **Não criar mensagens novas, utilizar as existentes!**
- Não replicar funcionalidades, procurar primeiro!
 - 1 procurar na net um *package* ROS.
 - 2 se não encontrar, procurar no LARtk.
 - 3 só depois criar.
- Tentem inspirarem-se nos *packages* já existentes, a nível de organização e nomenclatura.
- A criação de novos *packages* terá de ser enquadrada nos *meta-packages* existentes (*applications*, *bases*, etc.).
 - utilizar ferramentas *standard* para criação de *packages*, `catkin_create_pkg`.
- As pastas `build` e `devel` fazem parte do normal funcionamento do *catkin*. Estas pastas **não se encontram sob controlo de versões**.

Regras, referentes a código

- Colocar código fonte em `PACKAGE/src`.
- Colocar *headers* locais (apenas uso próprio) em `PACKAGE/src`.
- Colocar *headers* públicos (usados por outros *packages*) em `PACKAGE/include/PACKAGE`.
- Todos os ficheiros de código (*headers* e *sources*) criados têm de possuir **obrigatoriamente** um *copyright* no início do ficheiro. Este poderá ser encontrado nos ficheiros já existentes (licença BSD).
- Todos os *packages* necessitam de possuir **obrigatoriamente** um ficheiro `mainpage.dox` para colocação da documentação em formato *doxygen*.
- A incorporação de código de terceiros é permitida tendo em atenção a licença do material original.

Considerações finais

Com o *workflow* em *branches* não existe o problema de conflitos entre código de várias pessoas. Deste modo, é encorajado o recurso ao Git e a manter o servidor atualizado (mesmo que o código não compile).

Nem todos os alunos terão a necessidade de criar novos *package*. Quando iniciarem o trabalho num *package* já existente deverão adicionar o vosso nome à lista de autores e substituir o *maintainer* no ficheiro `package.xml`.