2013, Калуга

LinuxTesting.org

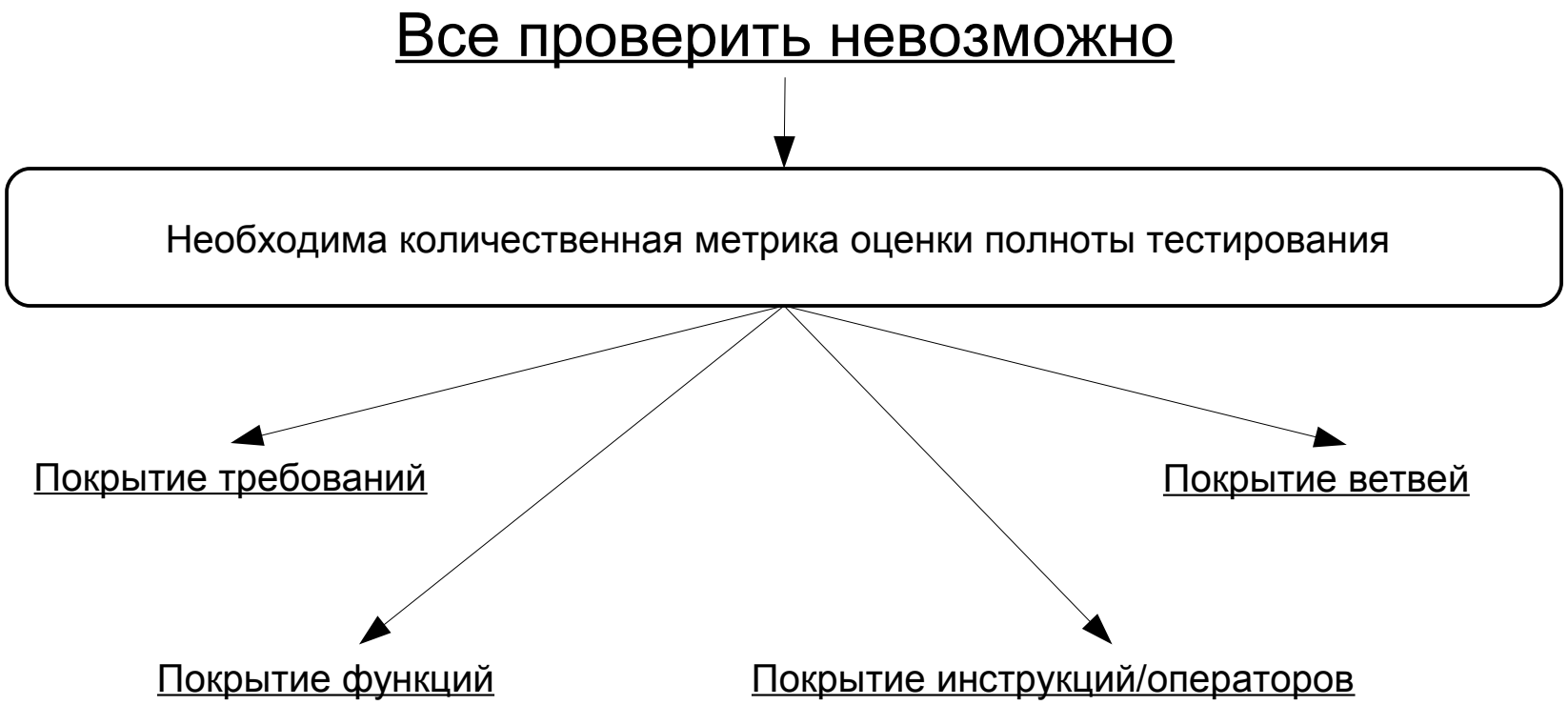
Оценка покрытия кода при статическом анализе

Андрианов Павел
andrianov@ispras.ru

http://linuxtesting.org/project/ldv

ISP RAS
Institute for System Programming of the Russian Academy of Sciences

Тестирование

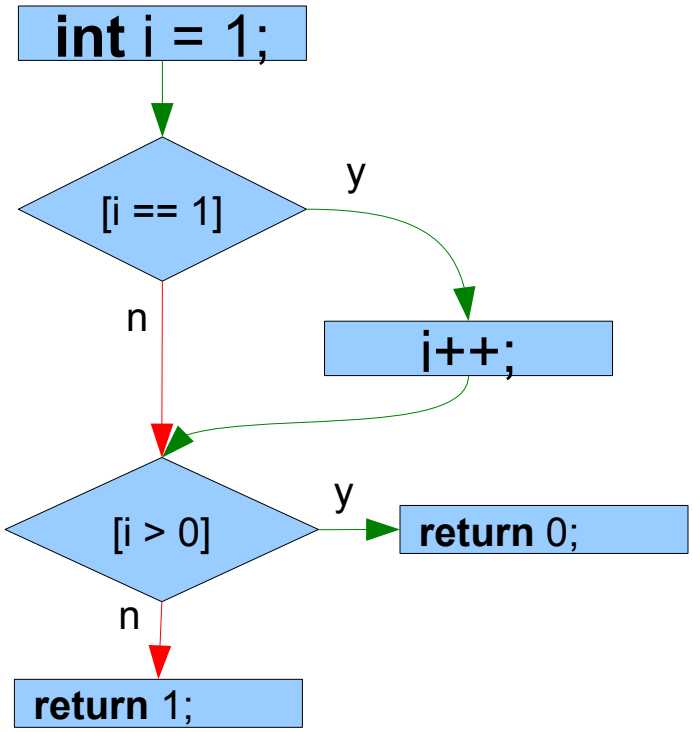


Покрытие инструкций/операторов

```
int func() {
    int i = 1;
    if (i == 1){
        i++;
    }
    if (i > 0) {
        return 0;
    } else {
        return 1;
    }
}
```

Выполнены 5 из 6 инструкций, покрытие 83%.

Покрытие ветвей



Выполнено 2 из 4 ветвей,
покрытие 50% по ветвям.

Динамический анализ

Плюсы:

- Ложные срабатывания практически исключены
- Не требуется исходный код

Минусы:

- Зависимость от входных данных
- Возможность выполнения некорректного кода

Статический анализ

Плюсы:

- Анализ всех путей исполнения программы
- Масштабируемость

Минусы:

- Большое число ложных срабатываний

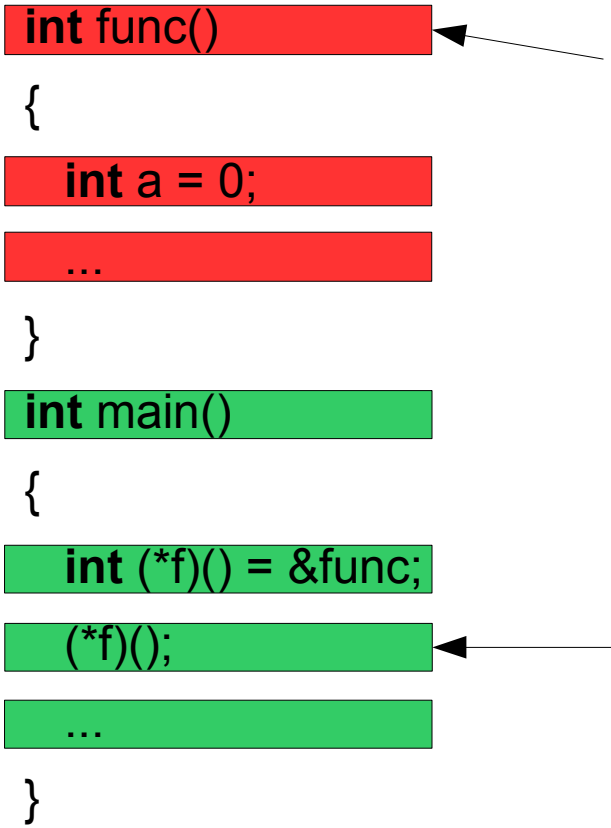
Пример ошибки

```
int func(int a) {
    int *i;
    i = malloc(sizeof(int));
    *i = 1;
    ...
}
```

- Такую ошибку сложно поймать динамическим анализом

Возможно разыменование NULL-указателя

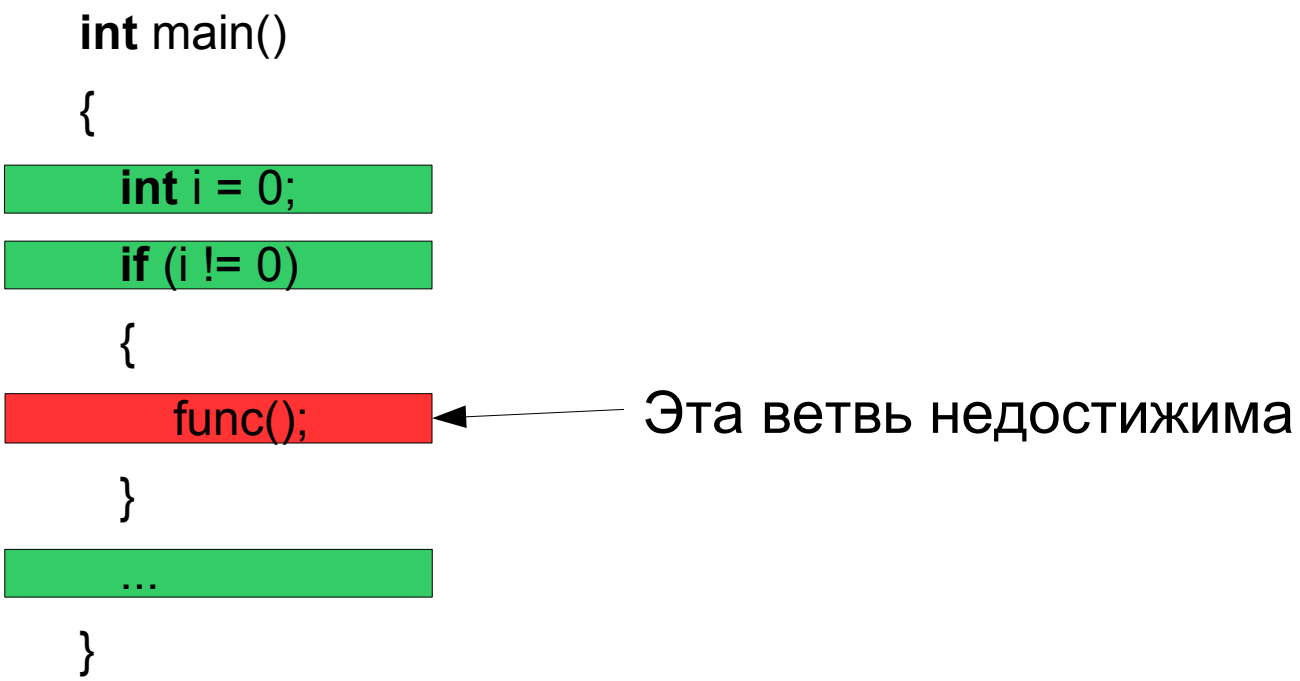
Зачем нужно покрытие?



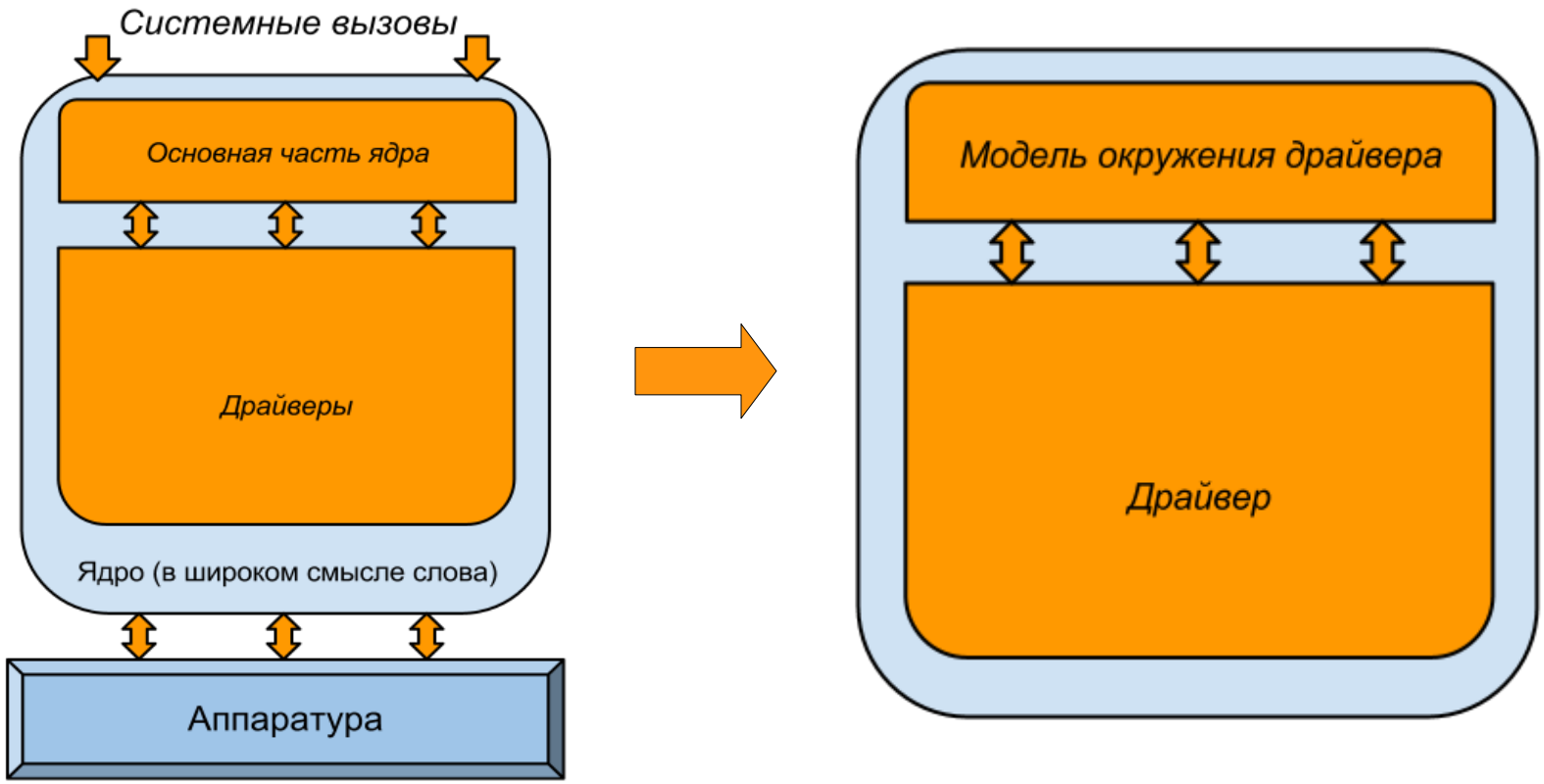
Функция func не анализировалась инструментом

На самом деле функция вызывается

Зачем нужно покрытие?



Анализ драйверов



Модель окружения

```
int ldv_probe(struct usb_interface *intf, const
struct usb_device_id *id){
      ...
}
static void ldv_disconnect(struct usb_interface
*intf){
      ...
}
static struct usb_driver ldv_driver = {
      .name =           "ldv-test",
      .probe =      ldv_probe,
      .disconnect =      ldv_disconnect,
};
int __init ldv_init(void){
      return usb_register(&ldv_driver);
}
void __exit ldv_exit(void){
      usb_deregister(&ldv_driver);
}
module_init(ldv_init);
module_exit(ldv_exit);
```

```
void entry_point(void) {
  if(ldv_init()) goto ldv_final;
   while(nondet_int()){
     switch(nondet_int()) {
     case 0:
       if(...) {
          res = ldv_probe(...);
          if(res)  goto ldv_exit;
        }
       break;
     case 1:
       if(...){
          ldv_disconnect(...);
          goto ldv_exit;
        }
       break;
     default: break;
     }
   }
  ldv_exit: ldv_exit();
  ldv_final: return;
}
```

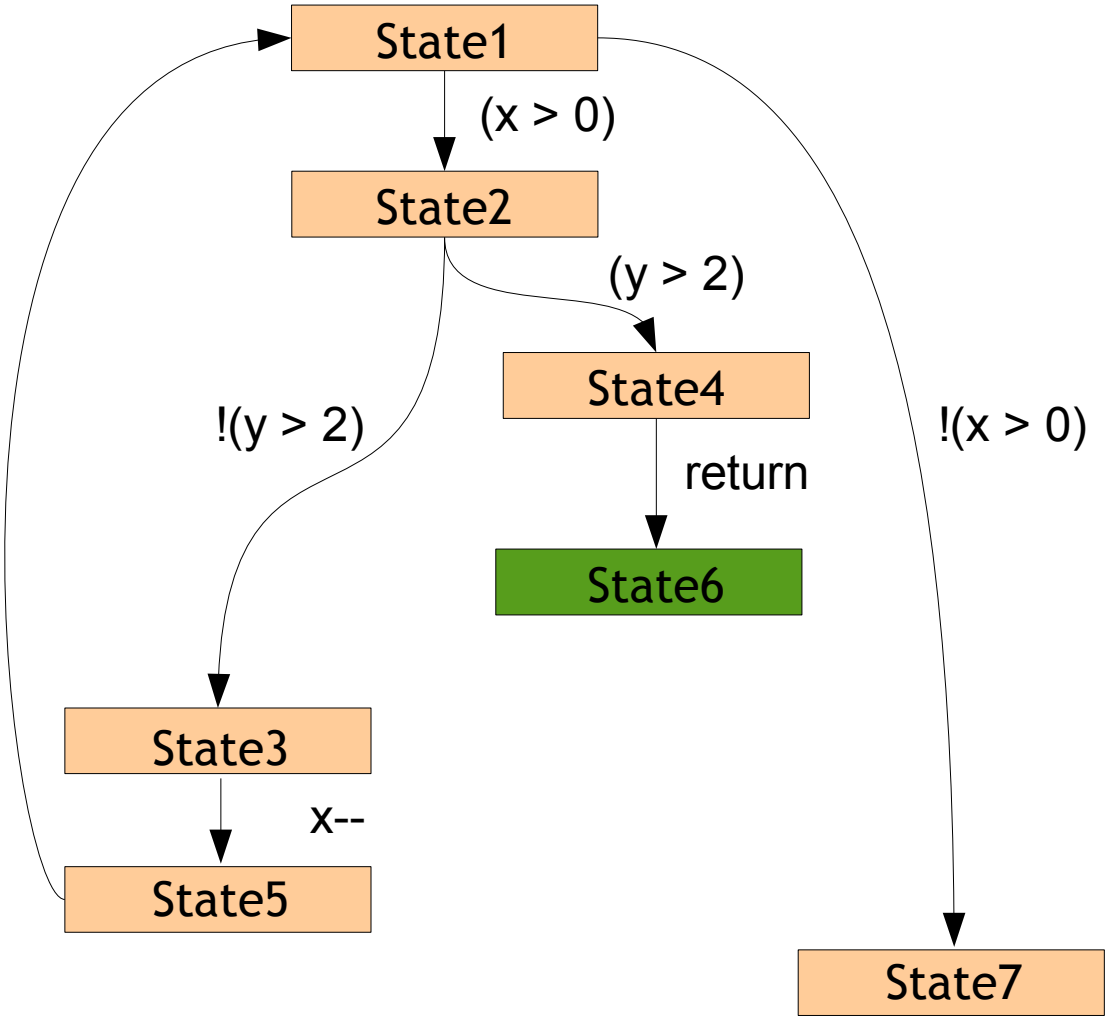
Возможные проблемы с моделью окружения

- Отсутствие вызова функции
- Передача некорректных аргументов
- Некорректная интерпретация возвращаемых значений
- Отсутствие реализации и модели для функции

Анализ покрытия может помочь выявить такие случаи

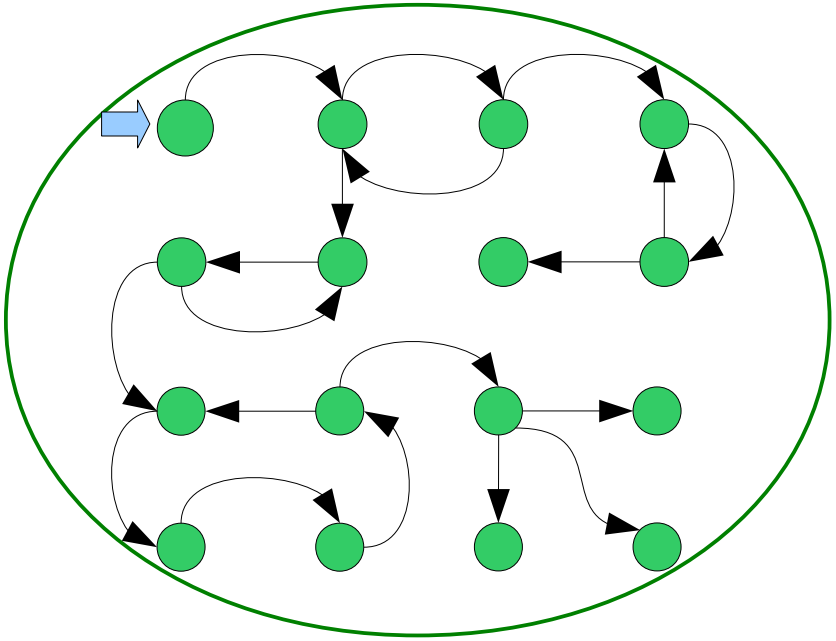
Общая идея статического анализа

```
while (x > 0) {

    if (y > 2) {

        return;

    }

    x--;
}
```

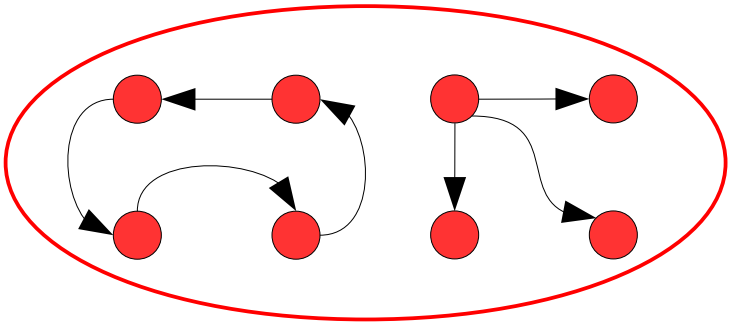


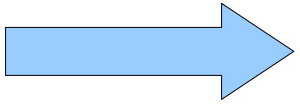
Предлагаемый подход

Демонстрация идеи



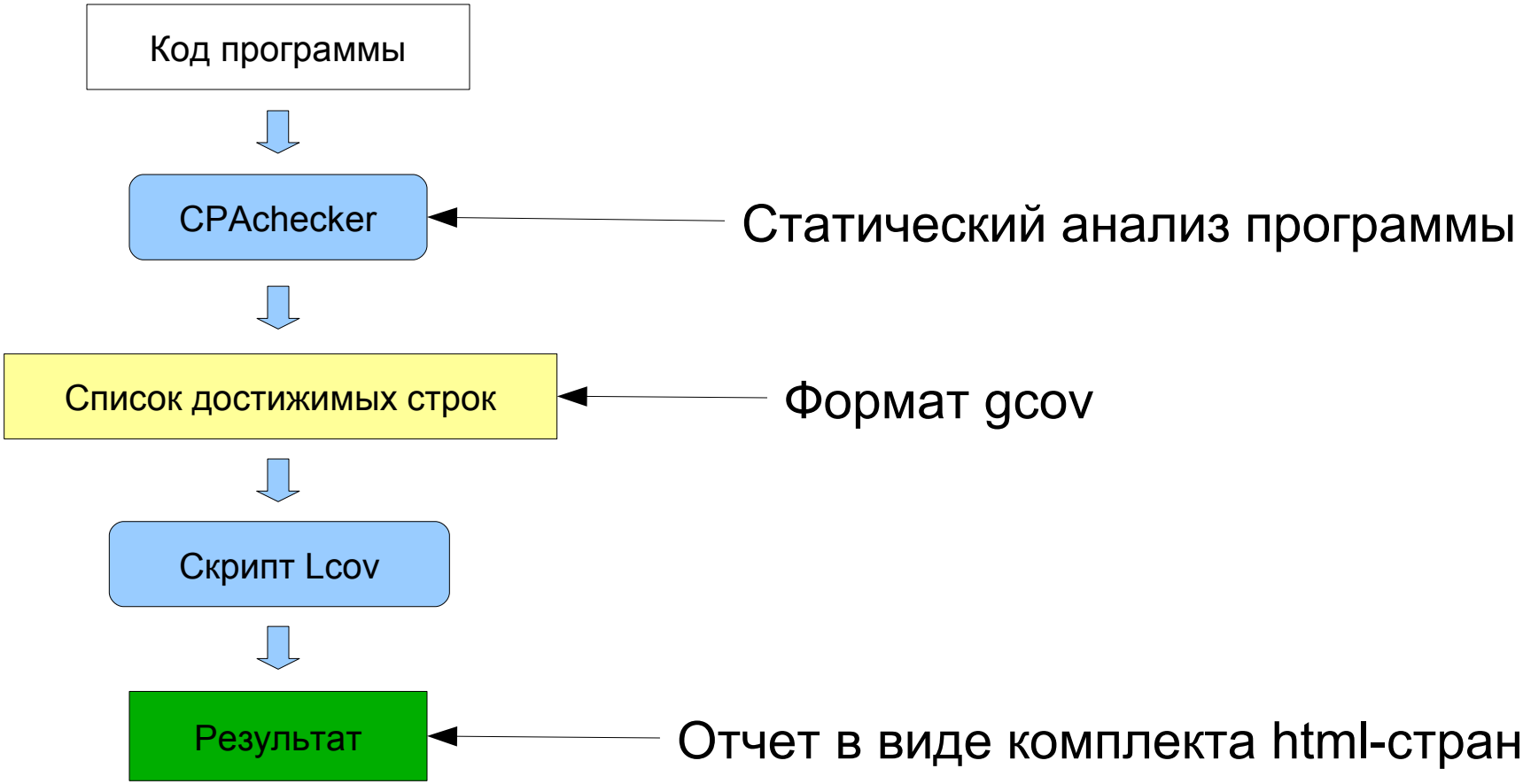
Множество покрытых строк





Множество непокрытых строк

Реализация


Код программы
CPAchecker
Статический анализ программы
Список достижимых строк
Формат gcov
Скрипт Lcov
Результат
Отчет в виде комплекта html-страниц

Пример отчета

LCOV - code coverage report

				Hit	Total	Coverage
Current view:	top level - cil-files - gcov.c (source / functions)		Lines:	7	10	70.0 %
Test:	coverage.info		Functions:	1	2	50.0 %
Date:	2013-09-19					

```
    Line data    Source code
 1          0 : int func() {
 2          0 :         int a = 0;
 3          0 :         return a;
 4            : }
 5            :
 6          1 : int main()
 7            : {
 8            :   int k;
 9          1 :   int* j = malloc(sizeof(int));
10          1 :   int (*f)() = &func;
11            :
12          1 :   if (j != 0)
13            :   {
14          1 :           *j = (*f)();
15            :   }
16            :   else {
17          1 :           return -1;
18          1 :   }
19            : }
20            :
```

LinuxTesting.org

top level
coverage.orig
2013-08-10
Rating: low: < 75 % | medium: >= 75 % | high: >= 90 %

	Hit	Total	Coverage
Lines:	1163351	2173410	53.5 %
Functions:	67390	165218	40.8 %
Branches:	0	0	-

Directory	Line Coverage		Functions		Branches	
/home/ldvuser/ldv_envgen2/kernel-rules/verifier	85.7 %	6 / 7	100.0 %	3 / 3	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/arch/x86/include/asm	77.5 %	458 / 591	71.4 %	155 / 217	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/arch/x86/include/asm/uv	66.1 %	37 / 56	76.2 %	16 / 21	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/arch/x86/include/asm/xen	65.5 %	38 / 58	60.0 %	6 / 10	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/arch/x86/include/uapi/asm	100.0 %	6 / 6	100.0 %	2 / 2	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/acpi/apei	30.8 %	4 / 13	20.0 %	1 / 5	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/ata	100.0 %	64 / 64	100.0 %	7 / 7	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/atm	100.0 %	43 / 43	100.0 %	3 / 3	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/block	90.9 %	1128 / 1241	91.9 %	136 / 148	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/block/drbd	1.5 %	6 / 402	3.3 %	3 / 91	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/block/paride	65.8 %	25 / 38	75.0 %	3 / 4	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/block/xen-blkback	0.0 %	0 / 42	0.0 %	0 / 2	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/char/tpm	100.0 %	23 / 23	100.0 %	5 / 5	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/cpufreq	100.0 %	5 / 5	100.0 %	1 / 1	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/dma	11.1 %	4 / 36	9.1 %	1 / 11	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/dma/ioat	21.5 %	17 / 79	26.9 %	7 / 26	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/edac	54.1 %	20 / 37	50.0 %	4 / 8	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/firewire	70.6 %	12 / 17	66.7 %	4 / 6	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/gpu/drm	0.0 %	0 / 9	0.0 %	0 / 3	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/gpu/drm/ast	100.0 %	15 / 15	100.0 %	8 / 8	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/gpu/drm/cirrus	100.0 %	2 / 2	100.0 %	1 / 1	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/gpu/drm/i2c	100.0 %	4 / 4	100.0 %	2 / 2	-	0 / 0
/work/ldvuser/zakharov_benchmarks/bench/cpa/inst/current/envs/linux-3.8-rc1/linux-3.8-rc1/drivers/gpu/drm/i915	0.0 %	0 / 155	0.0 %	0 / 55	-	0 / 0

Пример ложного срабатывания

```
ldv_initialize()
usbpn_xmit(var_group2, var_group1)
{
    req = usb_alloc_urb(0, 32)
    usb_fill_bulk_urb(req, *(pnd).usb, *(pnd).tx_pipe, *(skb).data, *(skb).len,
&(tx_complete), skb);
    err = usb_submit_urb(req, 32);
    tmp___9 = spinlock_check(&(pnd)->tx_lock);
    flags = _raw_spin_lock_irqsave(tmp___9);
    *(pnd).tx_queue = *(pnd).tx_queue + 1;
    spin_unlock_irqrestore(&(pnd)->tx_lock, flags);
}
ldv_check_final_state()
```

Результаты

- Реализована возможность сбора покрытия во время статического анализа на основе инструмента CPAchecker.
- Инструмент активно используется в проекте LDV, что позволило с помощью анализа покрытия обнаружить неточности в некоторых видах анализа и в модели окружения.

LinuxTesting.org

Спасибо за внимание!

Андрианов Павел
andrianov@ispras.ru

http://linuxtesting.org/project/ldv

ISP RAS
Institute for System Programming of the Russian Academy of Sciences