

Programming Summer Camp For Japanese Students

LinuxCon Japan 2011

June 3, 2011

Satoru Takeuchi
<takeuchi_satoru@jp.fujitsu.com>

Contents

- Background
- What is Security and Programming Camp?
- What linux kernel course did?
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

Contents

- **Background**
- What is Security and Programming Camp?
- What linux kernel course did?
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

Background

- I think linux kernel community needs to promote new-ages.
- There are two kind of information which make me think so.
 1. Statistical information
 2. Visual information

1. Statistical information

- I investigated how long the active developers have been in linux kernel community.
- The data source is the full git history of Linux.

<http://lwn.net/Articles/285366/>



The screenshot shows the LWN.net website interface. At the top left is the LWN.net logo with the tagline "Linux info from the source". A navigation menu includes links for "Weekly edition", "Kernel", "Security", "Distributions", "Contact Us", and "Search", with sub-links for "Archives", "Calendar", "Subscribe", "Write for LWN", "LWN.net FAQ", and "Sponsors". Below the navigation is a sidebar with "Ads by Google" for a "paper band filter" and a "Not logged in" section with links for "Log in now", "Create an account", and "Subscribe to LWN". The main content area features the article title "Full git history of Linux" and the following metadata:

From: "yoann paci oleau" <yoann.paci.oleau@gmail.com>
To: linux-kernel@vger.kernel.org
Subject: Full git history of Linux
Date: Fri, 6 Jun 2008 10:50:48 -0500
Message-ID: <390282f40806060850v106f8defid2694d4de6ba2135@mail.gmail.com>
Cc: kernel-janitors@vger.kernel.org
Archive-link: [Article](#), [Thread](#)

The article body begins with "Hi," followed by a paragraph: "If anyone is interested, I have put on my website[1] a git repository containing the full history of Linux, from 0.01 to now. It's built from 3 other git repositories: - the one from Dave Jones from 0.01 to 2.4.0, - the one from tgix from 2.4.0 to 2.6.12, - the one from Linus Torvalds from 2.6.12 to now." The text continues with "I used the 'graft' feature of git (thanks to Junio and people on #git for the tip) to link them together. I also modified (via a git-filter-branch) the dates of some commits as for instance all commits from the Dave Jones's repo had the same date (20 Nov 2007). For this I initially used the following code:"

1. Statistical information

- Here are the top 10 active developers.

name	# of patches per year	first contribution [year]
Linus Torvalds	1421	2002
Andrew Morton	751	2002
Al Viro	716	2002
David S. Miller	669	2002
Russell King	372	2002
Greg Kroah-Hartman	371	2002
Ingo Molnar	358	2002
Takashi Iwai	295	2002
Thomas Gleixner	277	2004
Tejun Heo	267	2004

1. Statistical information

- Here are the top 10 active developers.

name	# of patches per year	first contribution [year]
Linus Torvalds	1421	2002
Andrew Morton	751	2002
		2002
		2002
		2002
		2002
		2002
Takashi Iwai	295	2002
Thomas Gleixner	277	2004
Tejun Heo	267	2004

All of them have been on linux kernel community since early days!

1. Statistical information

- How much the ratio of newcomers among the core active developers?

	The ratio
top 10	0%
top 20	5%
top 30	6.7%
top 50	24%
top 100	32%

2. Visual information



- Here's the group photo of 2009's kernel summit and is on an LWN's article.
<http://lwn.net/Articles/357651/>
- One of its comment says...

2. Visual information

Kernel summit group photo

Posted Oct 20, 2009 4:49 UTC (Tue) by **proski** (subscriber, #104) [[Link](#)]

No women this time :-(

Kernel summit group photo

Posted Oct 20, 2009 8:19 UTC (Tue) by **lkandrak** (subscriber, #43452) [[Link](#)]

And no young people either! :)

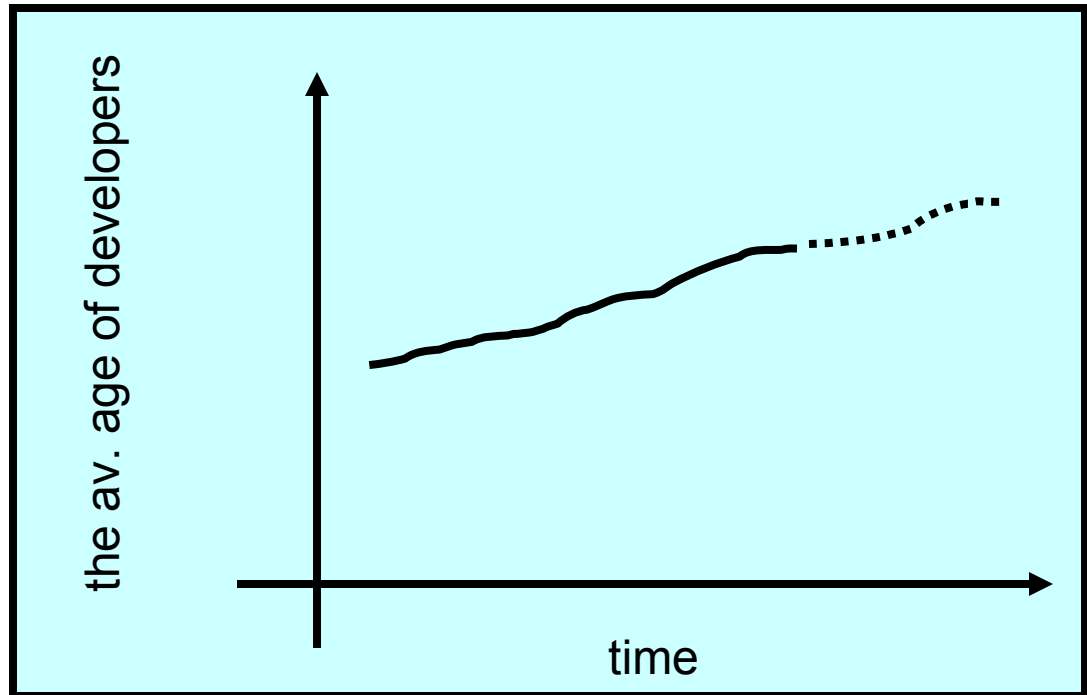
Kernel summit group photo

Posted Oct 20, 2009 9:08 UTC (Tue) by **thomasnoel** (subscriber, #46836) [[Link](#)]

Err ? I see at least 75 young people ;)

Why kernel community need to promote new-ages.

- The average age of core developers would be increasing gradually.



A solution

- Linux kernel community needs to promote new generation hackers.
- There is one of the solution. It's the way of Security and Programming Camp.

Contents

- Background
- **What is Security and Programming Camp?**
- What linux kernel course did?
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

What is Security and Programming Camp

- In Japan, there is an annual Security and Programming Camp (SPCamp) since 2004.

集え、コンピュータの未来を守るIT戦士たち。
セキュリティ&プログラミングキャンプ2010

4泊5日 交通費・宿泊費を含め、無料です。 **参加無料!**
2010.08.12 - 08.16
参加資格：22歳以下の学生・生徒

主催 独立行政法人情報処理推進機構/IPA
共催 後援 経済産業省 文部科学省 財団法人日本情報処理開発協会(JIPDEC) 特定非営利活動法人日本ネットワークセキュリティ協会(UNSA)

ホーム 概要 講義内容 講師・実行委員 応募要領 応募用紙 よくある質問

Security & Programming Camp 2010

未来を守る力キを見つけろ。

INFORMATION 事務局からのお知らせ

2010年12月21日
「セキュリティ&プログラミングキャンプ2010実施報告会」終了しました
12月18日(土)に開催しました「セキュリティ&プログラミングキャンプ2010実施報告会」は100名以上の方にご参加いただき、無事終了いたしました。ご協力いただきました皆様へ感謝いたします。

2010年11月18日
12月18日(土)に「セキュリティ&プログラミングキャンプ2010実施報告会」を開催します

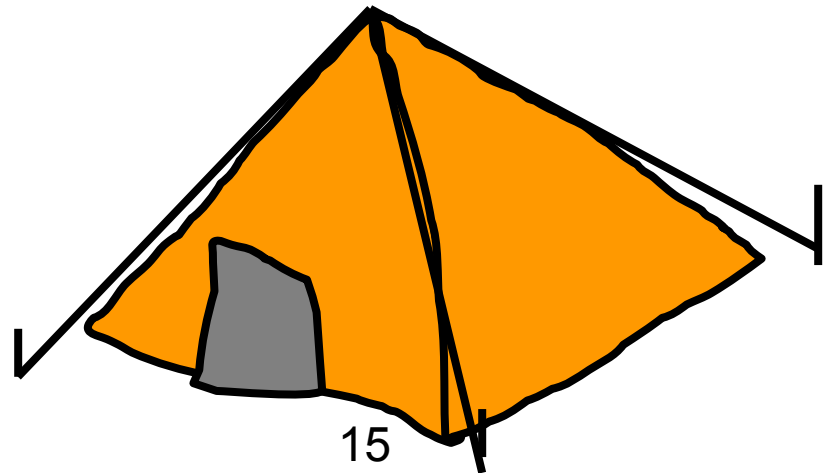
過去の
キャンプ
レポート

- 2010 Camp Report
- 2009 Camp Report
- 2008 Camp Report
- 2007 Camp Report
- 2006 Camp Report
- 2005 Camp Report
- 2004 Camp Report

セキュリティ&プログラミングキャンプ2010事務局
〒113-6591 東京都文京区本駒込2-28-1
情報処理推進機構 17人研修棟5F 基本研修棟5F3号室
TEL: 03-5978-7536(平日の10:00~18:00) / FAX: 03-5978-7516
E-mail: spcamp2010@ipa.go.jp(問い合わせ専用)

What is Security and Programming Camp?

- Purpose: promoting skilled computer engineers
- Target: 22-year-old or less students interested in computer.
- There are several courses: One of them is linux kernel course.
- Partly sponsored by Japanese Government and it's free of charge.

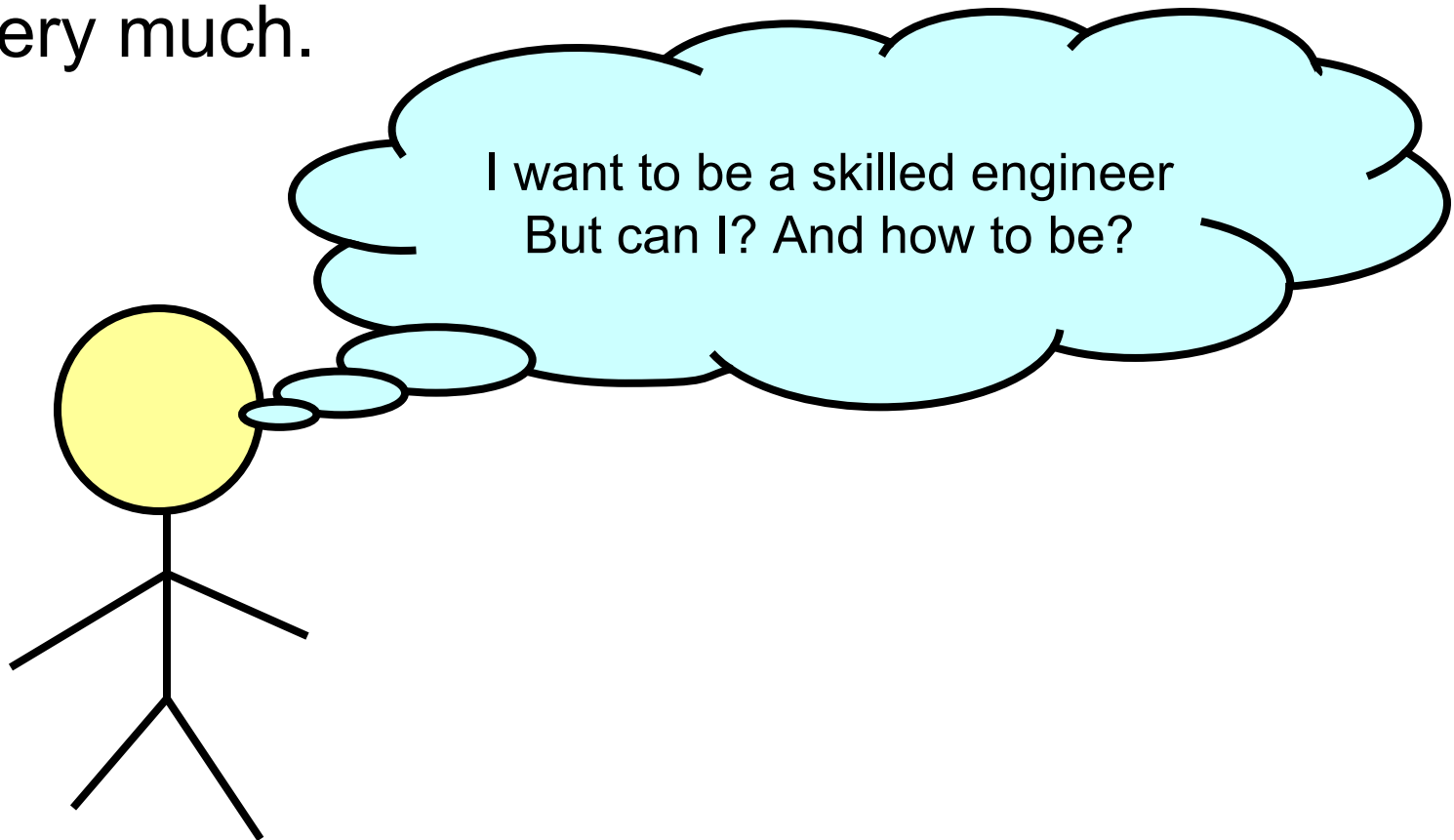


What is Security and Programming Camp?

- SPCamp has succeeded to promote students by the following way for many years.
 - All trainers are the skilled engineers.
 - Valuing hands-on training rather than classroom learning.
 - Working on the training all day long for five days in one location.

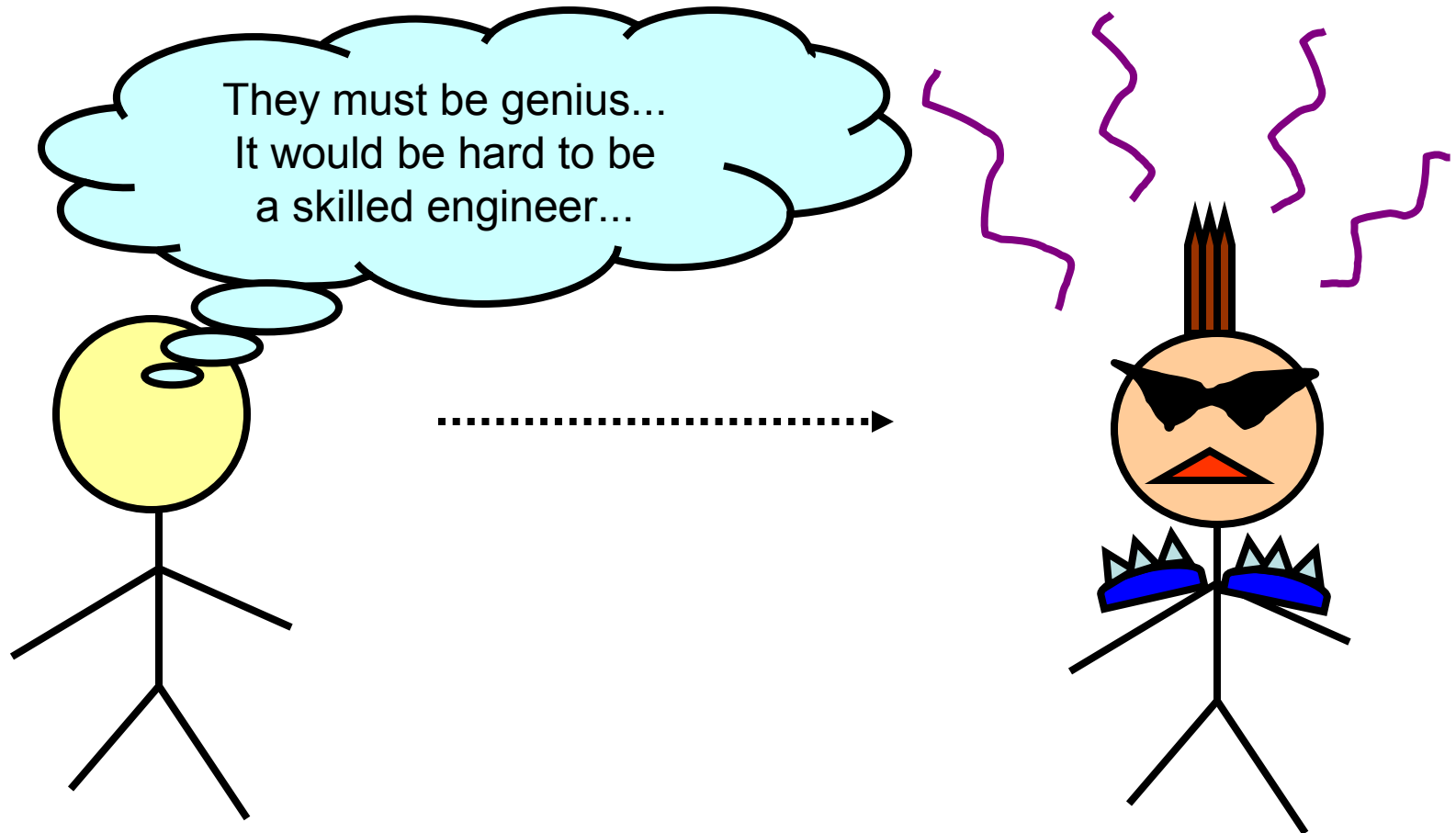
Why it works?

- There are many students who love computer very much.



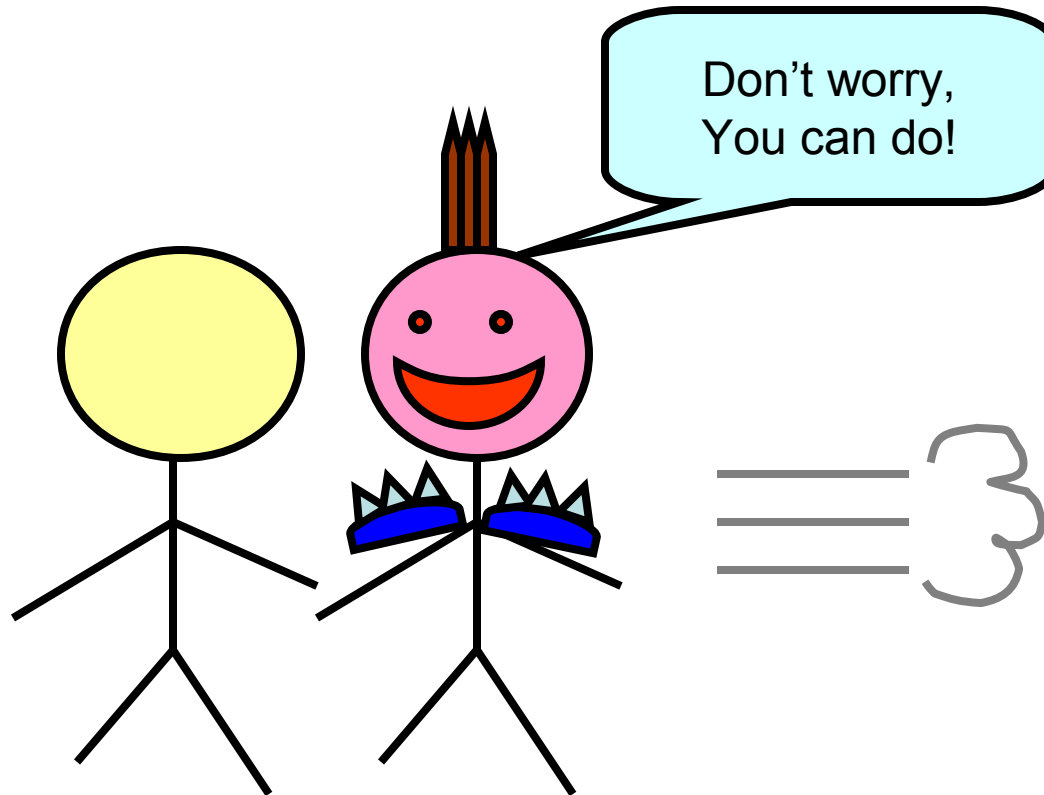
Why it works?

- From their view, the skilled engineers are a bit scary.



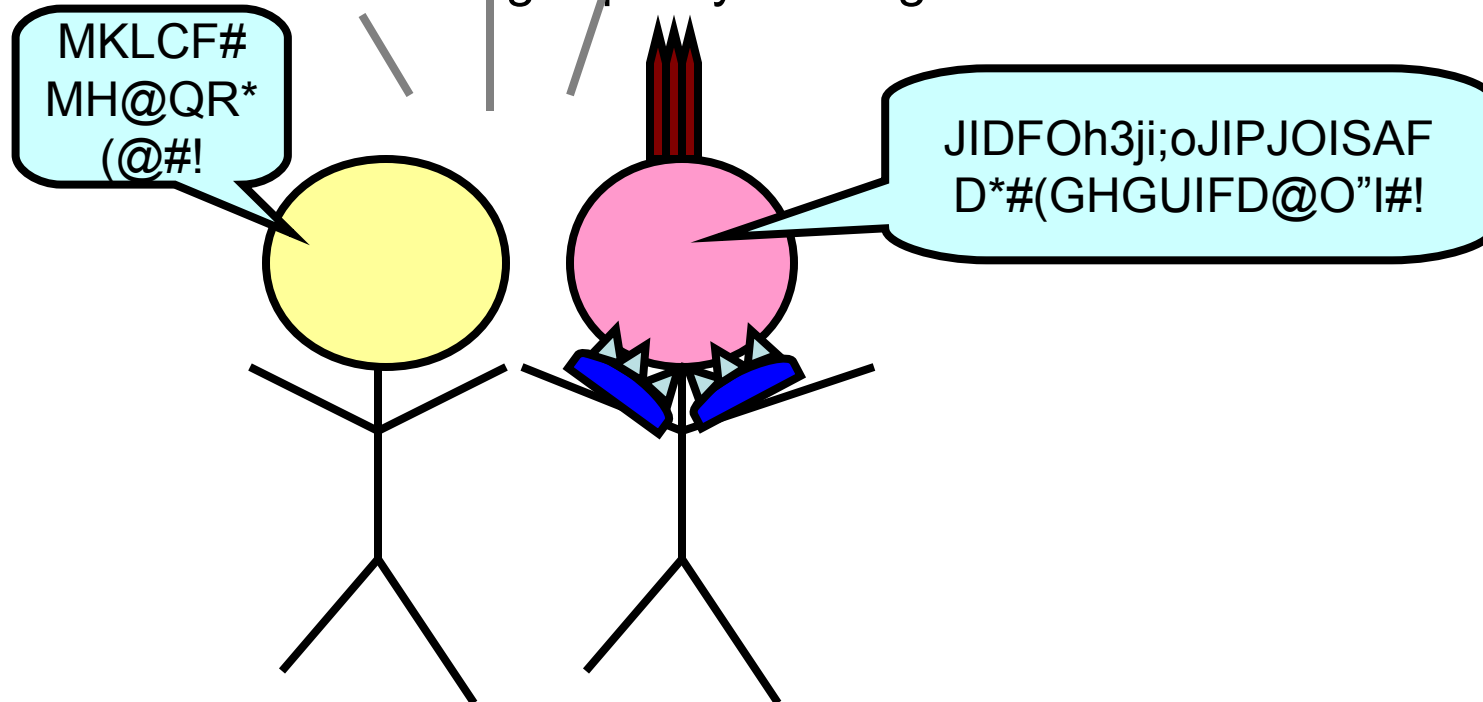
Why it works?

- SPCamp provides the opportunity introducing such students to skilled engineers.



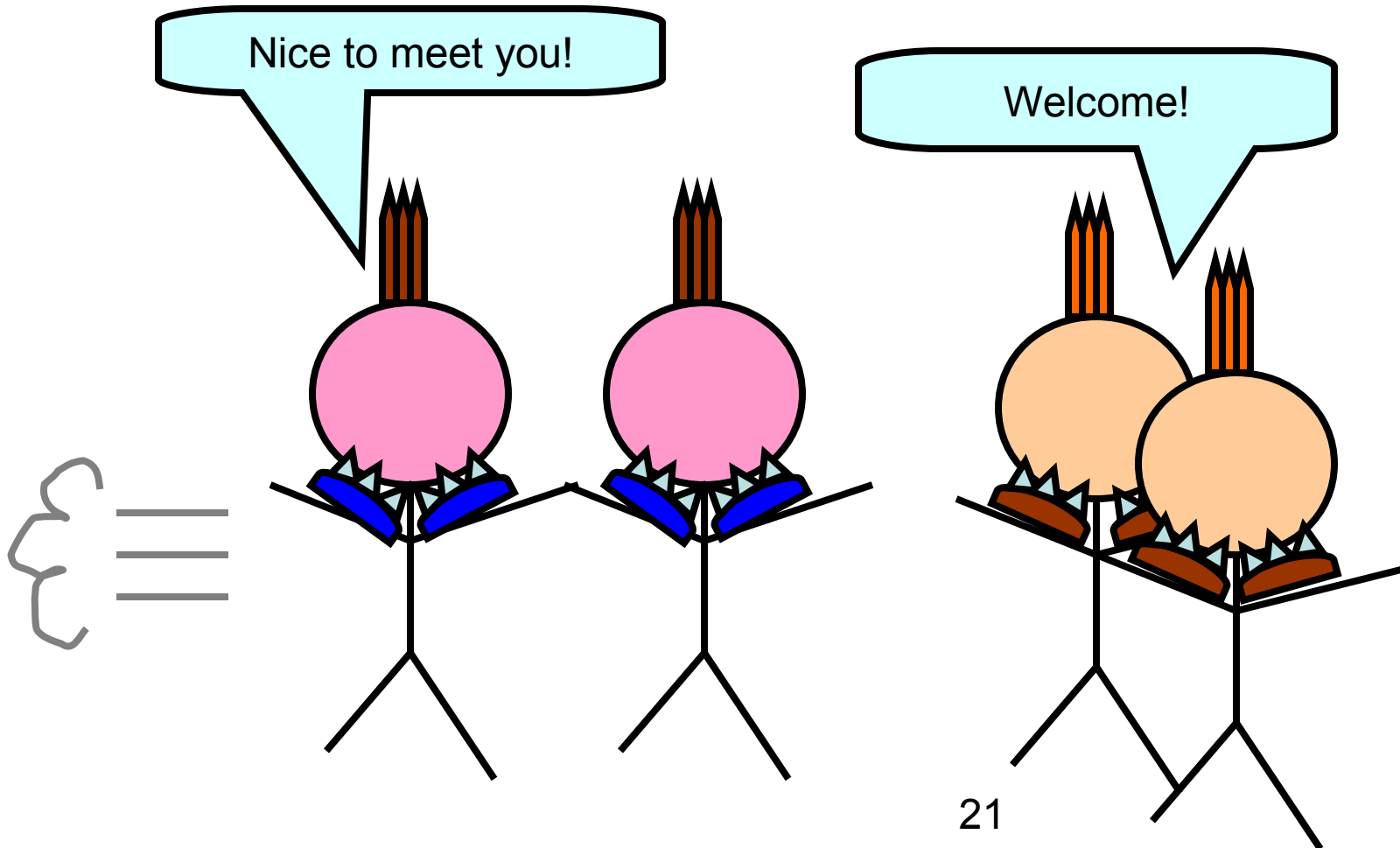
Why it works?

- During SPCamp, trainers(skilled engineers) promote trainees(students) carefully and severely.
- Since each course consists of at most 10 trainers and 10 trainees, it can achieve high-quality training.



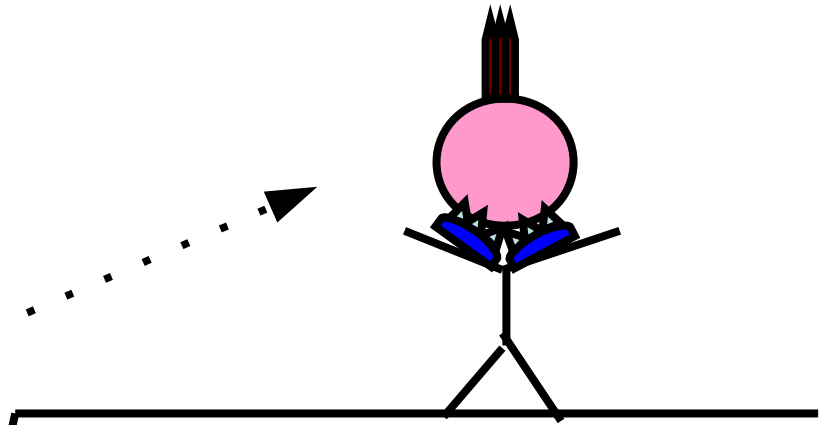
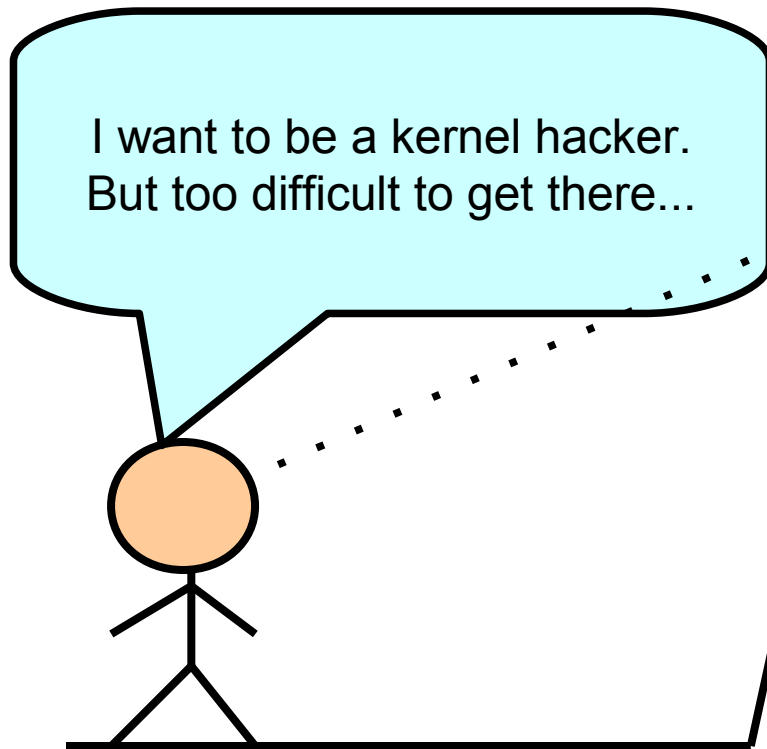
Why it works?

- When it ends, new skilled engineers are born.



Why it works for linux?

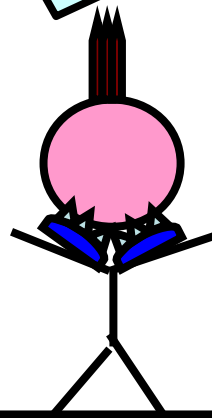
- There is a gap between students and linux kernel hackers.



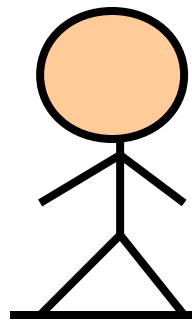
- Hardware knowledge
- Software(OS) knowledge
- Linux conventions

Why it works for linux?

- Kernel hackers can tell them a easy way to get to linux kernel community.



No, it's not so hard!



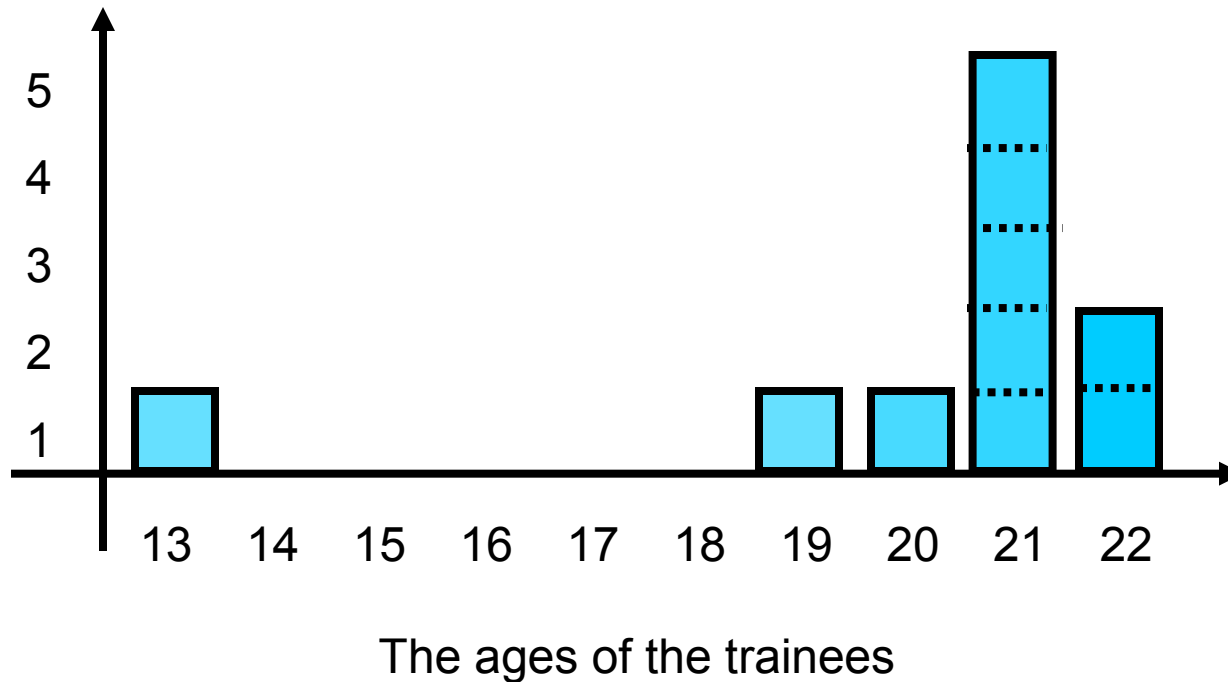
- Minimum Hardware knowledge
- Minimum Software(OS) knowledge
- Minimum Linux kernel conventions

Contents

- Background
- What is Security and Programming Camp?
- **What linux kernel course did?**
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

The members of linux kernel course

- The linux kernel course consists of eight trainers and 10 trainees



The members of linux kernel course

- All trainers are linux kernel hacker.
 - Four of us are subsystem maintainer.
 - We belong to various companies and universities respectively and gathered in SPCamp to train newbies.



Who are trainers?(1/2)

1. **[The Leader] Hideaki YOSHIFUJI**: one of a network subsystem maintainer
2. **Hiroyuki KAMEZAWA**: one of a maintainer of memory resource controller
3. **Hidehiro KAWAI**: dump, tracer, filesystem developer
4. **Motohiro KOSAKI**: core memory subsystem developer

The members of linux kernel course

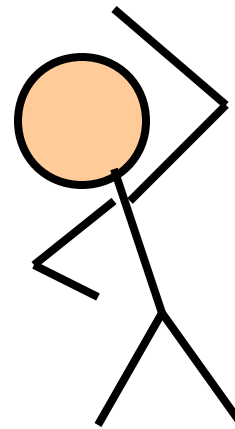
- The trainers of Linux course (2/2)
 5. **Satoru TAKEUCHI**: CPU/PCI hotplug and task scheduler developer
 6. **Masami HIRAMATSU**: one of maintainer of kprobe
 7. **Hiroshi MIURA**: the representative director of OSM Japan
 8. **Isaku YAMAHATA**: virtualization developer. The maintainer of Xen/ia64.

Contents

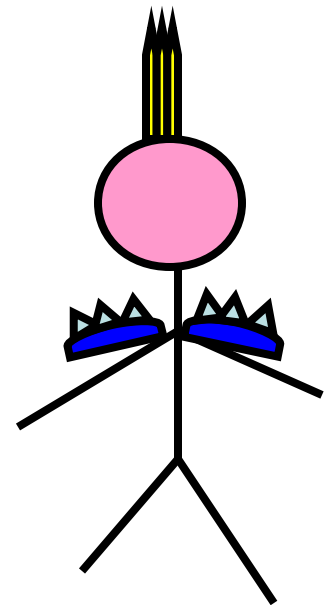
- Background
- What is Security and Programming Camp?
- **What linux kernel course did?**
 - **One month ago-Day0: Limbering up**
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

One month ago-Day0: Limbering up

- It was a mailing-list-base discussion.
- This step consists of the following two works:
 1. Getting basic knowledge by reading textbooks.
ex) Robert Love's "Linux Kernel Development 3rd".
 2. Trying several subjects which trainers provide.

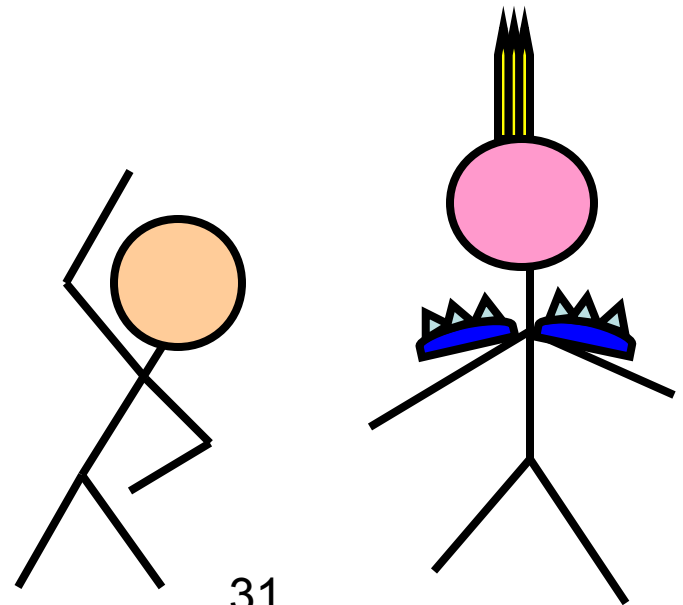


30



One month ago-Day0: Limbering up

- The subjects which trainees tried to:
 - Compiling their own kernel.
 - Making a simple change to the kernel.
 - Learning linux community conventions

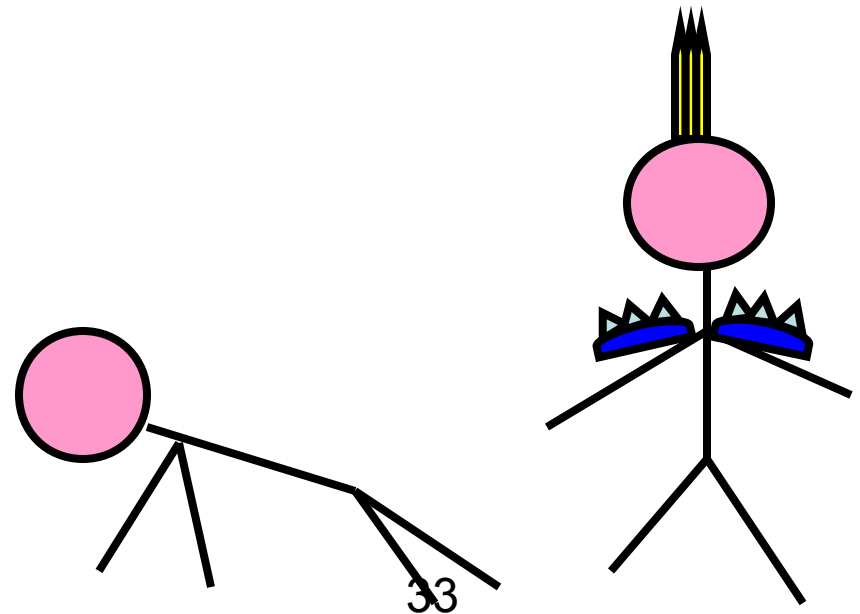


Contents

- Background
- What is Security and Programming Camp?
- **What linux kernel course did?**
 - One month ago-Day0: Limbering up
 - **Day1-Day3: Strength training**
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- Conclusion

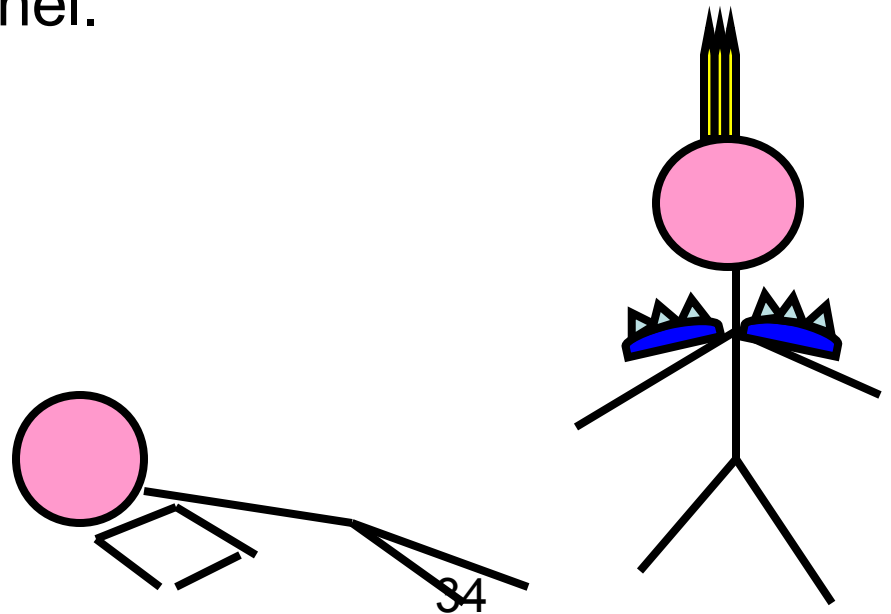
Day1-Day3: Strength training

1. Ice-breaking: Self-introduction and some recreations.
2. Common lectures.
3. Linux course specific lectures.



Day1-Day3: Strength training

- Linux course specific lectures:
 1. The base of computer architecture and operating system.
 2. Learning linux kernel via developing kernel module.
 3. How to debug linux kernel.



1. The base of computer architecture and operating system

[Computer Architecture]

Register
Memory
Storage
Interrupt
Cache Memory
SMP/NUMA

...

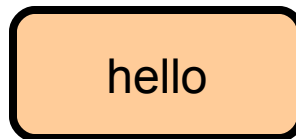
[Operating System]

Task scheduler
Memory management
Process management
Signal
File System

...

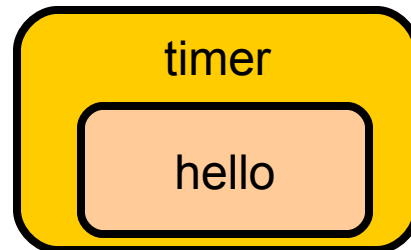
2. Learning linux kernel via developing kernel module

1. Make the simplest hello world module.



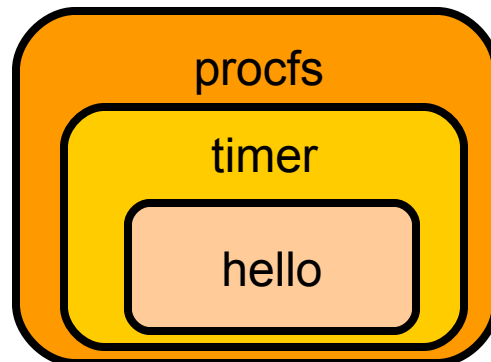
2. Learning linux kernel via developing kernel module

1. Make the simplest hello world module.
2. Add a timer to this module.



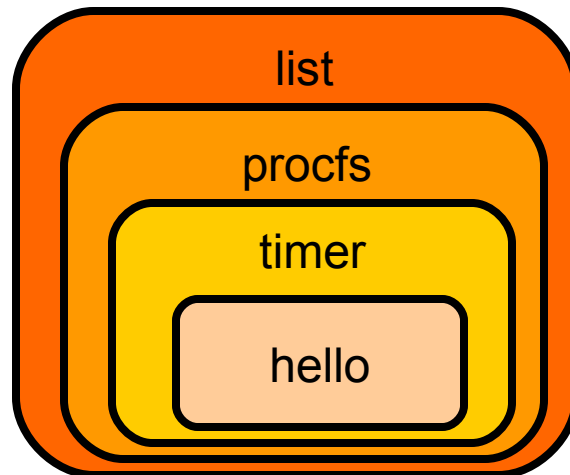
2. Learning linux kernel via developing kernel module

1. Make the simplest hello world module.
2. Add a timer to this module.
3. Add /proc interface to this timer.



2. Learning linux kernel via developing kernel module

1. Make the simplest hello world module.
2. Add a timer to this module.
3. Add /proc interface to the timer.
4. Add a stack data structure by using struct list.



How to debug linux kernel

- Learning how to debug linux kernel.
- Introduce many debug tools and use some of them.
 - test suite: LTP
 - bug detector: lockdep, kmemcheck
 - printk :-)
 - tracer: perf probe, systemtap
 - debuggder: gdb, kgdb
 - crash dump: kdump, crash



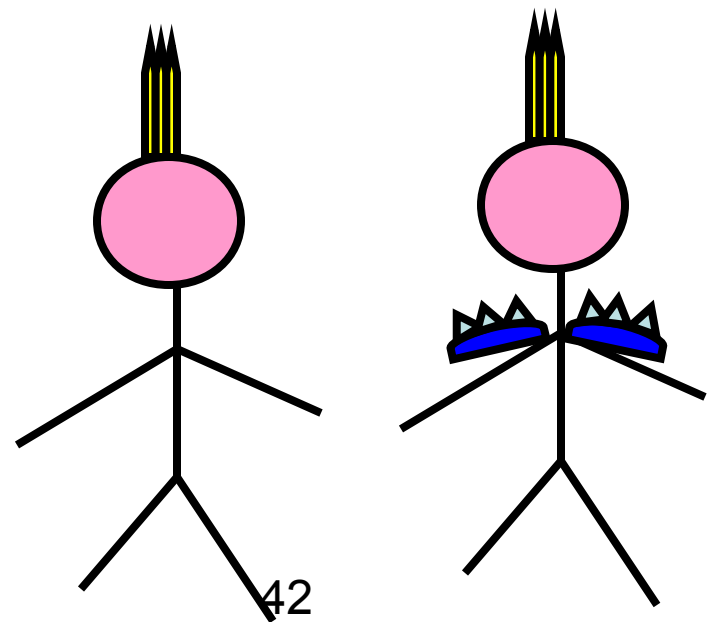
You Could Learn A Lot From The LTP.
www.linuxtestproject.org

Contents

- Background
- What is Security and Programming Camp?
- **What linux kernel course did?**
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - **Day4: Actual combat. Hack! Hack! Hack!**
 - Day5: Progress report
- The result
- Conclusion

Day4: Actual combat. Hack!Hack!Hack!

- The trainers provided some hacking themes.
- Each trainee selected one of a theme and worked on it all day long.



Day4: Actual combat. Hack!Hack!Hack!

- Here are the themes (1/2):
 1. Fix a bug on kprobe
 2. Fix a bug on “make localmodconfig”
 3. Estimate kernel compilation time: kbuild hack
 4. Visualize panic messages: video RAM hack

Day4: Actual combat. Hack!Hack!Hack!

- Here are the themes (2/2):
 5. Kernel space file shredder to delete confidential file safely and completely.
 6. Improving shared eventfd
 7. Generate the sounds on printk
 8. Speed up ps command.

Day4: Actual combat. Hack!Hack!Hack!

- On day4, two of trainees posted his own patch and these were applied to upstream kernel!
 - Shuhei KUMANO: Fix a bug on kprobe
 - Hiromu YAKURA: Fix a bug on “make localmodconfig”

Fix a bug on kprobe

commit 737480a0d525dae13306296da08029dff545bc72

Author: KUMANO Syuhei <kumano.prog@gmail.com>

Date: Sun Aug 15 15:18:04 2010 +0900

kprobes/x86: Fix the return address of multiple kretprobes

Fix the return address of subsequent kretprobes when multiple kretprobes are set on the same function.

For example:

```
# cd /sys/kernel/debug/tracing
```

```
# echo "r:event1 sys_symlink" > kprobe_events
```

```
# echo "r:event2 sys_symlink" >> kprobe_events
```

```
# echo 1 > events/kprobes/enable
```

```
# ln -s /tmp/foo /tmp/bar
```



(1/2)

Fix a bug on kprobe

(without this patch)

cat trace

```
In-897 [000] 20404.133727: event1: (kretprobe_trampoline+0x0/0x4c <- s
In-897 [000] 20404.133747: event2: (system_call_fastpath+0x16/0x1b <-
```

(with this patch)

cat trace

```
In-740 [000] 13799.491076: event1: (system_call_fastpath+0x16/0x1b <-
In-740 [000] 13799.491096: event2: (system_call_fastpath+0x16/0x1b <-
```

Signed-off-by: KUMANO Syuhei <kumano.prog@gmail.com>

Reviewed-by: Masami Hiramatsu <masami.hiramatsu.pt@hitachi.com>

Cc: Frederic Weisbecker <fweisbec@gmail.com>

Cc: Ananth N Mavinakayanahalli <ananth@in.ibm.com>

Cc: Peter Zijlstra <a.p.zijlstra@chello.nl>

Cc: YOSHIFUJI Hideaki <yoshfuji@linux-ipv6.org>

LKML-Reference: <1281853084.3254.11.camel@camp10-laptop>

Signed-off-by: Ingo Molnar <mingo@elte.hu>



(2/2)

Fix a bug on “make localmodconfig”

Date: Sun Aug 15 13:13:17 2010 +0900

kconfig: Fix variable name typo %prompts in streamline_config.pl

When I added "use strict;" to streamline_config.pl, I saw the following warnings:

- > Global symbol "%prompt" requires explicit package name at scripts/kconfig/streamline_config.pl line 183.
- > Global symbol "%prompt" requires explicit package name at scripts/kconfig/streamline_config.pl line 368.

The declaration of %prompt was incorrect, and should have been %prompts.

Cc: Toralf Foerster <toralf.foerster@gmx.de>

Cc: KAMEZAWA Hiroyuki <kamezawa.hiroyu@jp.fujitsu.com>

Cc: YOSHIFUJI Hideaki <yoshfuji@linux-ipv6.org>

Signed-off-by: Hiromu Yakura <hiromu1996@gmail.com>

LKML-Reference: <1281845597.11566.5.camel@camp10-laptop>

Signed-off-by: Steven Rostedt <rostedt@goodmis.org>

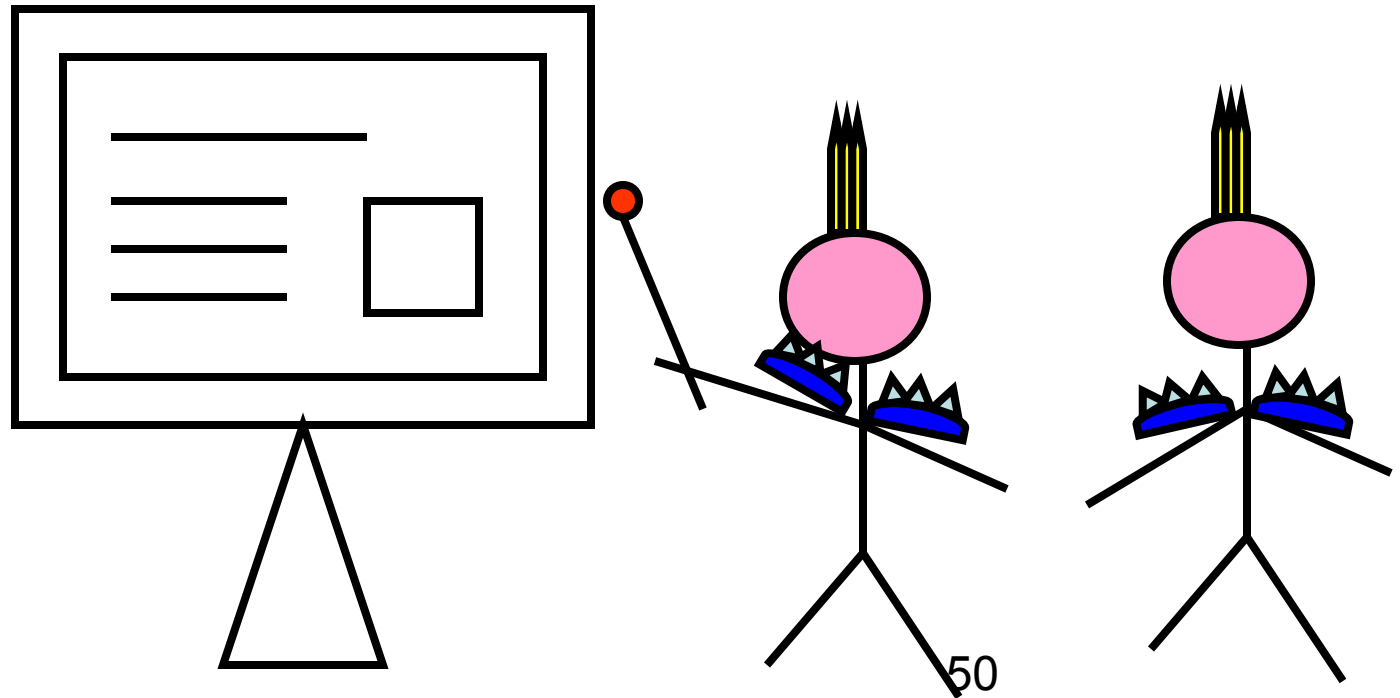


Contents

- Background
- What is Security and Programming Camp?
- **What linux kernel course did?**
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - **Day5: Progress report**
- The result
- Conclusion

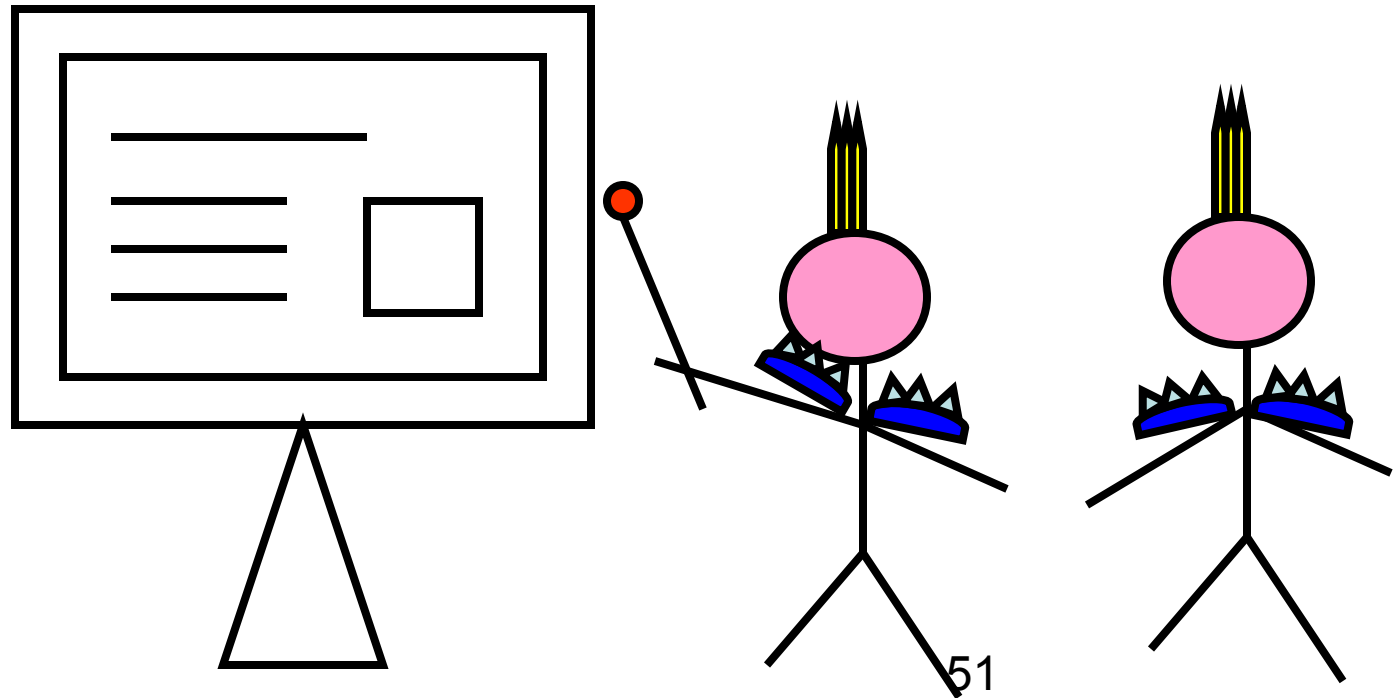
Day5: Progress report

- All trainees survived SPCamp.
- They explained what they did and how it was to all participants.



Day5: Progress report

- I introduce two of their presentation.
 1. Generate the sound on printk
 2. Visualize panic message



1. Generate the sound on printk

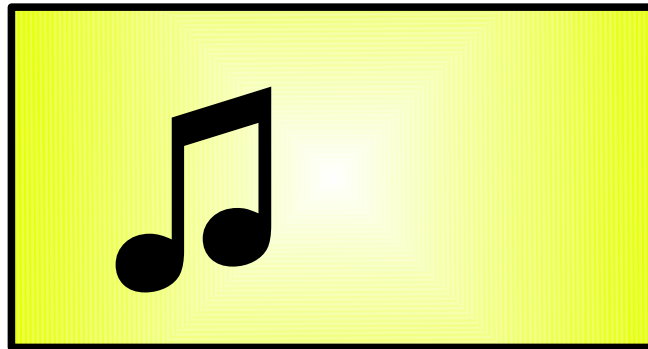
- Developers/Admins sometimes need to get the information from dmesg
- But they are always busy.
- They don't want to stop their work.



Hiroki MIYAOKA

1. Generate the sound on printk

- Hiroki's solution: How about converting printk message to the sound?



On inserting a USB memory...

2. Visualize panic message

- The kernel panic message is complicated and difficult to understand for admin.

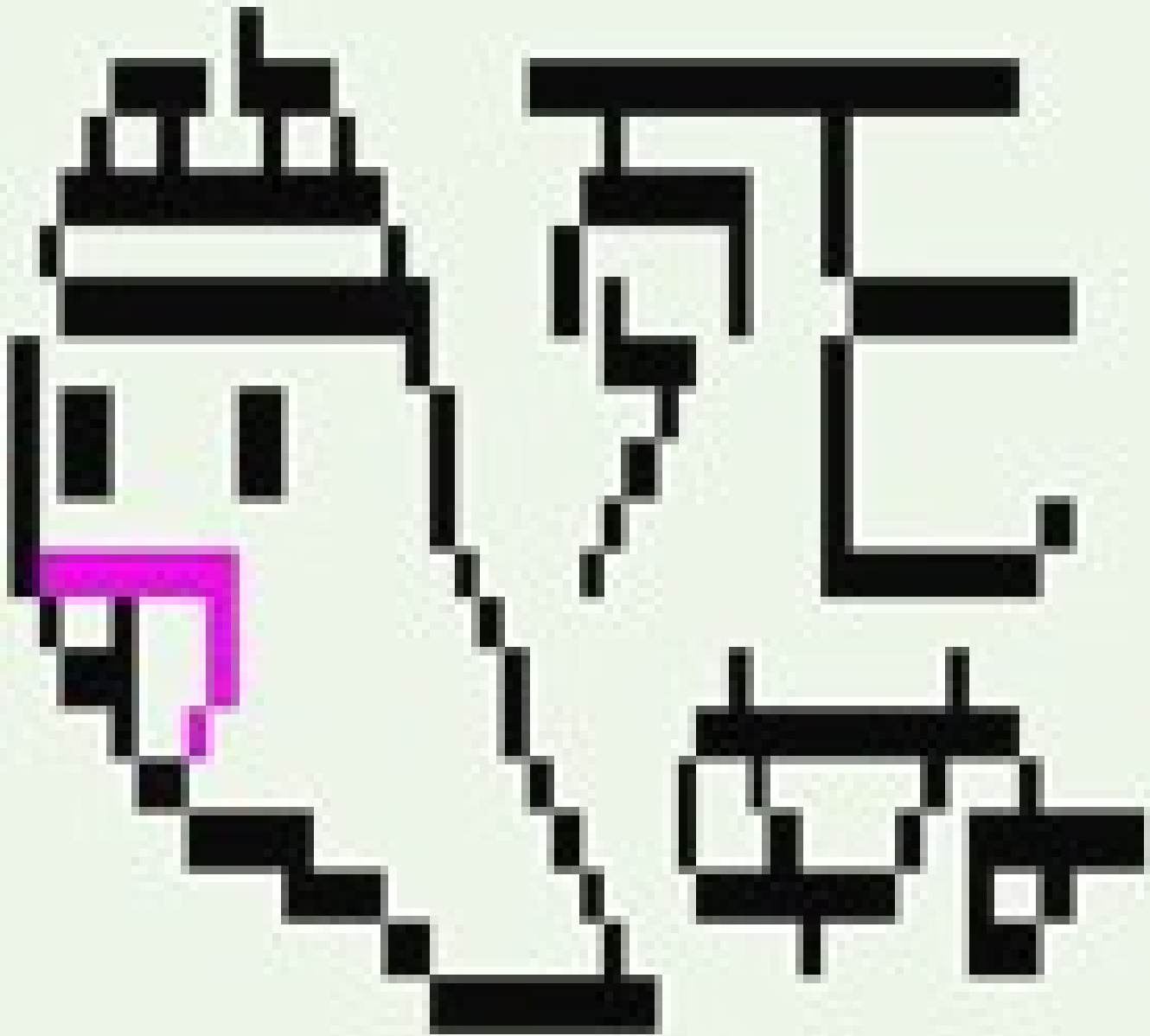
```
Warning: kree_skb on hard IRQ XXXXXXXXX  
scheduling while atomic: foobar/0xXXXXXXXXXX/8215  
[<XXXXXXXXXX>] schedule+YY/ZZ  
[<XXXXXXXXXX>] sys_sched_yield+YY/ZZ  
[<XXXXXXXXXX>] coredump_wait+YY/ZZ  
[<XXXXXXXXXX>] do_coredump+YY/ZZ  
[<XXXXXXXXXX>] d_alloc+YY/ZZ.....,
```

2. Visualize panic message

- Kunihiro's solution: How about visualize the panic message?

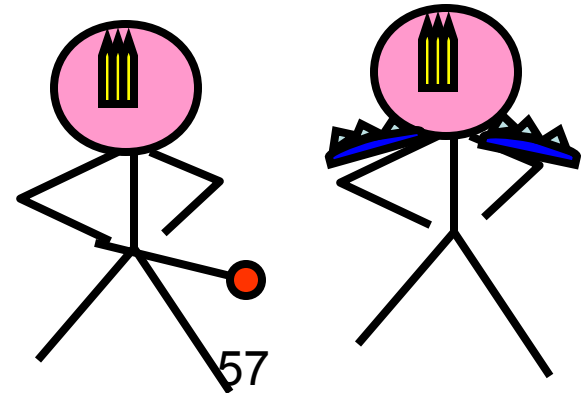
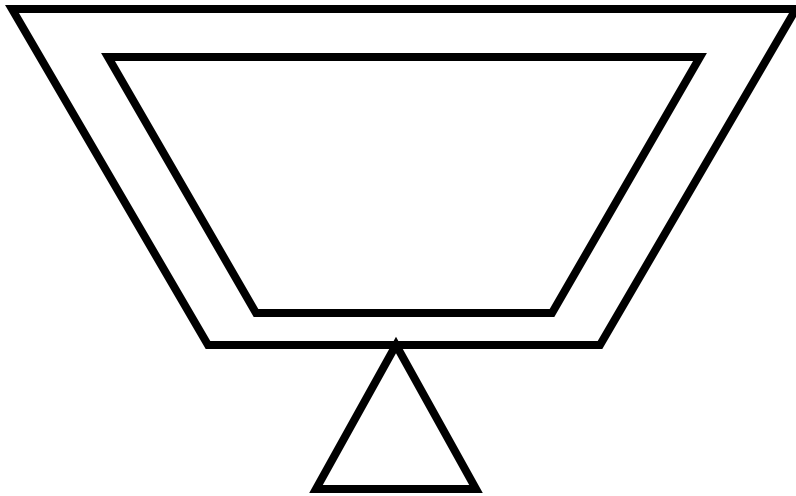


Kunihiro UEDA



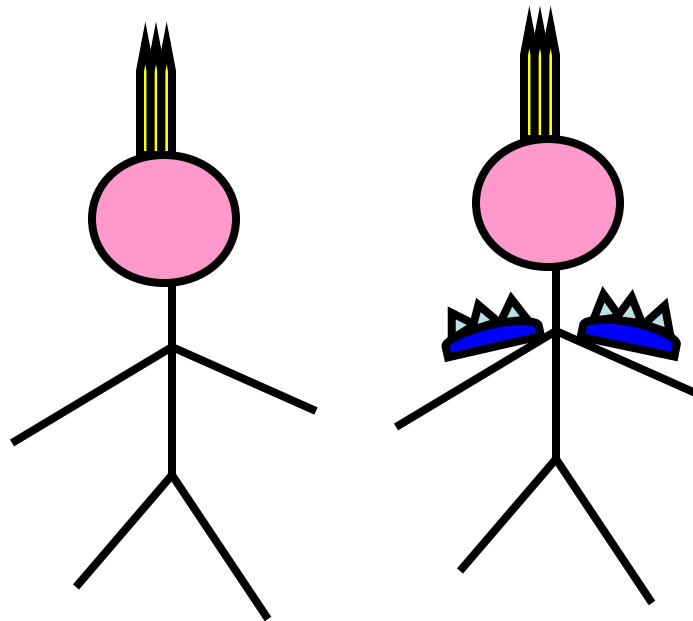
Day5: Progress report

- After their presentation, all schedules were finished.



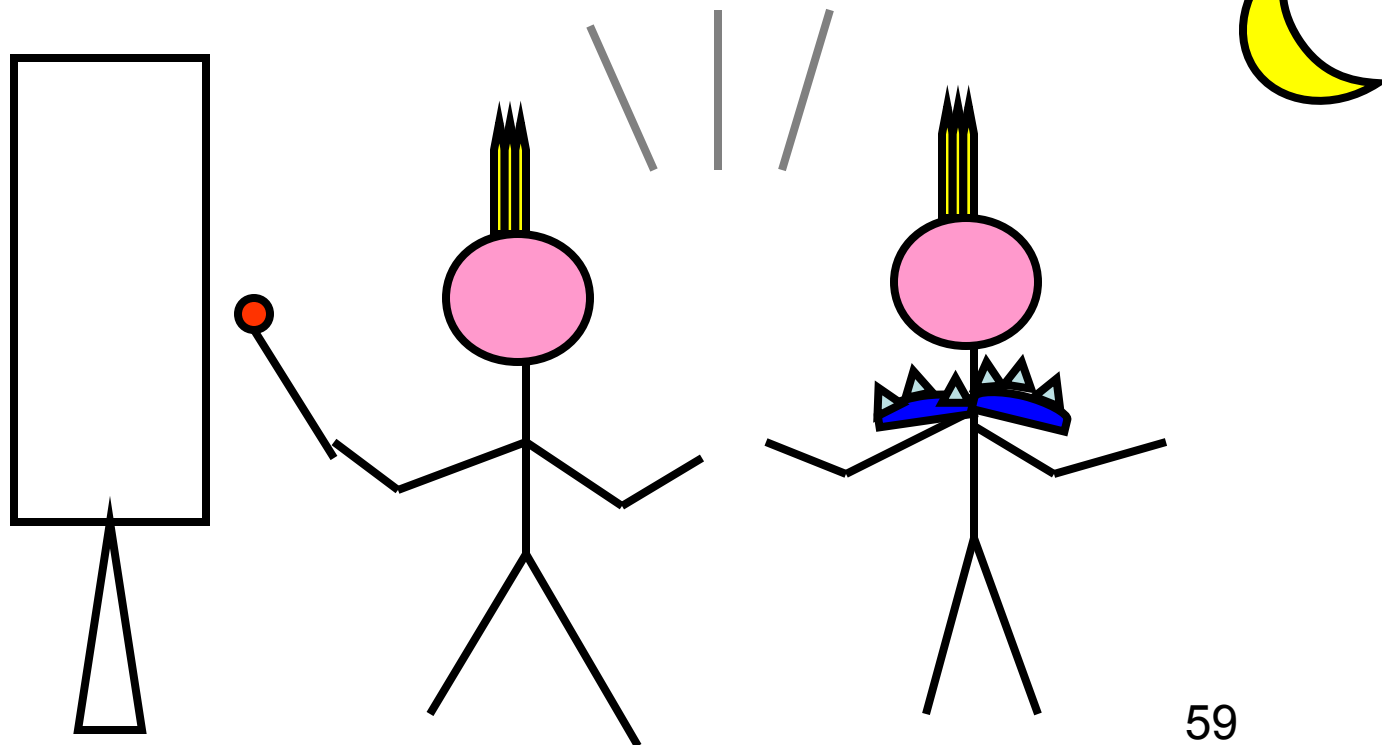
An episode

- Linux course was very hard. There is an episode which express the situation at that time very well.



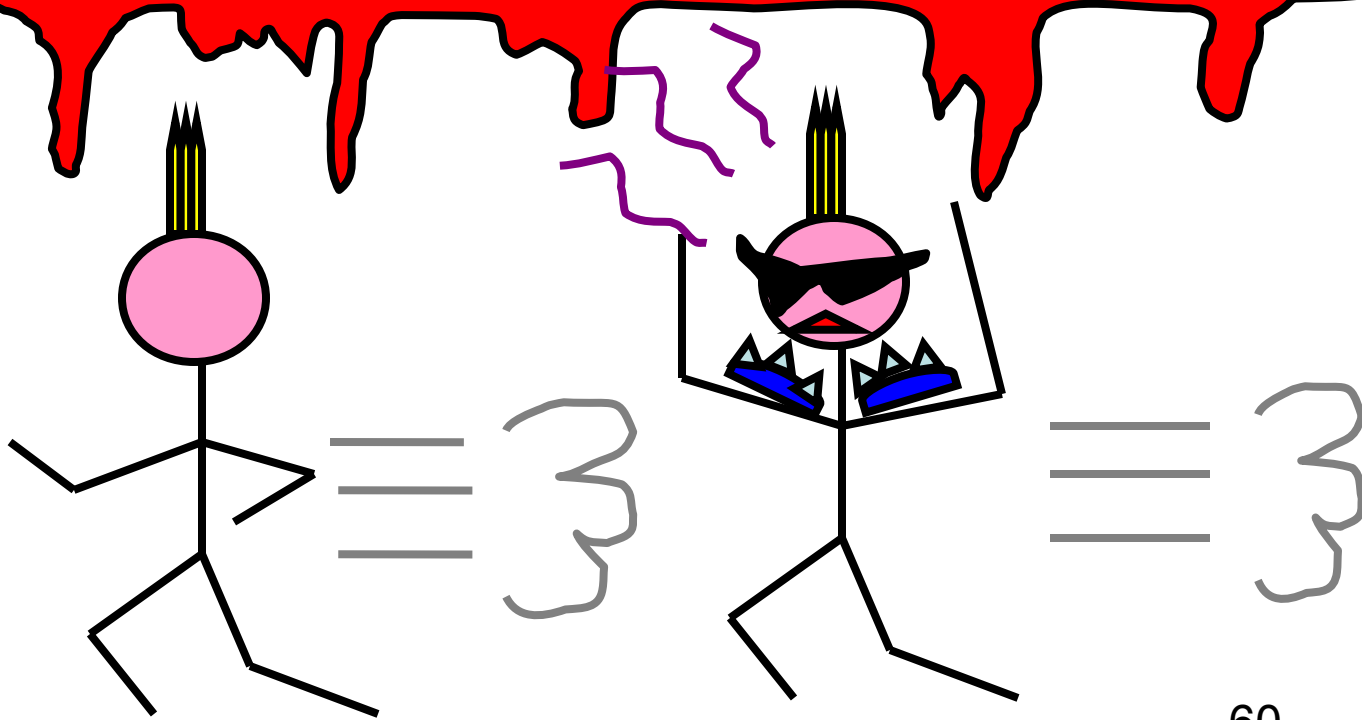
An episode

- The midnight of Day4, the trainers were reviewing the materials of trainees for the progress report in the next day.



An episode

Our materials were critisized and rejected again and again... What crazy guys!



Contents

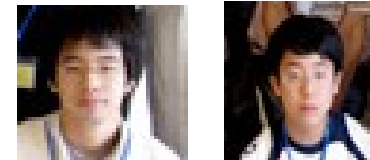
- Background
- What is Security and Programming Camp?
- What we did?
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- **The result**
- Conclusion

The Result

1. Technical growing up
2. Others

1. Technical growing up

- All trainees got kernel hacking skills.
 - How to use many tools: diff, patch, kbuild, source code tagging systems, git, and debugging tools
 - How to get/modify linux kernel and send his own patch to LKML.
- Two of trainees succeeded to apply their patch to upstream kernel during SPCamp.



1. Technical growing up

- Two of trainees voluntary sent their patches to LKML and four patches have already been applied.
 - Hiromu YAKURA: kconfig fix.
 - Naohiro AOTA: kconfig fix.
 - Naohiro AOTA: kernel-doc warning fix.



1. Technical growing up

- Yuya TANAKA sent his patch to procps community.
- It can speed up ps command 10 times in some situation.



Yuya TANAKA

1. Technical growing up

- Yuya TANAKA send his patch to procps community.
- It can speed up ps command 10 times faster in some situation.
- **No one replied to him yet :-)**



Yuya TANAKA

1. Technical growing up

- In addition, there is also a continuing discussion.
 - Hiromu YAKURA: Fixing kconfig

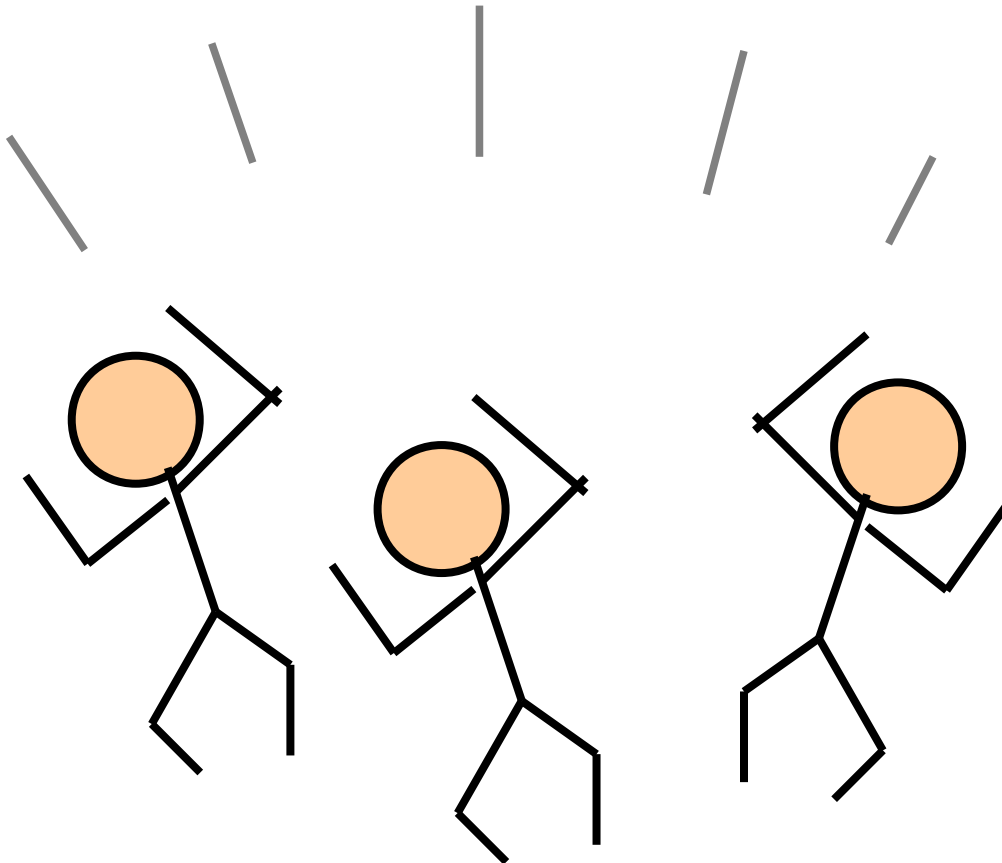


2. Others

- The trainees becomes frequent participant of many events about software development, for example LinuxCon Japan.
- They found many friends and the networks among all participants are formed.

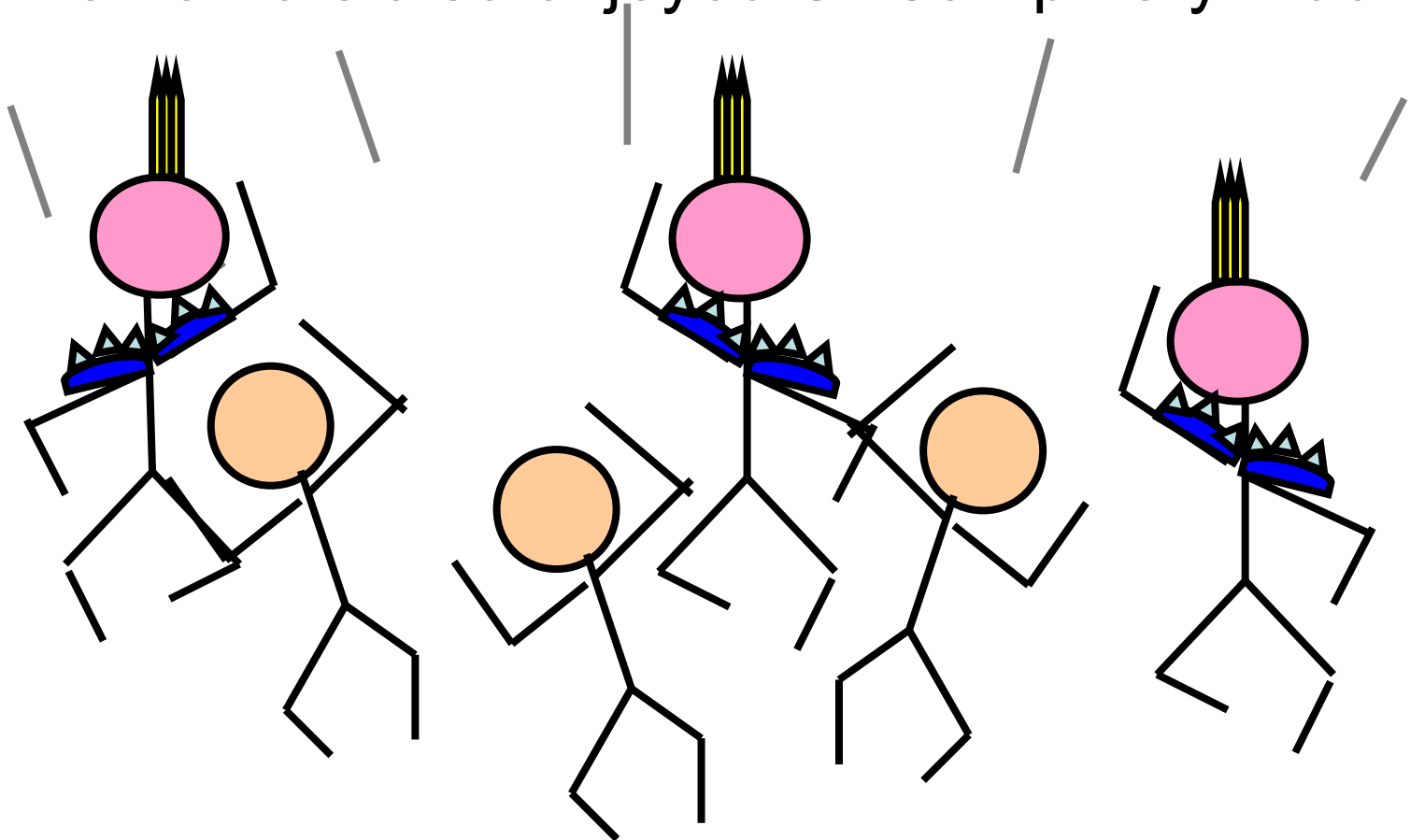
2. After Others

- The trainees enjoyed SPCamp very much!



2. Others

- The trainers also enjoyed SPCamp very much!



Contents

- Background
- What is Security and Programming Camp?
- What we did?
 - One month ago-Day0: Limbering up
 - Day1-Day3: Strength training
 - Day4: Actual combat. Hack! Hack! Hack!
 - Day5: Progress report
- The result
- **Conclusion**

Conclusion

- SPCamp has succeeded to promote young students for many years.
- This way is proved to be efficient for promoting new linux kernel hackers.
- I hope this way spreads from Japan to entire linux kernel community.

That's all, Thank you very much!



Any question?

