

Menu

[Skip to content](#)

- [Home](#)
- [About](#)
- [Calendar](#)

[Home](#) > [Study](#) > [2\) Computer Engineering](#) > OpenLava Installation

OpenLava Installation

Posted on [2015/06/10](#) by [gumdaeng](#) — [Leave a comment](#)

지난 1주일간, 분산되어 관리되고 있던 연구실 컴퓨터에 OpenLava를 설치했다[1]. OpenLava는 컴퓨터 클러스터에 작업 스케줄링을 해주는 오픈 소스 스케줄러이다[2]. 클러스터에 작업 스케줄링을 한다는 개념 자체가 생소했기에 설치도 어려울 것으로 생각했는데, 생각보다 설치는 어렵지 않았다. [Quickstart](#) 또는 [documentation](#)을 참고해 설치하면 된다.

1. OpenLava 설치

다음의 명령어로 OpenLava를 설치할 수 있다.

/opt/openlava-3.0 디렉토리 아래에 OpenLava 3.0이 설치된다.

```
1 | #wget http://www.openlava.org/tarball/openlava-3.0.tar.gz
2 | #tar -xzvf ./openlava-3.0.tar.gz
3 | #cd openlava-3.0
4 | #./configure
5 | #make
6 | #make install
```

다음으로, 설정 파일을 복사한다.

```
1 | #cd ./config
2 | #for f in `ls | grep -v in | grep -v Makefile` ; do cp $f /opt/openlava-3.0/etc; dc
```

OpenLava를 root 계정으로 실행할 것이므로, openlava 계정을 따로 생성하지 않는다.

동일한 이유로 /opt/openlava-3.0의 소유자를 변경하지 않는다.

(필요하다면 Quickstart를 참고해 openlava 계정 생성)

OpenLava를 서비스에 등록하고 실행한다.

```
1 | #cp /opt/openlava-3.0/etc/openlava /etc/init.d
2 | #cp /opt/openlava-3.0/etc/openlava.* /etc/profile.d
3 | #service openlava start
```

2. 설정 파일 변경

하지만, 이렇게 실행한 OpenLava는 정상적으로 실행되지는 않는다.

클러스터에 속하는 컴퓨터들의 정보를 모르기 때문이다.

설정 파일들이 /opt/openlava-3.0/etc 아래에 있다. 이를 수정해주어야 한다.

```
1 | lsb.hosts - 호스트 정보에 대한 파일 (중요)
2 | lsb.params
3 | lsb.queues - 작업 대기 행렬의 정의
```

```
4 lsb.users - 사용자 및 사용자 그룹 정의
5 lsf.cluster.openlava - 클러스터 정의 (중요)
6 lsf.conf -
7 lsf.shared - 자원 타입 등의 정의. 개별 호스트의 연산 능력을 정의할 수 있다.
8 lsf.task
```

반드시 수정해주어야 하는 것은 lsb.hosts와 lsf.cluster.openlava이다.

2.1 lsb.hosts

*lsb.hosts*는 호스트 정보를 정의하는 파일이다.

MXJ은 해당 노드에서 실행할 수 있는 최대 작업 수이다.

그 외의 필드는 [5] 또는 [6]을 참고하면 확인할 수 있다.

man 명령어를 사용해 확인할 수도 있다. (man lsb.hosts)

중요하지 않은 값은 () 또는 -을 사용해 생략할 수 있다.

다음은 우리 연구실 클러스터의 lsb.hosts 파일이다.

```
1 #
2 # The section "host" is optional.  If no hosts are listed here, all hosts
3 # known by LSF will be used by Batch.  Otherwise only the hosts listed will
4 # be used by Batch.  The value of keyword HOST_NAME may be an official host
5 # name (see gethostbyname(3)), a host type/model name (see lsf.shared(5)), or
6 # the reserved word "default".  The type/model name represents each of the
7 # hosts which are of that particular host type/model.  The reserved
8 # word "default" represents all other hosts in the LSF cluster.
9
10 # MXJ is the maximum number of jobs which can run on the host at one time.
11 # JL/U is the maximum number of jobs belonging to a user that can run on the
12 # host at one time.  The default is no limit.
13
14 # DISPATCH_WINDOW is the time windows when the host is available to run
15 # batch jobs.  The default dispatch window is always open.
16
17 # Other columns specify scheduling and stopping thresholds for LIM load
18 # indices.  A "(" or "-" is used to specify the default value in a column
19 # and cannot be omitted.
20
21 # All the host names (except default) in this example are commented out,
22 # since they are just examples which may not be suitable for some sites.
23 # Don't use non-default thresholds unless job dispatch needs to be controlled.
24 Begin Host
25 HOST_NAME      MXJ    JL/U    r1m    pg     ls     tmp    DISPATCH_WINDOW
26 cluster        10    ()     ()     ()     ()     ()     ()
27 taurus00       16    ()     ()     ()     ()     ()     ()
28 taurus01       12    ()     ()     ()     ()     ()     ()
29 taurus02       12    ()     ()     ()     ()     ()     ()
30 taurus03       12    ()     ()     ()     ()     ()     ()
31 taurus04       12    ()     ()     ()     ()     ()     ()
32 taurus05       12    ()     ()     ()     ()     ()     ()
33 taurus06       12    ()     ()     ()     ()     ()     ()
34 taurus07       12    ()     ()     ()     ()     ()     ()
35 taurus08       12    ()     ()     ()     ()     ()     ()
36 taurus09       12    ()     ()     ()     ()     ()     ()
37 taurus10       12    ()     ()     ()     ()     ()     ()
38 taurus11       12    ()     ()     ()     ()     ()     ()
39 taurus12       12    ()     ()     ()     ()     ()     ()
40 taurus13       12    ()     ()     ()     ()     ()     ()
41 taurus14       12    ()     ()     ()     ()     ()     ()
42 taurus15       12    ()     ()     ()     ()     ()     ()
43 taurus16       12    ()     ()     ()     ()     ()     ()
44 taurus17       12    ()     ()     ()     ()     ()     ()
45 taurus18       12    ()     ()     ()     ()     ()     ()
46 taurus19       12    ()     ()     ()     ()     ()     ()
```

```

47 ursus01      16  ()  ()  ()  ()  ()  ()
48 ursus02      16  ()  ()  ()  ()  ()  ()
49 ursus03      16  ()  ()  ()  ()  ()  ()
50 ursus04      16  ()  ()  ()  ()  ()  ()
51 ursus05      16  ()  ()  ()  ()  ()  ()
52 ursus06      16  ()  ()  ()  ()  ()  ()
53 ursus07      16  ()  ()  ()  ()  ()  ()
54 ursus08      16  ()  ()  ()  ()  ()  ()
55 ursus09      16  ()  ()  ()  ()  ()  ()
56 ursus10      16  ()  ()  ()  ()  ()  ()
57 ursus11      16  ()  ()  ()  ()  ()  ()
58 canis01       6  ()  ()  ()  ()  ()  ()
59 canis02       6  ()  ()  ()  ()  ()  ()
60 canis03       6  ()  ()  ()  ()  ()  ()
61 canis04       6  ()  ()  ()  ()  ()  ()
62 canis05       6  ()  ()  ()  ()  ()  ()
63 canis06       6  ()  ()  ()  ()  ()  ()
64 canis07       6  ()  ()  ()  ()  ()  ()
65 canis08       8  ()  ()  ()  ()  ()  ()
66 canis09       8  ()  ()  ()  ()  ()  ()
67 canis10       8  ()  ()  ()  ()  ()  ()
68 canis11       6  ()  ()  ()  ()  ()  ()
69 canis12       6  ()  ()  ()  ()  ()  ()
70 canis13       8  ()  ()  ()  ()  ()  ()
71 canis14       8  ()  ()  ()  ()  ()  ()
72 canis15       4  ()  ()  ()  ()  ()  ()
73 canis16      12  ()  ()  ()  ()  ()  ()
74 canis17       8  ()  ()  ()  ()  ()  ()
75 End Host
76
77 # Host groups can be referenced by the queue file.  Each line defines a host
78 # group.  The first line contains key words; each subsequent line contains a
79 # group name, followed by white space, followed by the list of group members.
80 # The list of members should be enclosed in parentheses and separated by white
81 # space.  This section is optional.
82 Begin HostGroup
83 GROUP_NAME    GROUP_MEMBER
84 taurus      (taurus00 taurus01 taurus02 taurus03 taurus04 taurus05 taurus06 tauru
85 ursus      (ursus01 ursus02 ursus03 ursus04 ursus05 ursus06 ursus07 ursus08 ursu
86 canis      (canis01 canis02 canis03 canis04 canis05 canis06 canis07 canis08 cani
87 End HostGroup

```

중요한 것은, hostname과 /etc/hosts의 hostname, 각 노드의 hostname이 일치해야 한다는 점이다. 이것이 일치하지 않으면 해당 노드에서 서비스가 제대로 시작되지 않는다.

2.2 lsf.cluster.openlava

*lsf.cluster.openlava*는 클러스터를 정의하는 파일이다.

관리자 아이디, 그리고 클러스터를 구성하는 호스트들을 입력한다.

관리자 아이디는 서비스를 실행하는 사용자와 일치해야 하며,

/opt/openlava-3.0/ 아래의 모든 파일 또한 관리자 소유여야 한다.

```

1  #-----
2  # THIS IS A ONE PER CLUSTER FILE
3  #
4  # This is a sample cluster definition file.  There is a cluster
5  # definition file for each cluster.  This file's name should be
6  # lsf.cluster.<cluster-name>.
7  # See lsf.cluster(5) and the "LSF Administrator's Guide".
8  #
9
10 Begin ClusterAdmins
11 Administrators = root

```

```

12 End ClusterAdmins
13
14 Begin Host
15 HOSTNAME model type server r1m RESOURCES
16 cluster ursus linux 1 3.5 ()
17 taurus00 taurus linux 1 3.5 ()
18 taurus01 taurus linux 1 3.5 ()
19 taurus02 taurus linux 1 3.5 ()
20 taurus03 taurus linux 1 3.5 ()
21 taurus04 taurus linux 1 3.5 ()
22 taurus05 taurus linux 1 3.5 ()
23 taurus06 taurus linux 1 3.5 ()
24 taurus07 taurus linux 1 3.5 ()
25 taurus08 taurus linux 1 3.5 ()
26 taurus09 taurus linux 1 3.5 ()
27 taurus10 taurus linux 1 3.5 ()
28 taurus11 taurus linux 1 3.5 ()
29 taurus12 taurus linux 1 3.5 ()
30 taurus13 taurus linux 1 3.5 ()
31 taurus14 taurus linux 1 3.5 ()
32 taurus15 taurus linux 1 3.5 ()
33 taurus16 taurus linux 1 3.5 ()
34 taurus17 taurus linux 1 3.5 ()
35 taurus18 taurus linux 1 3.5 ()
36 taurus19 taurus linux 1 3.5 ()
37 ursus01 ursus linux 1 3.5 ()
38 ursus02 ursus linux 1 3.5 ()
39 ursus03 ursus linux 1 3.5 ()
40 ursus04 ursus linux 1 3.5 ()
41 ursus05 ursus linux 1 3.5 ()
42 ursus06 ursus linux 1 3.5 ()
43 ursus07 ursus linux 1 3.5 ()
44 ursus08 ursus linux 1 3.5 ()
45 ursus09 ursus linux 1 3.5 ()
46 ursus10 ursus linux 1 3.5 ()
47 ursus11 ursus linux 1 3.5 ()
48 canis01 canis linux 1 3.5 ()
49 canis02 canis linux 1 3.5 ()
50 canis03 canis linux 1 3.5 ()
51 canis04 canis linux 1 3.5 ()
52 canis05 canis linux 1 3.5 ()
53 canis06 canis linux 1 3.5 ()
54 canis07 canis linux 1 3.5 ()
55 canis08 ursus linux 1 3.5 ()
56 canis09 ursus linux 1 3.5 ()
57 canis10 ursus linux 1 3.5 ()
58 canis11 canis linux 1 3.5 ()
59 canis12 canis linux 1 3.5 ()
60 canis13 canis linux 1 3.5 ()
61 canis14 canis linux 1 3.5 ()
62 canis15 canis linux 1 3.5 ()
63 canis16 canis linux 1 3.5 ()
64 canis17 canis linux 1 3.5 ()
65 End Host
66
67 Begin ResourceMap
68 RESOURCENAME LOCATION
69 # tmp2 [default]
70 # nio [all]
71 # console [default]
72 End ResourceMap

```

2.3 그 외

더 효율적인 클러스터 구성을 위해서 `lsb.queues`, `lsf.shared`를 수정해주면 좋다.

`lsb.queues`는 대기 행렬을 정의하는데, 설치하면 기본으로 `normal` 대기 행렬만 정의하고 있다. 다른 대기 행렬을 주석 해제하거나, 값을 변경해 사용할 수 있다.

`lsf.shared`는 일종의 전역 변수를 정의하는데, 호스트 타입과 연산 능력을 정의할 수 있다.

다음은 연구실에서 클러스터 구성에 사용한 `lsf.shared` 파일이다.

```
1 # $Id: lsf.shared,v 1.1.4.4 2006/04/28 21:36:01 mblack Exp $
2
3 # -----
4 # T H I S   F I L E:  Is shared by all clusters in the openlava system.
5 #
6 # This file contains all definitions referenced by individual
7 # lsf.cluster.<clustername> files. The definitions in this file can be
8 # a superset, i.e., not all definitions in this file need to be used in
9 # other files.
10 #
11 # See lsf.cluster(5)
12 # -----
13
14 Begin Cluster
15 ClusterName                # Keyword
16 openlava
17 End Cluster
18
19 Begin HostType
20 TYPENAME                    # Keyword
21 linux
22 End HostType
23
24 #
25 # The CPU factor values are derived from SPECfp95 given by hardware vendors
26 # or SpecBench (unless indicated otherwise)
27 # See http://www.specbench.org for more information on CPU benchmarking
28 # To find out an architecture string for a new model, run 'lim -t'
29 #
30 Begin HostModel
31 MODELNAME  CPUFACTOR  ARCHITECTURE # keyword
32 # CPU factors are only comparisons.
33 taurus      100      (x86_64)
34 ursus       100      (x86_64)
35 canis        40      (x86_64)
36 End HostModel
37
38 Begin Resource
39 RESOURCENAME  TYPE      INTERVAL  INCREASING  DESCRIPTION      # Keywords
40 fs            Boolean ()      ()          (File server)
41 cs            Boolean ()      ()          (Compute server)
42 # nio         Numeric 60      Y           (Network I/O in Kbytes/second)
43 # console     String 60      ()          (Console user name)
44 End Resource
```

3. 서비스 재시작

이렇게 설정한 설정 파일을 모든 노드에 반영하고, `openlava` 서비스를 재시작한다.

`bhosts` 명령어로 올바르게 구성되었는지 확인할 수 있다.

```
1 #service openlava restart
2 #pssh -h ~/cluster_nodes -i 'service openlava restart'
3 #bhosts
4 HOST_NAME      STATUS      JL/U      MAX  NJOBS      RUN  SSUSP  USUSP      RSV
5 canis01        ok          -         6    0          0    0      0          0
6 canis02        ok          -         6    0          0    0      0          0
```

7	canis03	ok	-	6	0	0	0	0	0
8	canis04	ok	-	6	0	0	0	0	0
9	canis05	ok	-	6	0	0	0	0	0
10	canis06	ok	-	6	0	0	0	0	0
11	canis07	ok	-	6	0	0	0	0	0
12	canis08	ok	-	8	0	0	0	0	0
13	canis09	ok	-	8	0	0	0	0	0
14	canis10	ok	-	8	0	0	0	0	0
15	canis11	ok	-	6	0	0	0	0	0
16	canis12	ok	-	6	0	0	0	0	0
17	canis13	ok	-	8	0	0	0	0	0
18	canis14	ok	-	8	0	0	0	0	0
19	canis15	ok	-	4	0	0	0	0	0
20	canis16	ok	-	12	0	0	0	0	0
21	canis17	ok	-	8	0	0	0	0	0
22	cluster	ok	-	10	0	0	0	0	0
23	taurus00	ok	-	16	0	0	0	0	0
24	taurus01	ok	-	12	0	0	0	0	0
25	taurus02	ok	-	12	0	0	0	0	0
26	taurus03	ok	-	12	0	0	0	0	0
27	taurus04	ok	-	12	0	0	0	0	0
28	taurus05	ok	-	12	0	0	0	0	0
29	taurus06	ok	-	12	0	0	0	0	0
30	taurus07	ok	-	12	0	0	0	0	0
31	taurus08	ok	-	12	0	0	0	0	0
32	taurus09	ok	-	12	0	0	0	0	0
33	taurus10	ok	-	12	0	0	0	0	0
34	taurus11	ok	-	12	0	0	0	0	0
35	taurus12	ok	-	12	0	0	0	0	0
36	taurus13	ok	-	12	0	0	0	0	0
37	taurus14	ok	-	12	0	0	0	0	0
38	taurus15	ok	-	12	0	0	0	0	0
39	taurus16	ok	-	12	0	0	0	0	0
40	taurus17	ok	-	12	0	0	0	0	0
41	taurus18	ok	-	12	0	0	0	0	0
42	taurus19	ok	-	12	0	0	0	0	0
43	ursus01	ok	-	16	0	0	0	0	0
44	ursus02	ok	-	16	0	0	0	0	0
45	ursus03	ok	-	16	0	0	0	0	0
46	ursus04	ok	-	16	0	0	0	0	0
47	ursus05	ok	-	16	0	0	0	0	0
48	ursus06	ok	-	16	0	0	0	0	0
49	ursus07	ok	-	16	0	0	0	0	0
50	ursus08	ok	-	16	0	0	0	0	0
51	ursus09	ok	-	16	0	0	0	0	0
52	ursus10	ok	-	16	0	0	0	0	0
53	ursus11	ok	-	16	0	0	0	0	0

References:

- [1] OpenLava, <http://www.openlava.org/>
- [2] OpenLava, Wikipedia, <http://en.wikipedia.org/wiki/OpenLava>
- [3] Quickstart, OpenLava, <http://www.openlava.org/documentation/quickstart.html>
- [4] Install from tarball, OpenLava, http://www.openlava.org/documentation/guide/getting_started.html#install-from-tarball
- [5] Documentation, OpenLava, <http://www.openlava.org/documentation/guide/index.html>
- [6] Configuration Files, IBM Knowledge Center, https://www-01.ibm.com/support/knowledgecenter/SSETD4_9.1.3/lfs_config_ref/part_files.dita

[About these ads](#)

Share this:

- [Twitter](#)
- [Facebook](#)
- [Google](#)
- [Print](#)
- [Email](#)
-

Like this:

Like Loading...

Related

< [20150610](#)

[Ubuntu Install Automation](#) >

Tagged with: [cluster](#), [클러스터](#), [연구실](#), [OpenLava](#), [server](#)

Posted in [2\) Computer Engineering](#)

Leave a Reply



Email

(Address never made public)

Name

Website

Notify me of new comments via email.

Notify me of new posts via email.

search here ...

Categories

- [Gumdaeng's Story](#)
 - [1\) My Story](#)
 - [2\) My Playlist](#)
- [Study](#)
 - [1\) Memo](#)
 - [2\) Computer Engineering](#)
 - [3\) English](#)

Archives

- [June 2015](#)
- [May 2015](#)
- [April 2015](#)
- [March 2015](#)
- [February 2015](#)
- [January 2015](#)
- [December 2014](#)
- [November 2014](#)
- [October 2014](#)
- [September 2014](#)
- [August 2014](#)
- [July 2014](#)
- [June 2014](#)
- [May 2014](#)
- [April 2014](#)
- [March 2014](#)
- [February 2014](#)
- [January 2014](#)
- [December 2013](#)
- [November 2013](#)
- [October 2013](#)
- [August 2013](#)
- [July 2013](#)
- [June 2013](#)
- [April 2013](#)
- [March 2013](#)
- [February 2013](#)
- [January 2013](#)
- [December 2012](#)
- [November 2012](#)
- [October 2012](#)
- [September 2012](#)
- [August 2012](#)
- [June 2012](#)
- [May 2012](#)
- [April 2012](#)
- [March 2012](#)
- [December 2011](#)
- [November 2011](#)
- [July 2011](#)
- [June 2011](#)
- [April 2011](#)
- [January 2011](#)
- [June 2010](#)
- [April 2010](#)
- [March 2010](#)
- [January 2010](#)
- [December 2009](#)
- [October 2009](#)
- [September 2009](#)
- [August 2009](#)
- [June 2009](#)
- [April 2009](#)

- [February 2009](#)
- [December 2008](#)
- [November 2008](#)
- [October 2008](#)
- [August 2008](#)
- [July 2008](#)
- [June 2008](#)
- [May 2008](#)
- [April 2008](#)
- [December 2007](#)
- [July 2006](#)
- [October 2003](#)
- [September 2003](#)
- [July 2001](#)
- [June 1999](#)
- [March 1999](#)
- [January 1999](#)
- [October 1998](#)

Recent Posts

- [Protected: 20150610 – Research](#) 2015/06/10
- [Ubuntu Install Automation](#) 2015/06/10
- [OpenLava Installation](#) 2015/06/10
- [20150610](#) 2015/06/10
- [20150604](#) 2015/06/04
- [20150602](#) 2015/06/02
- [Protected: 20150601 – Research](#) 2015/06/01
- [20150601](#) 2015/06/01
- [20150529](#) 2015/05/29
- [Protected: 20150528 – Research](#) 2015/05/28
- [20150528](#) 2015/05/28
- [Protected: 20150527 – Research](#) 2015/05/27
- [20150527](#) 2015/05/27
- [Protected: 20150526 – Research](#) 2015/05/26
- [20150526](#) 2015/05/26

Digital Library

- [ACM Digital Library](#)
- [IEEE Xplore Digital Library](#)

Favorites

- [네이버 만화 – 넓이 사는 이야기](#)
- [네이버 만화 – 가우스전자](#)
- [네이버 캐스트 – 오늘의 과학](#)
- [빈곶](#)
- [nixCraft](#)
- [Phd Comics](#)
- [Top Conferences in Computer Science](#)
- [Truly Teach Me Tarot](#)

Friends

- [Anna](#)

- [Jeongim](#)
- [NoSyu](#)
- [phil](#)
- [xpenguin](#)
- [yulistic](#)

Gumdaeng

- [gumdaeng's ask.fm](#)
- [gumdaeng's facebook](#)
- [gumdaeng's Github](#)
- [gumdaeng's Google +](#)
- [gumdaeng's Twitter](#)

KAIST

- [Computer Architecture Laboratory](#)
- [대나무숲](#)
- [아라](#)
- [KAIST](#)
- [KAIST Computer Science](#)
- [KAIST Dormitory](#)
- [KAIST Library](#)
- [KLMS](#)
- [Suebok Moon's Blog](#)

SKKU

- [SKKU College of Information & Communication Engineering](#)
- [SKKU Dormitory](#)
- [SKKU Icampus](#)
- [SKKU Library](#)
- [SKKU Love](#)
- [Sungkyunkwan University](#)

태그

2014 [bash](#) [benchmark](#) [BoB](#) [build](#) [casker](#) [code](#) [command](#) [ebs](#) [excel](#) [facebook](#) [git](#) [google](#) [humor](#) [installation](#)

[KAIST](#) [linux](#) [make](#) [parov](#) [stelar](#) [Private](#) [programming](#) [python](#) [security](#) [simulation](#) [simulator](#) [SKKU](#)

[source](#) [TA](#) [terminal](#) [ubuntu](#) [vba](#) [vim](#) [wordpress](#) [xen](#) [경영학과](#) [고등학생](#) [고슴도치](#) [공부](#) [금성고등학교](#) [기숙사](#) [대학교](#) [대학](#)

[생](#) [대학원](#) [리눅스](#) [보안](#) [브로콜리](#) [너마저](#) [블로그](#) [석사](#) [성균관대학교](#) [소스](#) [수업](#) [술](#) [시간표](#) [시험](#) [신입생](#) [엑](#)

[셀](#) [연구](#) [연구실](#) [옷](#) [워드프레스](#) [입시](#) [재무관리](#) [재미](#) [전산학과](#) [졸업](#) [카이스트](#) [캐스커](#) [컴퓨터](#) [컴퓨터공학과](#) [컴](#)

[퓨터구조](#) [코드](#) [파이썬](#) [프로그래밍](#) [프로젝트](#) [학기](#)

누적 방문자 수

- 42,965 hits

 [xkcd](#)

- [The Martian](#) 2015/06/10
- [Words for Pets](#) 2015/06/08
- [Beer](#) 2015/06/05
- [Antique Factory](#) 2015/06/03

- [06/05/15 PHD comic: 'The Exercycle'](#) 2015/06/07
- [06/03/15 PHD comic: 'Kickstarter Doodles'](#) 2015/06/03
- [05/29/15 PHD comic: 'Things you can do in Academia that would get you fired in the Real World'](#) 2015/05/31
- [05/27/15 PHD comic: 'Academic Contradictions'](#) 2015/05/27
- [05/22/15 PHD comic: 'The Doctoral Dilemma'](#) 2015/05/23

© 2015 [gumdaeng](#)



[Blog at WordPress.com.](#) | [The Responsive Theme.](#)

[Follow](#)

Follow “gumdaeng”

Get every new post delivered to your Inbox.

[Build a website with WordPress.com](#)

%d bloggers like this:

