



GNU+KDE como entorno de desarrollo

Aleix Pol Gonzàlez

March 29, 2008



- Estudio Ingeniería Informática en la FIB/UPC
- Colaboro en KDE con KAlgebra en KDE-Edu i KDevelop principalmente



- Estudio Ingeniería Informática en la FIB/UPC
- Colaboro en KDE con KAlgebra en KDE-Edu i KDevelop principalmente



!!!Humanos!!!

Producimos código erróneo

Esperanza

- Tenemos que (poder) corregir los errores
- Tenemos que (poder) minimizar el impacto de los bugs
- Tenemos que producir el mínimo número de errores posible



!!!Humanos!!!

Producimos código erróneo

Esperanza

- Tenemos que (poder) corregir los errores
- Tenemos que (poder) minimizar el impacto de los bugs
- Tenemos que producir el mínimo número de errores posible



!!!Humanos!!!

Producimos código erróneo

Esperanza

- Tenemos que (poder) corregir los errores
- Tenemos que (poder) minimizar el impacto de los bugs
- Tenemos que producir el mínimo número de errores posible



!!!Humanos!!!

Producimos código erróneo

Esperanza

- Tenemos que (poder) corregir los errores
- Tenemos que (poder) minimizar el impacto de los bugs
- Tenemos que producir el mínimo número de errores posible



!!!Humanos!!!

Producimos código erróneo

Esperanza

- Tenemos que (poder) corregir los errores
- Tenemos que (poder) minimizar el impacto de los bugs
- Tenemos que producir el mínimo número de errores posible



1 Introducción

2 Edición

- Editores

- Versionado

3 Debug

- gdb

- Valgrind

4 Integración

- KDevelop

5 Preguntas



Quién escribe tu código?

- Kate
- emacs, vim, ¿tu_editor_favorito?



Quién escribe tu código?

- Kate
- emacs, vim, ¿tu_editor_favorito?



Quién escribe tu código?

- Kate
- emacs, vim, ¿tu_editor_favorito?



1 Introducción

2 Edición

- Editores

- **Versionado**

3 Debug

- gdb

- Valgrind

4 Integración

- KDevelop

5 Preguntas



Supervivencia en el tiempo

- Historial de versiones
- Compartición de flujo de trabajo
- bzd, svn, git



Supervivencia en el tiempo

- Historial de versiones
- Compartición de flujo de trabajo
- bzd, svn, git



Supervivencia en el tiempo

- Historial de versiones
- Compartición de flujo de trabajo
- bzd, svn, git



Supervivencia en el tiempo

- Historial de versiones
- Compartición de flujo de trabajo
- bzd, svn, git



1 Introducción

2 Edición

- Editores
- Versionado

3 Debug

- gdb
- Valgrind

4 Integración

- KDevelop

5 Preguntas



La cucaracha

- Permite debug por pasos
- watchpoints
- catchpoints
- Frontends: ddd, KDbg, KDevelop



La cucaracha

- Permite debug por pasos
- watchpoints
- catchpoints
- Frontends: ddd, KDbg, KDevelop



La cucaracha

- Permite debug por pasos
- watchpoints
- catchpoints
- Frontends: ddd, KDbg, KDevelop



La cucaracha

- Permite debug por pasos
- watchpoints
- catchpoints
- Frontends: ddd, KDbg, KDevelop



La cucaracha

- Permite debug por pasos
- watchpoints
- catchpoints
- Frontends: ddd, KDbg, KDevelop



- 1 Introducción
- 2 Edición
 - Editores
 - Versionado
- 3 **Debug**
 - gdb
 - **Valgrind**
- 4 Integración
 - KDevelop
- 5 Preguntas



El cocal

- Inicialmente una herramienta para detección de *leaks*
- Entorno de trabajo para analizadores de código



El cocal

- Inicialmente una herramienta para detección de *leaks*
- Entorno de trabajo para analizadores de código



El cugal

- Inicialmente una herramienta para detección de *leaks*
- Entorno de trabajo para analizadores de código



Encontrando errores

- Herramienta para comprobaciones de memoria
- Controla usos de memoria no inicializada
- Controla usos incorrectos en el *heap*
- El módulo exp-omega con comprobaciones adicionales.



Encontrando errores

- Herramienta para comprobaciones de memoria
- Controla usos de memoria no inicializada
- Controla usos incorrectos en el *heap*
- El módulo exp-omega con comprobaciones adicionales.



Encontrando errores

- Herramienta para comprobaciones de memoria
- Controla usos de memoria no inicializada
- Controla usos incorrectos en el *heap*
- El módulo exp-omega con comprobaciones adicionales.



Encontrando errores

- Herramienta para comprobaciones de memoria
- Controla usos de memoria no inicializada
- Controla usos incorrectos en el *heap*
- El módulo *exp-omega* con comprobaciones adicionales.



Encontrando errores

- Herramienta para comprobaciones de memoria
- Controla usos de memoria no inicializada
- Controla usos incorrectos en el *heap*
- El módulo exp-omega con comprobaciones adicionales.



¿Qué hace nuestro programa?

- Cachegrind: Hace "profiling" de los accesos a la cache
- Callgrind: "Profiling" de llamadas.
- KCachegrind: Analisis de los resultados



¿Qué hace nuestro programa?

- Cachegrind: Hace "profiling" de los accesos a la cache
- Callgrind: "Profiling" de llamadas.
- KCachegrind: Analisis de los resultados



¿Qué hace nuestro programa?

- Cachegrind: Hace "profiling" de los accesos a la cache
- Callgrind: "Profiling" de llamadas.
- KCachegrind: Analisis de los resultados



¿Qué hace nuestro programa?

- Cachegrind: Hace "profiling" de los accesos a la cache
- Callgrind: "Profiling" de llamadas.
- KCachegrind: Analisis de los resultados



- Hellgrind: Detección de posibles *races*
- Massif: Análisis de uso de memoria (a usar con `ms_print`)
- Diferentes herramientas de ejemplo (e.g. `lackey`)



- Hellgrind: Detección de posibles *races*
- Massif: Análisis de uso de memoria (a usar con `ms_print`)
- Diferentes herramientas de ejemplo (e.g. `lackey`)



- Hellgrind: Detección de posibles *races*
- Massif: Análisis de uso de memoria (a usar con `ms_print`)
- Diferentes herramientas de ejemplo (e.g. `lackey`)



- Hellgrind: Detección de posibles *races*
- Massif: Análisis de uso de memoria (a usar con `ms_print`)
- Diferentes herramientas de ejemplo (e.g. `lackey`)



1 Introducción

2 Edición

- Editores
- Versionado

3 Debug

- gdb
- Valgrind

4 Integración

- KDevelop

5 Preguntas



El cosmos

- Es un entorno integrado, no el α y ω (como MSVC)
- Edición integrada: resaltado inteligente, autocompletación
- Navegación inteligente
- Debugger integrado (gdb, valgrind)



El cosmos

- Es un entorno integrado, no el α y ω (como MSVC)
- Edición integrada: resaltado inteligente, autocompletación
- Navegación inteligente
- Debugger integrado (gdb, valgrind)



El cosmos

- Es un entorno integrado, no el α y ω (como MSVC)
- Edición integrada: resaltado inteligente, autocompletación
- Navegación inteligente
- Debugger integrado (gdb, valgrind)



El cosmos

- Es un entorno integrado, no el α y ω (como MSVC)
- Edición integrada: resaltado inteligente, autocompletación
- Navegación inteligente
- Debugger integrado (gdb, valgrind)



El cosmos

- Es un entorno integrado, no el α y ω (como MSVC)
- Edición integrada: resaltado inteligente, autocompletación
- Navegación inteligente
- Debugger integrado (gdb, valgrind)



Alguna pregunta?

Konqi wants you!

