GNOME™
Desktop for Everyone

Kukuh Syaafat
cho2@di.blankon.in

Parade Software Merdeka, 30 Agustus 2015

GNOME

Kenalan dulu ...

- **Kukuh Syafaat**
  - Pengembang dan pengguna BlankOn (blankon.id)
  - GNU/Linux *enthusiast*



Apa itu GNOME™?

- Lingkungan destop yang berjalan diatas sistem operasi GNU/Linux (**GNOME Shell**)
- Dikembangkan oleh GNOME Projects
- Ditulis dalam bahasa C, C++, Vala, Python dan JavaScript
- Rilis stabil terakhir versi 13.6.3

Kenapa GNOME™?

- Digunakan oleh distribusi GNU/Linux yang populer seperti Debian, Fedora, OpenSUSE dan Arch Linux.
- Sebagian besar distribusi GNU/Linux juga menggunakan GNOME™ sebagai *desktop*-nya *http://distrowatch.com/search.php?desktop=GNOME*
- Lisensi GPL, LGPL
- Lokalisasi lebih dari 40 bahasa
- Dukungan komunitas yang luas

GNOME

Rilis GNOME™

- 2.2: 2003-02-06
- . . .
- 2.24: 2008-09-24
- 2.26: 2009-03-18
- 2.28: 2009-09-23
- 2.30: 2010-03-31
- 2.32: 2010-09-29
- 3.0: 2011-04-06
- 3.2: 2011-09-28
- 3.4: 2012-03-28
- 3.6: 2012-09-26
- 3.8: 2013-03-27
- 3.10: 2013-09-25
- 3.12: 2014-03-26
- 3.14: 2014-09-24
- 3.16: 2015-03-25

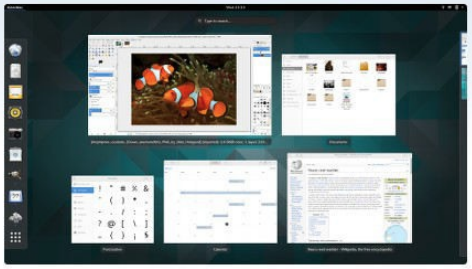
Activities Web Sat 13:51

GNOME
https://www.gnome.org/

GNOME.org

GNOME

GNOME 3 Applications About Us Get Involved Foundation Search



GNOME 3: Ease, comfort and control

GNOME 3 is an easy and elegant way to use your computer. It is designed to put you in control and bring freedom to everybody. GNOME 3 is developed by the GNOME community, a diverse, international group of contributors that is supported by an independent, non-profit foundation.

Discover GNOME 3 Get GNOME 3

Make a donation and become a Friend of GNOME!

Your donation will ensure that GNOME continues to be a free and open source desktop by providing resources to developers, software and education for end users, and promotion for GNOME worldwide.

Get involved!

The GNOME Project is a diverse international community which involves hundreds of contributors, many of whom are volunteers. Anyone can contribute to the GNOME!

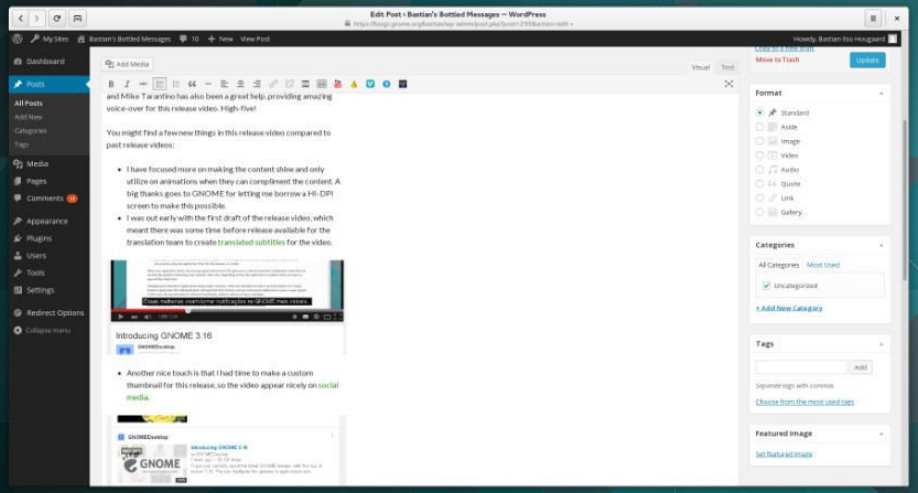
Latest news

March 25, 2015 March 16, 2015 February 16, 2015

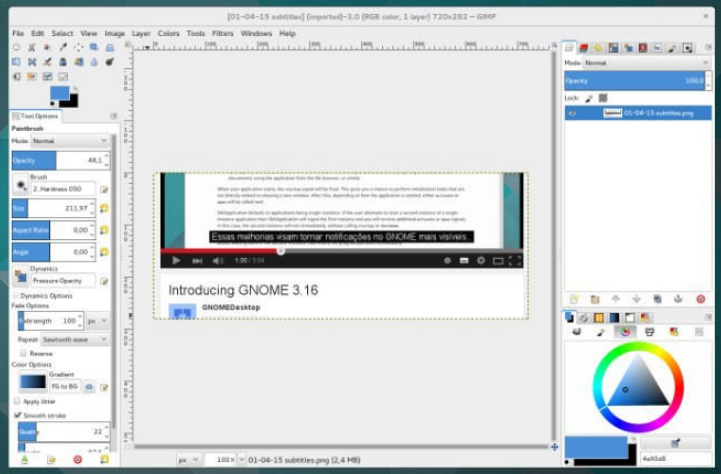
Activities

Sat 13:44

Type to search...



Edit Post ‹ Bastian's Bottled Messages — WordPress



[01-04-15 subtitles] (imported)-3.0 (RGB color, 1 layer) 720x282 – GIMP

HowDoI/GtkApplication – GNOME Wiki!
http://wiki.gnome.org/HowDoI/GtkApplication

GtkApplication is the base class of a Gtk Application. Its primary purpose is to separate your program from `main()`.

`main()` is an operating system implementation detail that is really uninteresting to applications. The philosophy of GtkApplication is that applications are interested in being told what needs to happen, when it needs to happen, in response to actions from the user. The exact mechanism by which the operating system starts applications is uninteresting.

To this end, GtkApplication exposes a set of signals (or virtual functions) that an application should respond to.

- `startup`: sets up the application when it first starts
- `shutdown`: performs shutdown tasks
- `activate`: shows the default first window of the application (like a new document). This corresponds to the application being launched by the desktop environment.
- `open`: opens files and shows them in a new window. This corresponds to someone trying to open a document (or documents) using the application from the file browser, or similar.

When your application starts, the `startup` signal will be fired. This gives you a chance to perform initialisation tasks that are not directly related to showing a new window. After this, depending on how the application is started, either `activate` or `open` will be called next.

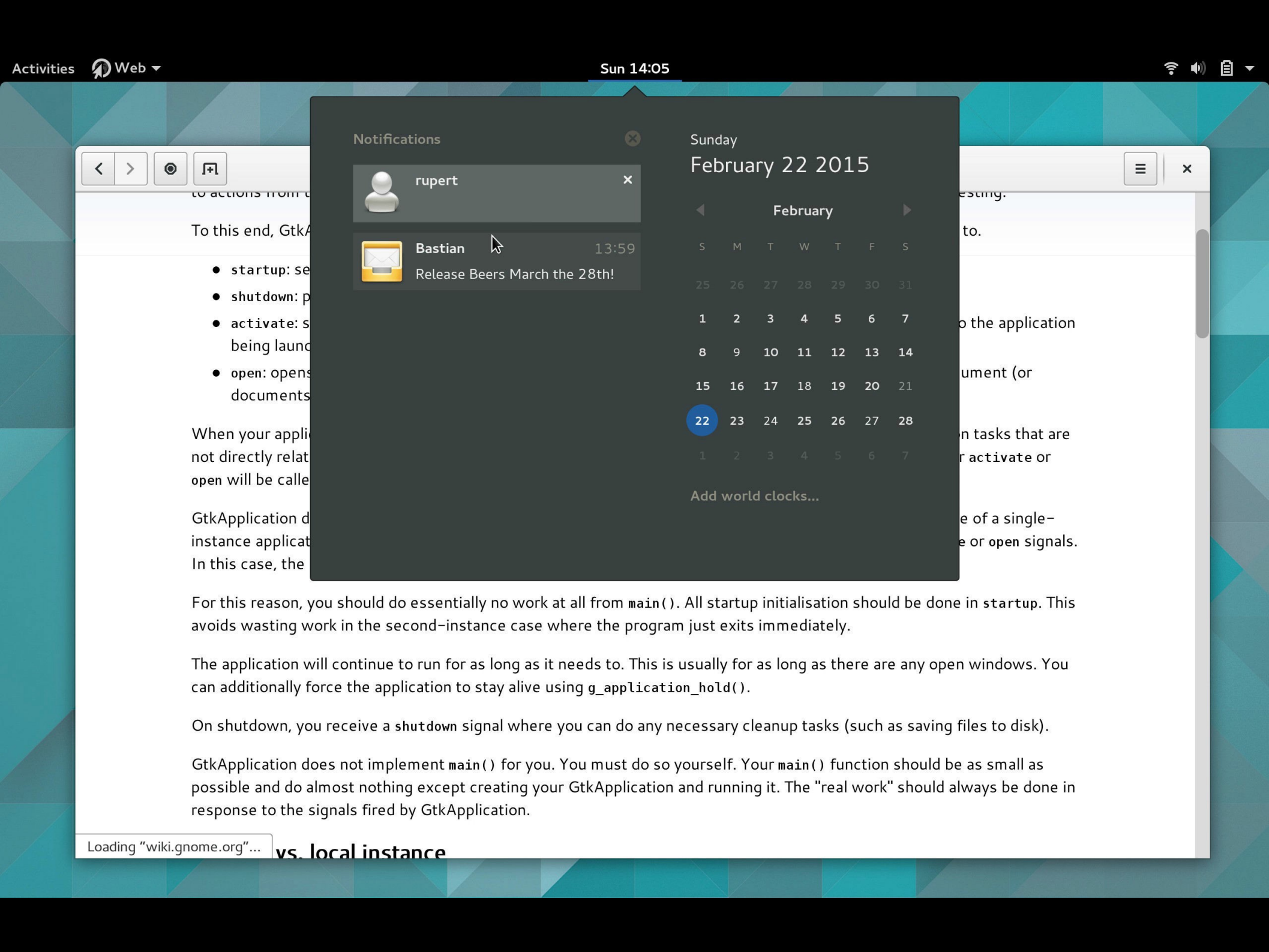
GtkApplication defaults to applications being single-instance. If the user attempts to start a second instance of a single-instance application then GtkApplication will signal the first instance and you will receive additional `activate` or `open` signals. In this case, the second instance will exit immediately, without calling `startup` or `shutdown`.

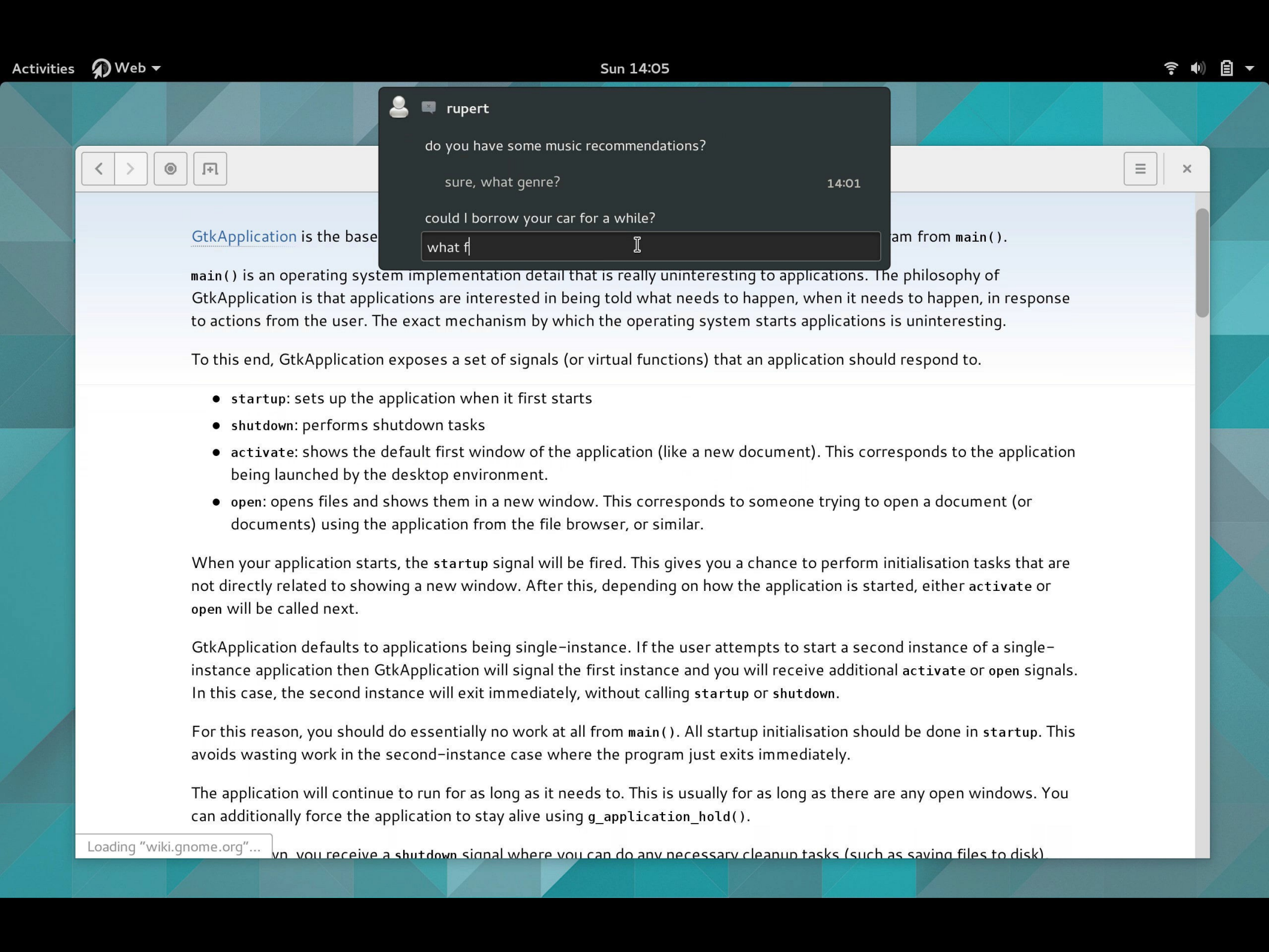
For this reason, you should do essentially no work at all from `main()`. All startup initialisation should be done in `startup`. This avoids wasting work in the second-instance case where the program just exits immediately.

The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using `g_application_hold()`.

Loading "wiki.gnome.org"...

...vn, you receive a `shutdown` signal where you can do any necessary cleanup tasks (such as saving files to disk).


Activities
Web
Sun 14:05
Notifications
rupert
Bastian
13:59
Release Beers March the 28th!
Sunday
February 22 2015
February
S M T W T F S
25 26 27 28 29 30 31
1 2 3 4 5 6 7
8 9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
1 2 3 4 5 6 7
Add world clocks...
startup: se
shutdown: p
activate: s
being launc
open: opens
documents
When your appli
not directly relat
open will be calle
GtkApplication d
instance applicat
In this case, the
to.
o the application
ument (or
n tasks that are
r activate or
e of a single-
e or open signals.
For this reason, you should do essentially no work at all from main(). All startup initialisation should be done in startup. This avoids wasting work in the second-instance case where the program just exits immediately.
The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using g_application_hold().
On shutdown, you receive a shutdown signal where you can do any necessary cleanup tasks (such as saving files to disk).
GtkApplication does not implement main() for you. You must do so yourself. Your main() function should be as small as possible and do almost nothing except creating your GtkApplication and running it. The "real work" should always be done in response to the signals fired by GtkApplication.
Loading "wiki.gnome.org"...
vs. local instance


Activities
Web
Sun 14:05
rupert
do you have some music recommendations?
sure, what genre?
14:01
could I borrow your car for a while?
what f
GtkApplication is the base
am from main().
main() is an operating system implementation detail that is really uninteresting to applications. The philosophy of GtkApplication is that applications are interested in being told what needs to happen, when it needs to happen, in response to actions from the user. The exact mechanism by which the operating system starts applications is uninteresting.
To this end, GtkApplication exposes a set of signals (or virtual functions) that an application should respond to.
startup: sets up the application when it first starts
shutdown: performs shutdown tasks
activate: shows the default first window of the application (like a new document). This corresponds to the application being launched by the desktop environment.
open: opens files and shows them in a new window. This corresponds to someone trying to open a document (or documents) using the application from the file browser, or similar.
When your application starts, the startup signal will be fired. This gives you a chance to perform initialisation tasks that are not directly related to showing a new window. After this, depending on how the application is started, either activate or open will be called next.
GtkApplication defaults to applications being single-instance. If the user attempts to start a second instance of a single-instance application then GtkApplication will signal the first instance and you will receive additional activate or open signals. In this case, the second instance will exit immediately, without calling startup or shutdown.
For this reason, you should do essentially no work at all from main(). All startup initialisation should be done in startup. This avoids wasting work in the second-instance case where the program just exits immediately.
The application will continue to run for as long as it needs to. This is usually for as long as there are any open windows. You can additionally force the application to stay alive using g_application_hold().
Loading "wiki.gnome.org"...

Activities

Sat 13:53

4*4

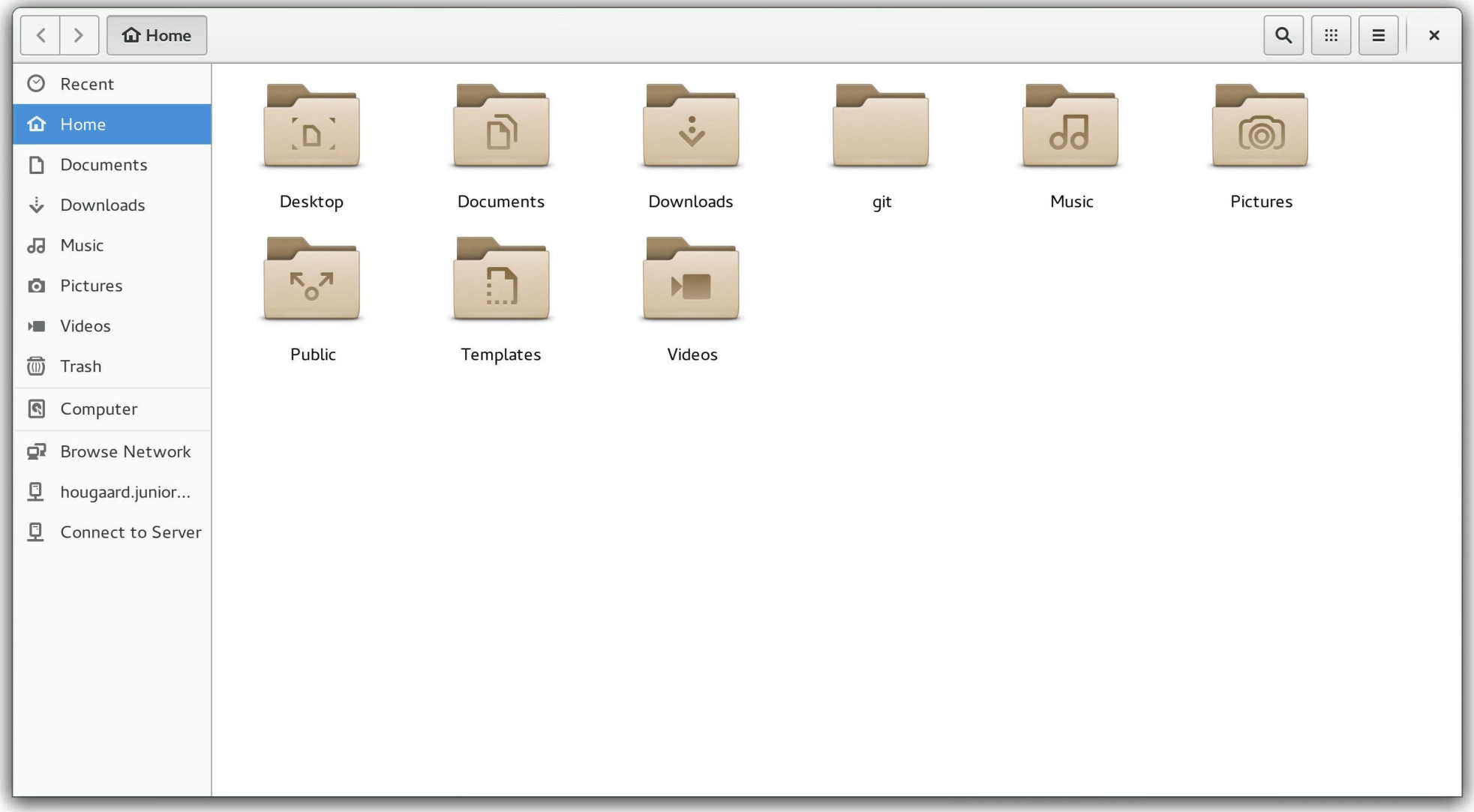
4*4 = 16

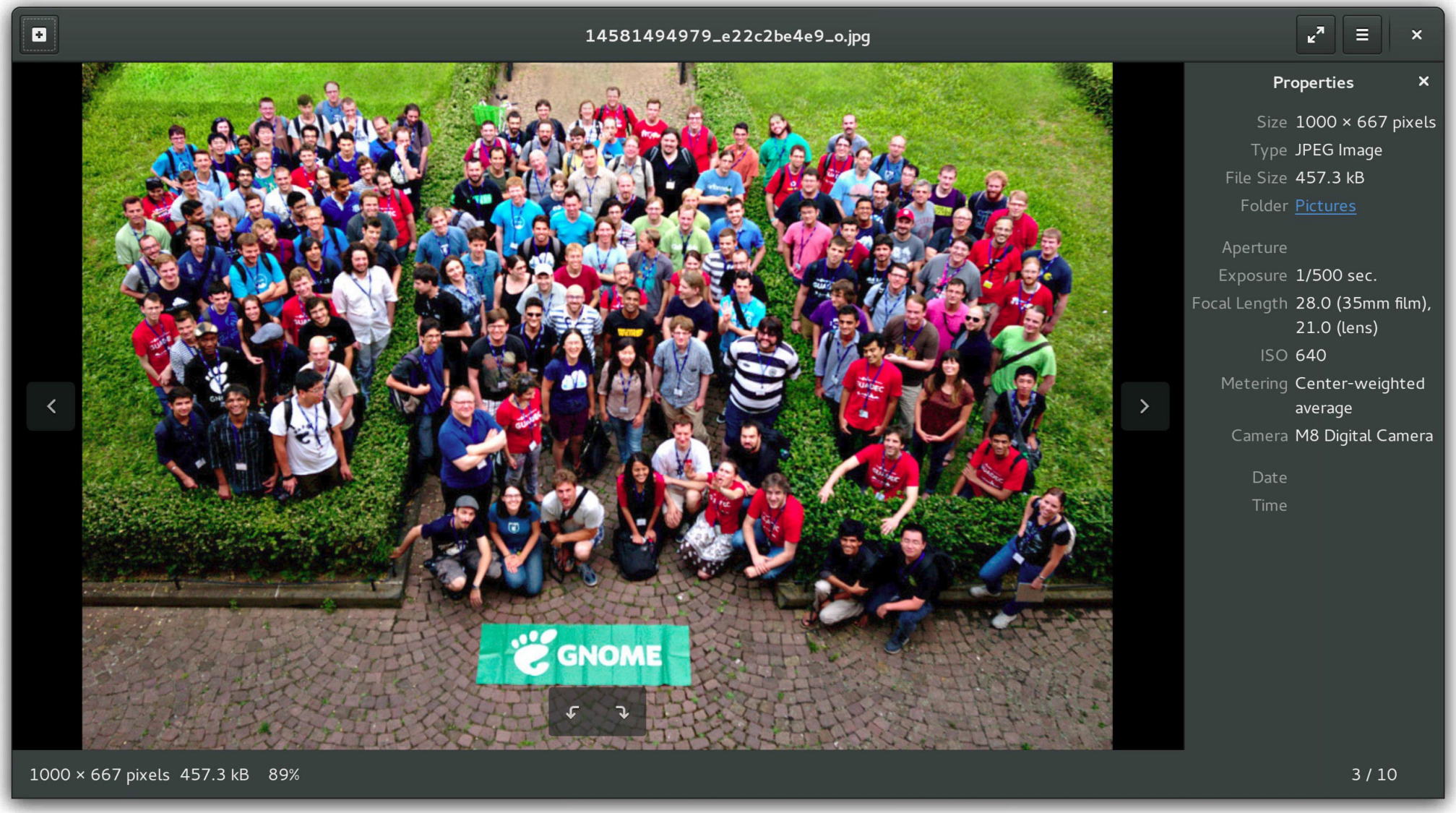
g8443

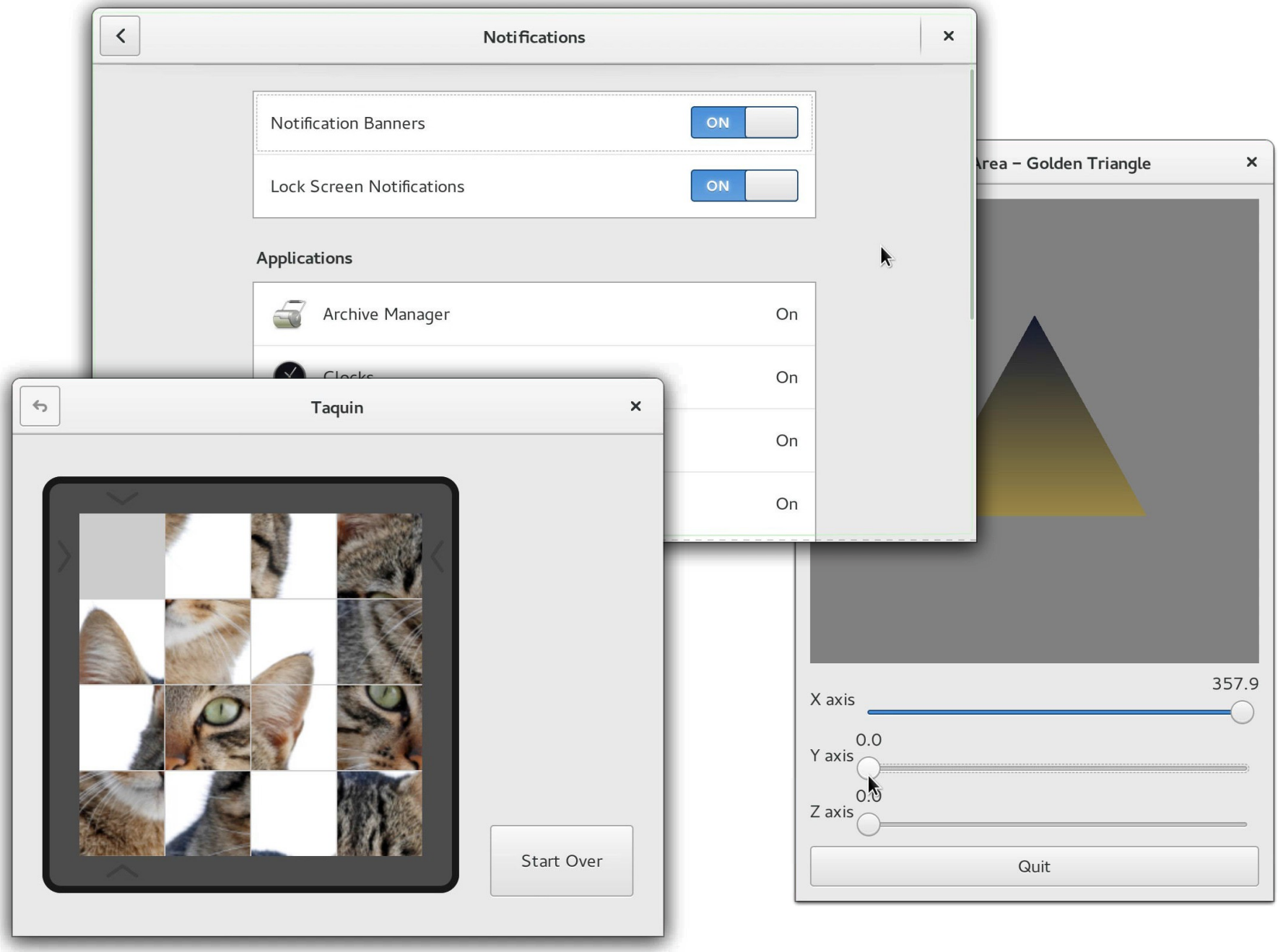
g8054

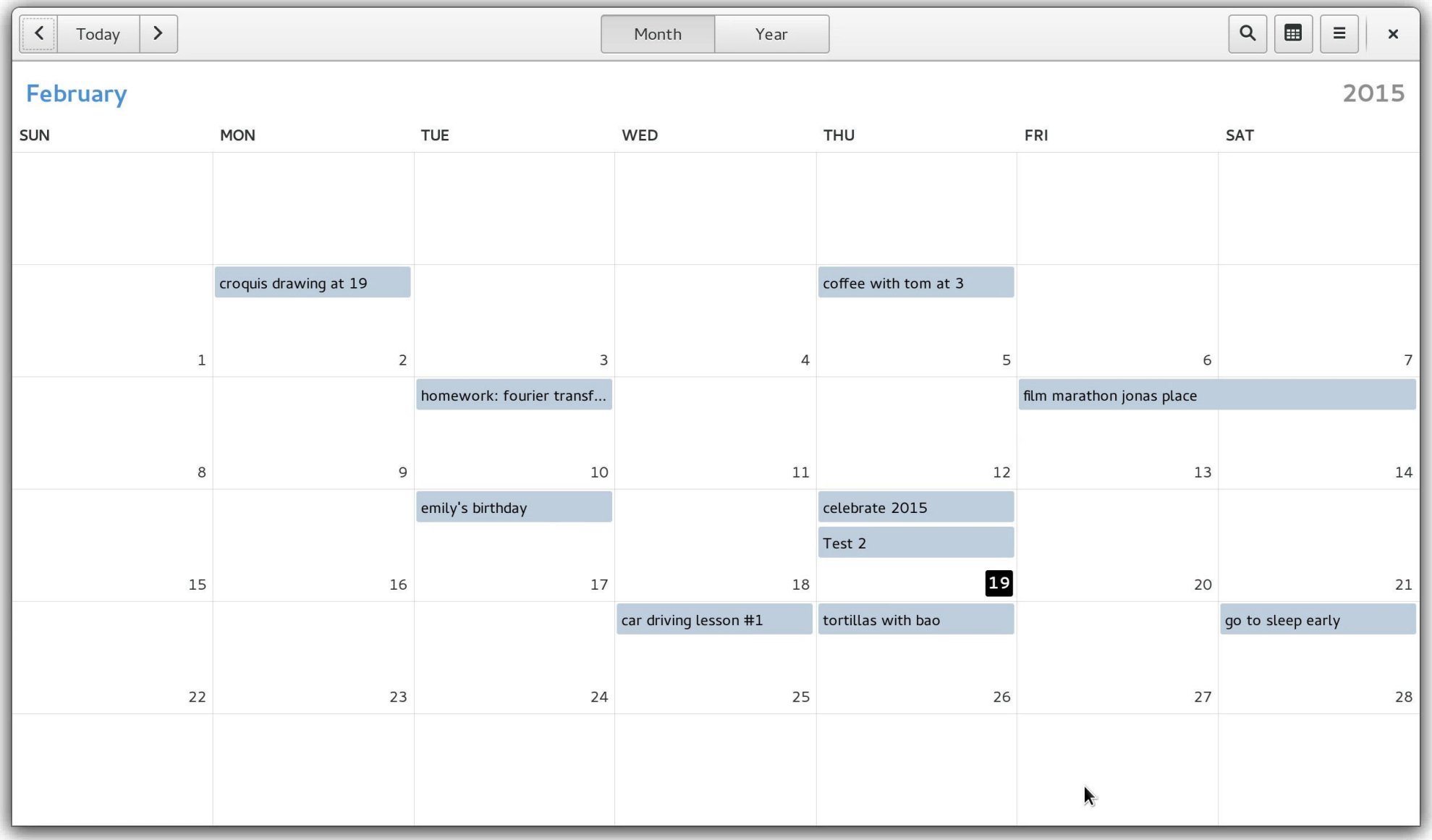
CIS224 Software Projects: Software Engineering and Research Methods

Search the Web for 4*4


Home
Recent
Home
Documents
Downloads
Music
Pictures
Videos
Trash
Computer
Browse Network
hougaard.junior...
Connect to Server
Desktop
Documents
Downloads
git
Music
Pictures
Public
Templates
Videos


14581494979_e22c2be4e9_o.jpg
Properties
Size 1000 × 667 pixels
Type JPEG Image
File Size 457.3 kB
Folder Pictures
Aperture
Exposure 1/500 sec.
Focal Length 28.0 (35mm film), 21.0 (lens)
ISO 640
Metering Center-weighted average
Camera M8 Digital Camera
Date
Time
GNOME
1000 × 667 pixels 457.3 kB 89%
3 / 10


Notifications
Notification Banners
ON
Lock Screen Notifications
ON
Applications
Archive Manager
On
On
On
On
Taquin
Start Over
rea – Golden Triangle
357.9
X axis
0.0
Y axis
0.0
Z axis
Quit

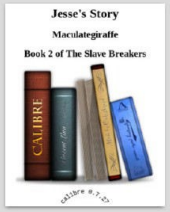

Today
Month
Year
February
2015
SUN
MON
TUE
WED
THU
FRI
SAT
croquis drawing at 19
coffee with tom at 3
homework: fourier transf...
film marathon jonas place
emily's birthday
celebrate 2015
Test 2
car driving lesson #1
tortillas with bao
go to sleep early
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28

grid-packing.c

```c
#include <gtk/gtk.h>


static void
activate (GtkApplication *app,
          gpointer        user_data)
{
  GtkWidget *window;
  GtkWidget *grid;
  GtkWidget *button;

  /* create a new window, and set its title */
  window = gtk_application_window_new (app);
  gtk_window_set_title (GTK_WINDOW (window), "Window");
  gtk_container_set_border_width (GTK_CONTAINER (window), 10);

  /* Here we construct the container that is going pack our buttons */
  grid = gtk_grid_new ();

  /* Pack the container in the window */
  gtk_container_add (GTK_CONTAINER (window), grid);

  button = gtk_button_new_with_label ("Button 1");
  g_signal_connect (button, "clicked", G_CALLBACK (print_hello), NULL);

  /* Place the first button in the grid cell (0, 0), and make it fill
   * just 1 cell horizontally and vertically (ie no spanning)
   */
  gtk_grid_attach (GTK_GRID (grid), button, 0, 0, 1, 1);

  button = gtk_button_new_with_label ("Button 2");
  g_signal_connect (button, "clicked", G_CALLBACK (print_hello), NULL);
```

Line 3, Column 1

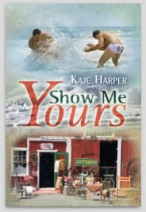
Recent | Collections



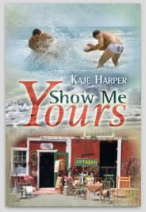
Jesse's Story
Maculategiraffe

The Social Cancer
JosÈ Rizal

Don't Read in the Closet: G...l Edition
authors, various

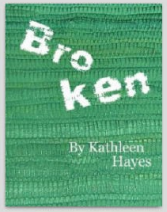


Show Me Yours
Harper, Kaje



Show Me Yours
Harper, Kaje

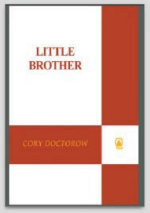
12090541



Hayes, Kathleen
Broken
O'Brien, Heather K

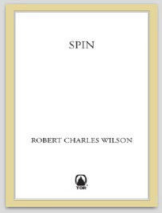
Stuff My Stocking: M-M Ro...Naughty
Authors, Various

MachineOfDeath_North_1372449942



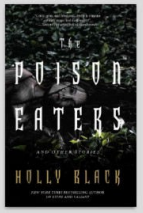
Little Brother
Doctorow, Cory

Shards of Honor (Vorkosigan Saga)
Lois McMaster Bujold



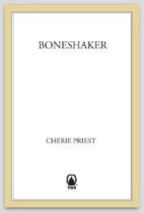
Spin
Wilson, Robert Charle...

The Last Unicorn
Beagle, Peter S.

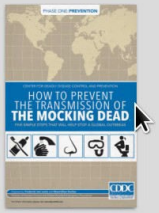


The Poison Eaters and Other Stories
Holly Black

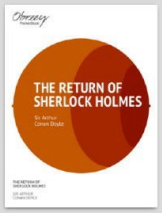
Just a Geek



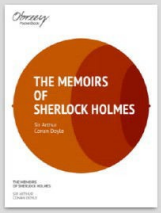
Boneshaker



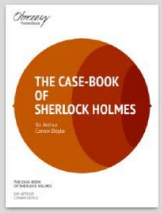
mockingdead_cbz_



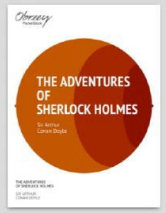
The Return of



The Memoirs of



The Case-Book of



The Adventures of

Aplikasi yang Dikembangkan GNOME™

https://wiki.gnome.org/Apps

- Abiword
- Banshee
- Cheese
- Dia
- Evince
- File Roller
- Gedit
- Hot SSH
- Iagno
- Klotski
- Logs
- ...



Proyek yang Dikembangkan GNOME™

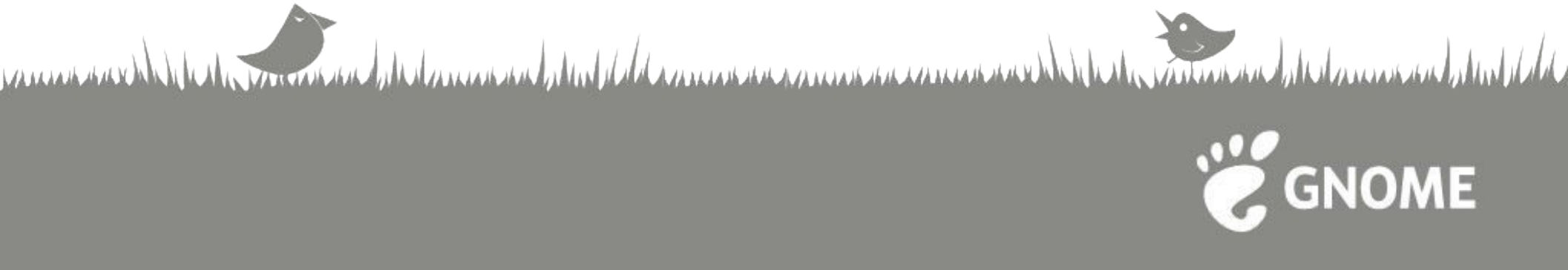
https://wiki.gnome.org/Project

- Cantarell
- GSstreamer
- GTK +
- Jhbuild
- Vala
- ...



GNOME™ Versi Berikutnya

- GNOME 3.17.x Development Series
  - https://wiki.gnome.org/ThreePointSeventeen
- GNOME 3.18
  - Rencana 23 September 2015



Tim dan Aktivitas di GNOME™

- https://wiki.gnome.org/Teams
  - Accessibility - works to make GNOME accessible to anyone, irrespective of their physical abilities.
  - Accounts - manages access to GNOME development systems.
  - Design - GNOME user interface and UX design.
  - Documentation - writes and maintains GNOME user and developer documentation.
  - Engagement - GNOME websites, blogging, press, marketing & promotion activities.
  - Foundation Board
  - Membership & Elections Committee - organizes the annual GNOME Foundation elections.
  - Maintainers Team - for App or Project maintainers
  - Moderators - stewards of the GNOME mailing lists.
  - Outreach - GNOME Outreach
  - QA - GNOME QA team
  - Bugsquad - maintains and manages the GNOME Bug Tracker.
  - Release Team - organizes and manages GNOME development.
  - Safety - User safety and privacy working group.
  - Sysadmin - maintain and develop GNOME's infrastructure.
  - Web Development
  - Translation - translates GNOME user interfaces and documentation into different languages.
  - User Groups - Regional GNOME user groups.

GNOME™ di Asia

- https://wiki.gnome.org/GnomeAsia
- Acara tahunan: GNOME.Asia Summit


KAZAKHSTAN
MONGOLIA
UZBEKISTAN
TURKMENISTAN
TURKEY
CHINA
SYRIA
IRAQ
IRAN
AFGHANISTAN
KASHMIR
PAKISTAN
NEPAL
BHUTAN
BANGLADESH
INDIA
MYANMAR (BURMA)
LAOS
THAILAND
VIETNAM
CAMBO
EGYPT
SAUDI ARABIA
OMAN
YEMEN
ERITREA
SUDAN
ETHIOPIA
SOMALIA
UGANDA
KENYA
TANZANIA
SEYCHELLES
COMOROS
MAYOTTE (FRANCE)
MOZAMBIQUE
ZAMBIA
ADAMAN AND NICOBAR ISLANDS (INDIA)
SRI LANKA
MALAYSIA
PHILIPPINES
INDONESIA
JAPAN
NORTH KOREA
SOUTH KOREA
2008
2014
Beijing
2013
2010
TaiWan
2012
HongKong
2011
Bangalore
2009
Ho Chi Minh
2015
Depok

GNOME™ Asia Summit

- Beijing, China (2008)
- Ho-Chi-Minh, Vietnam (2009)
- Taipei, Taiwan (2010)
- Bangalore, India (2011)
- Hong Kong (2012)
- Seoul, South Korea (2013)
- Beijing, China (2014)
- **Depok, Indonesia (2015)**

GNOME

GNOME™ Asia Summit 2015

- http://2015.gnome.asia/
- 322 peserta
- 48 pembicara
- 28 diantaranya pembicara lokal

GNOME.™ ASIA summit 2015

May 7-9, 2015 | Depok, Indonesia

GNOME™ Asia Summit 2015

GNOME

TERIMA KASIH

GNOME