# *Штатные механизмы QNX Neutrino для обеспечения отказоустойчивости АС*

Зыль С.Н.

ООО «СВД Встраиваемые Системы»

(www.kpda.ru)

# Что такое «отказоустойчивость»?

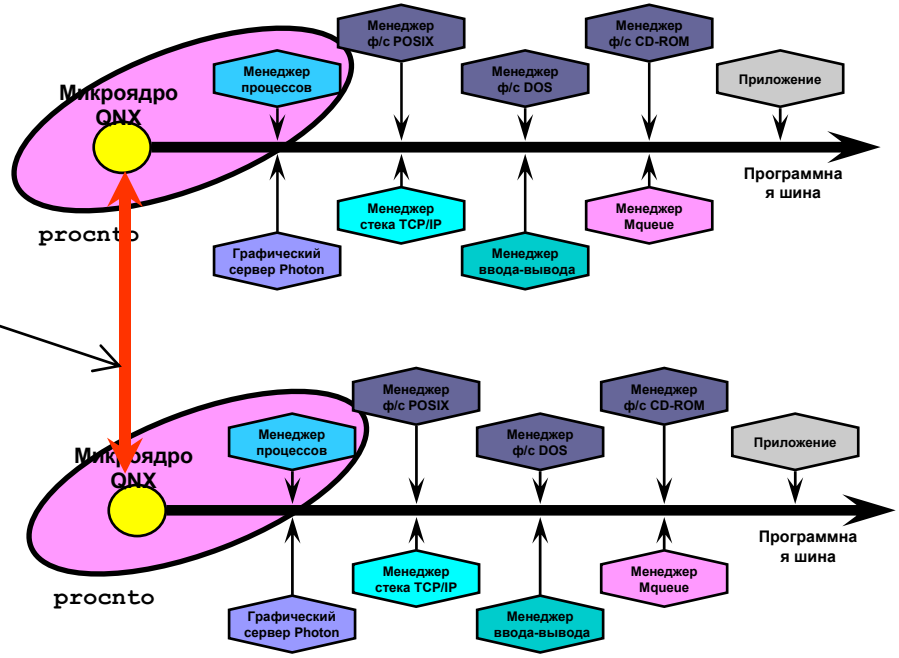
- Отказ (Failure)
  - IEC 61508 – «полная потеря функциональной единицей способности выполнять требуемую функцию»
- Неисправность (Fault)
  - IEC 61508 – «аномальное состояние, способное вызвать снижение или потерю функциональной единицей способности выполнять требуемую функцию»
  - IEV 191-05-01 - «состояние, характеризуемое неспособностью выполнять требуемую функцию, если таковая не вызвана проведением профилактических мероприятий технического обслуживания или других плановых мероприятий, а также отсутствием внешних ресурсов»
- Ошибка (Error)
  - IEC 61508 – «расхождение между рассчитанной на ЭВМ, наблюдаемой или измеренной величиной или состоянием и правильной, установленной или теоретически верной величиной или состоянием»
- Fault tolerance
  - IEC 61508 – «способность функциональной единицы продолжать выполнять требуемую функцию при наличии неисправностей и ошибок»

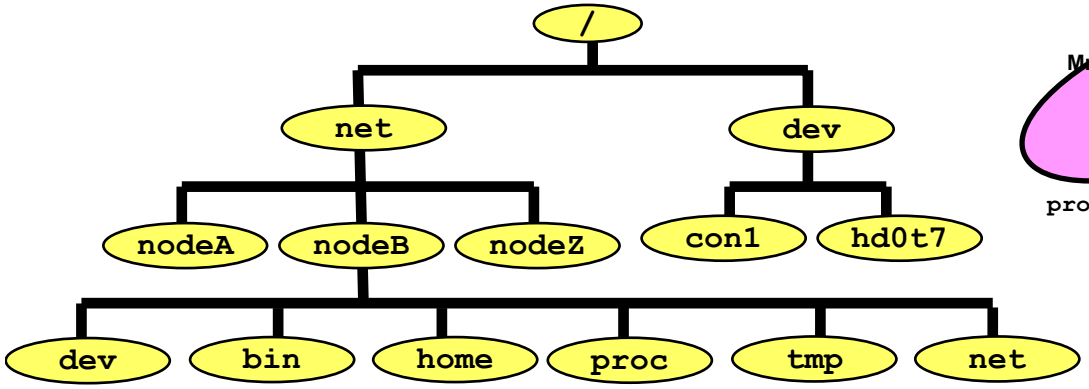
# Что значит «обеспечить отказоустойчивость»?

- Локализация сбоя – сократить «масштабы бедствия»
- Идентификация сбоя – выявить «зависание» или «крах» программного модуля так, что бы можно было проанализировать проблему и/или быстро восстановить систему
- Восстановление после сбоя – восстановить способность выполнять заданные функции

# Локализация – распределённая вычислит. среда

Технология QNX TDN на базе протокола Qnet (4 уровень ISO OSI)



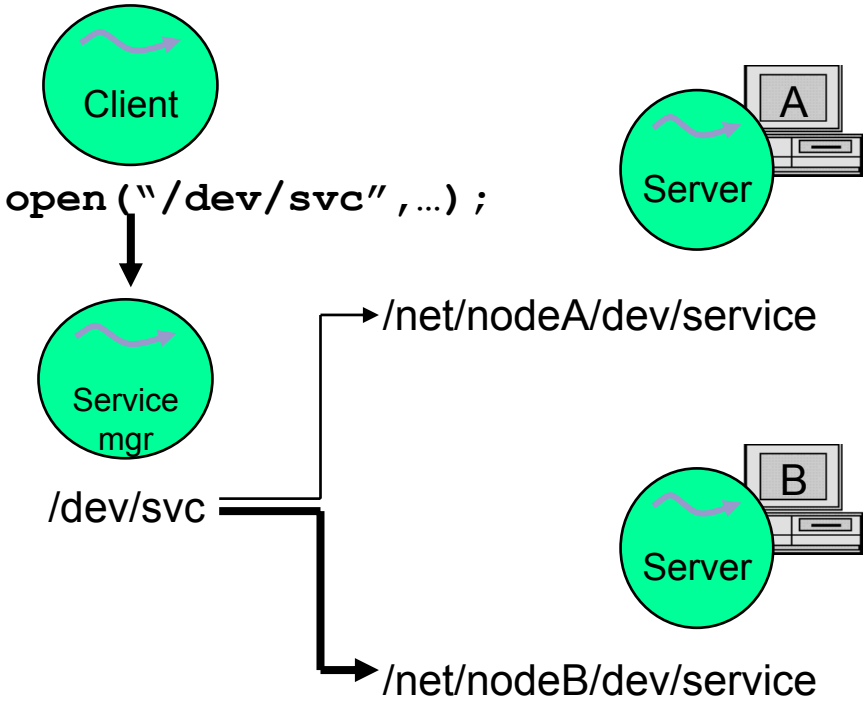




Менеджер Qnet регистрирует префикс /net на своём узле

## Запуск программы в распределенном режиме:

```
on -n nodeB \
      -f nodeC \
-t /net/nodeD~preferred:en1/dev/tty0 \
      /net/nodeE~exclusive:en2/bin/ls
```

# Локализация – распределение нагрузки между серверами



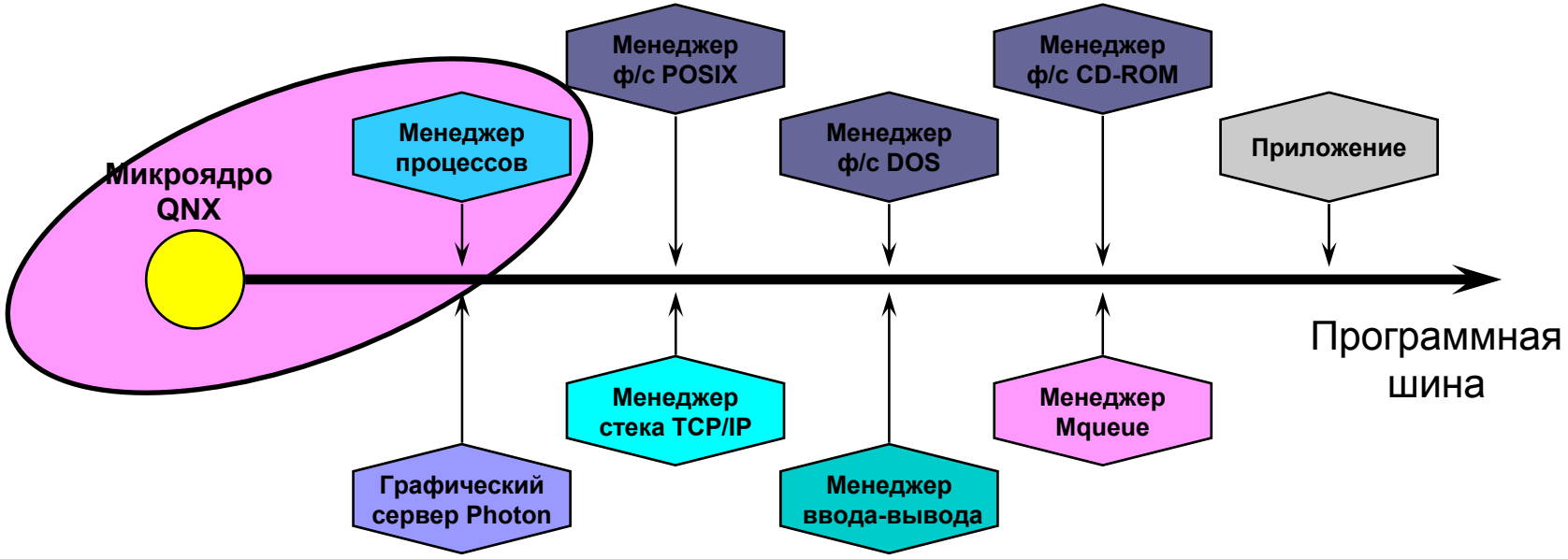

Код перенаправления клиентского запроса:

```
int
io_open (resmgr_context_t *ctp, io_open_t *msg,
         RESMGR_HANDLE_T *handle, void *extra)
{
   eflag = msg->connect.eflag;
   ftype = msg->connect.file_type;
   memset(&msg->link_reply, 0, sizeof(msg->link_reply));
   _IO_SET_CONNECT_RET(ctp, _IO_CONNECT_RET_LINK);
   msg->link_reply.file_type = ftype;
   msg->link_reply.eflag = eflag;
   msg->link_reply.nentries = 0;
   msg->link_reply.path_len = strlen(new_path) + 1;
   strcpy(((char *)msg + sizeof(msg->link_reply)), new_path);
   retlen  = sizeof(msg->link_reply) +
     msg->link_reply.path_len;
   return _RESMGR_PTR(ctp, msg, retlen);
}
```

Новый запрос **`open("/dev/svc", ...)`** поступит на один из узлов

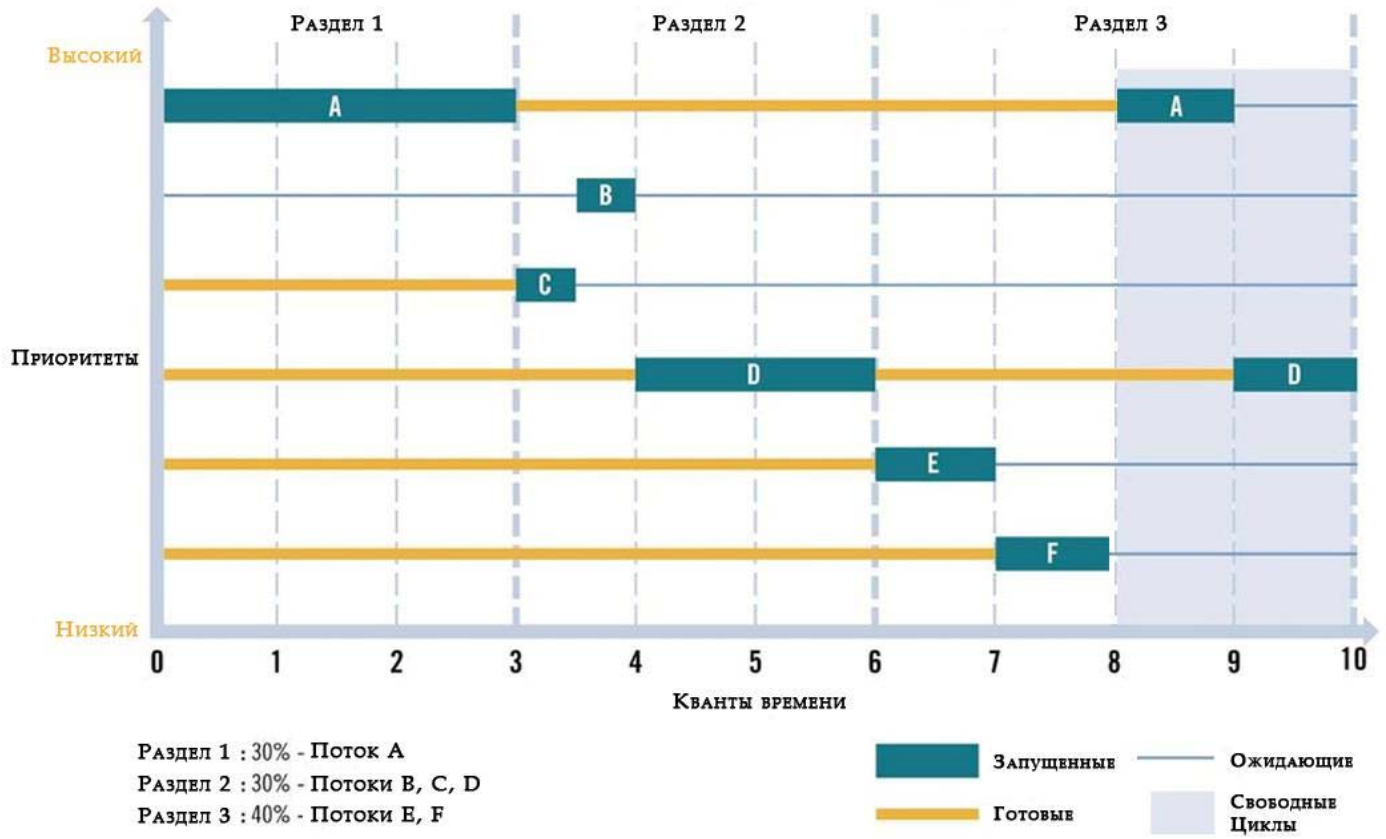
# Локализация - микроядерная архитектура

**`procnto`** – коммутатор процессов



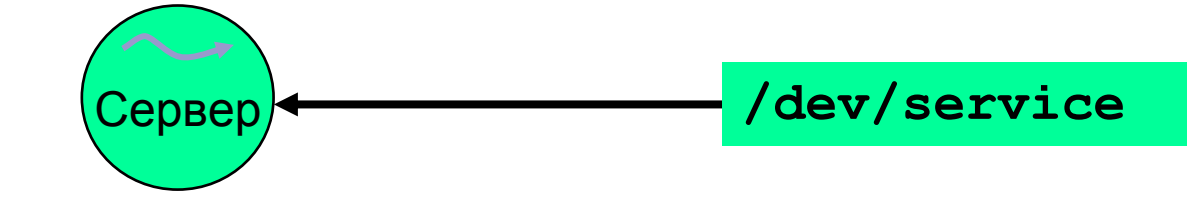

1) «Proven-in-Use» ядро QNX не требует модификации
2) Изоляция сбоев (в том числе системных сервисов)
3) Возможность перезапуска отказавшего сервиса
4) Маршрутизация клиентских запросов к сервисам

# Локализация – адаптивное квотирование ресурсов

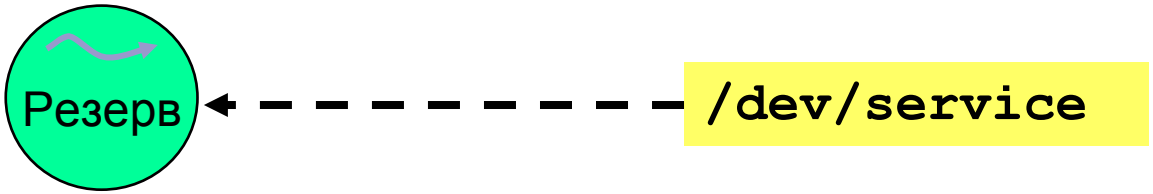



✓*Повышение защищённости и Кг АС за счёт невозможности монополизации ресурсов какой-либо программой (при DoS-атаках и некорректном коде);*
✓*Сокращение трудозатрат на сопровождение АС на 25-30% (по оценкам QSS) за счёт локализации аномального поведения.*

# Идентификация – редирект клиентских запросов




```
resmgr_attach(…,"/dev/service",…);
```



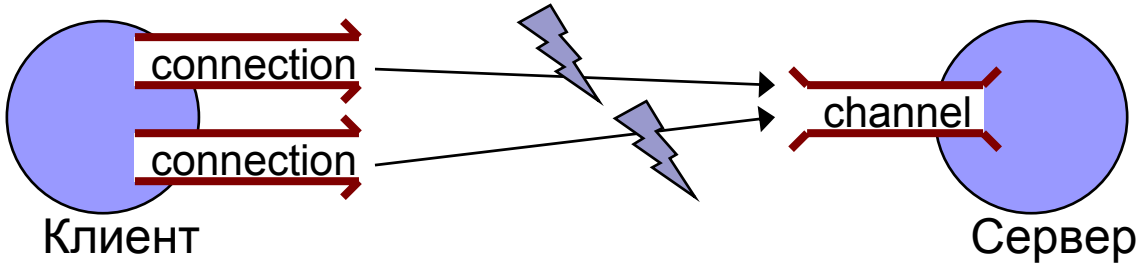

```
resmgr_attach(…,"/dev/service",_RESMGR_FLAG_AFTER,…);
```

При сбое основного сервера ядро автоматически направляет запросы клиентов на резервный сервер

# Идентификация – извещение о завершении клиента

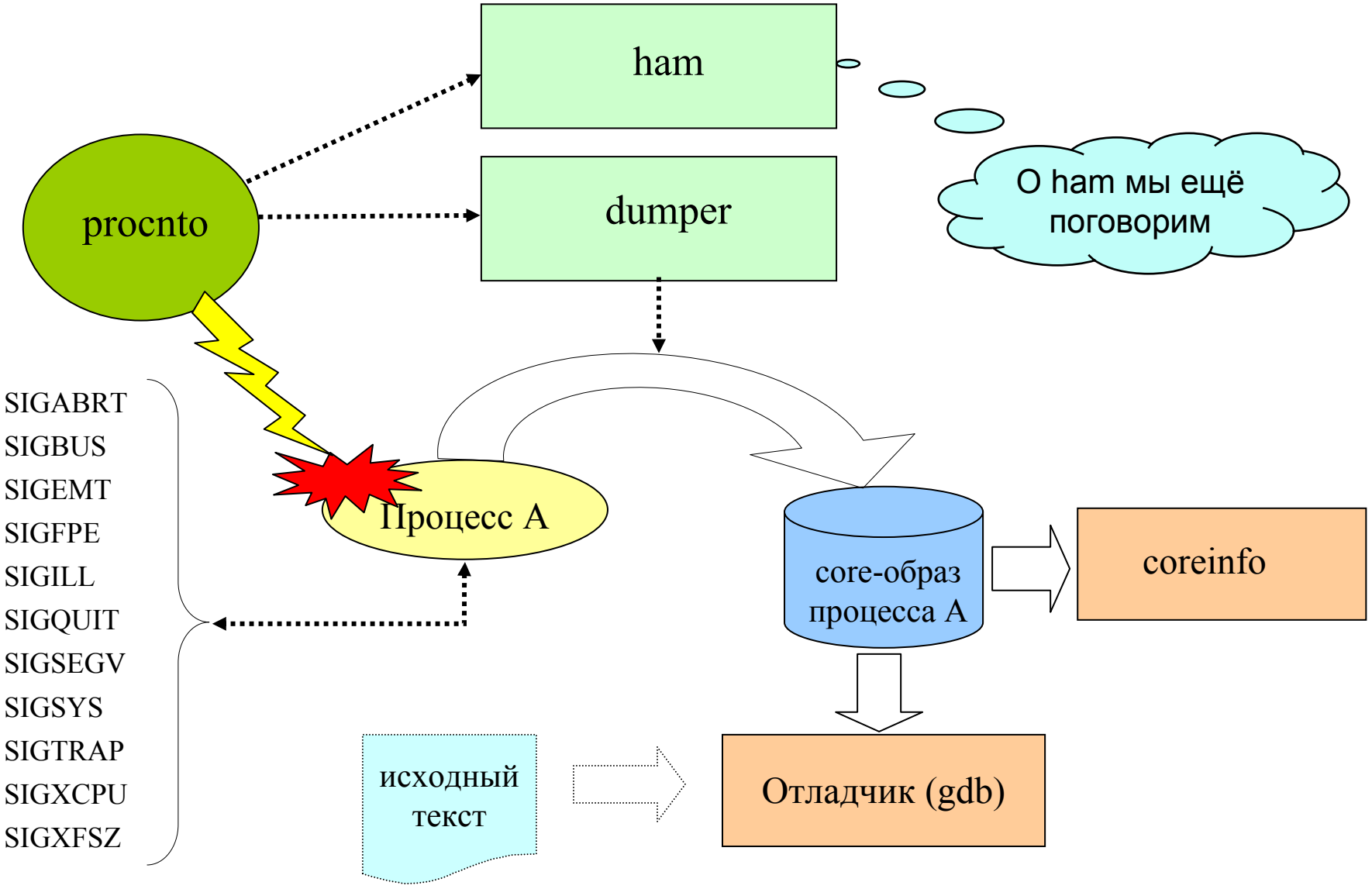
Использование флага _NTO_CHF_DISCONNECT при создании канала для извещения о событиях:

- Завершение клиента
- Разрыв клиентом всех соединений – *ConnectDetach()*
- Потеря сетевых соединений с клиентами через Qnet

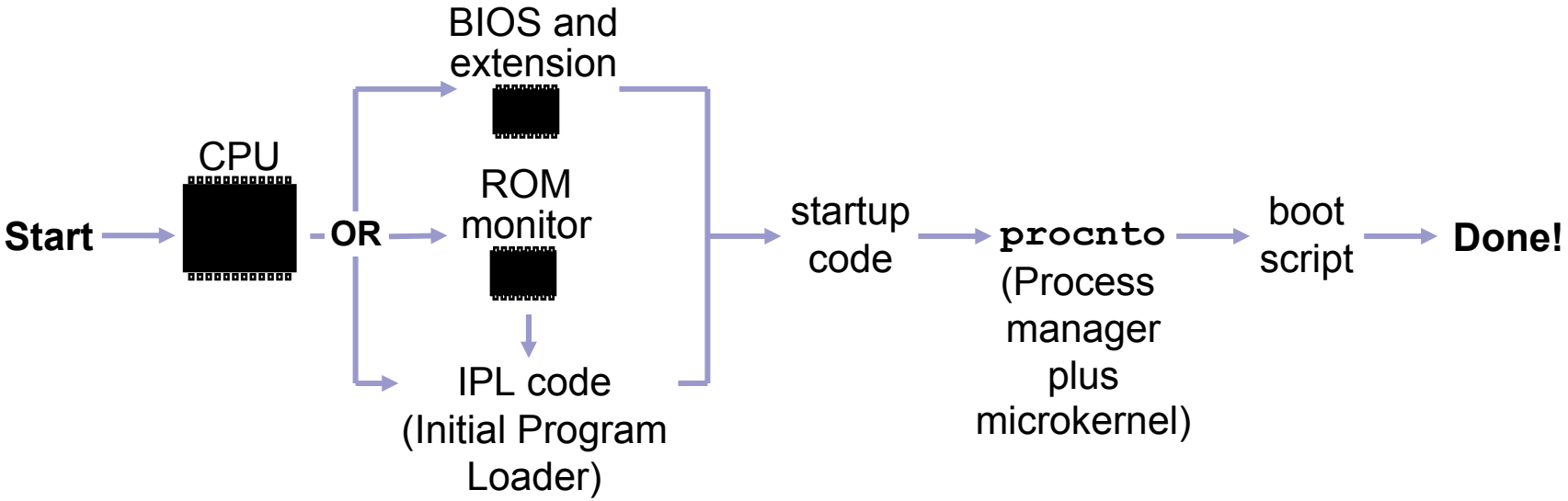



Ядро уведомляет импульсом с кодом _PULSE_CODE_DISCONNECT

# Идентификация – мониторы сбоя

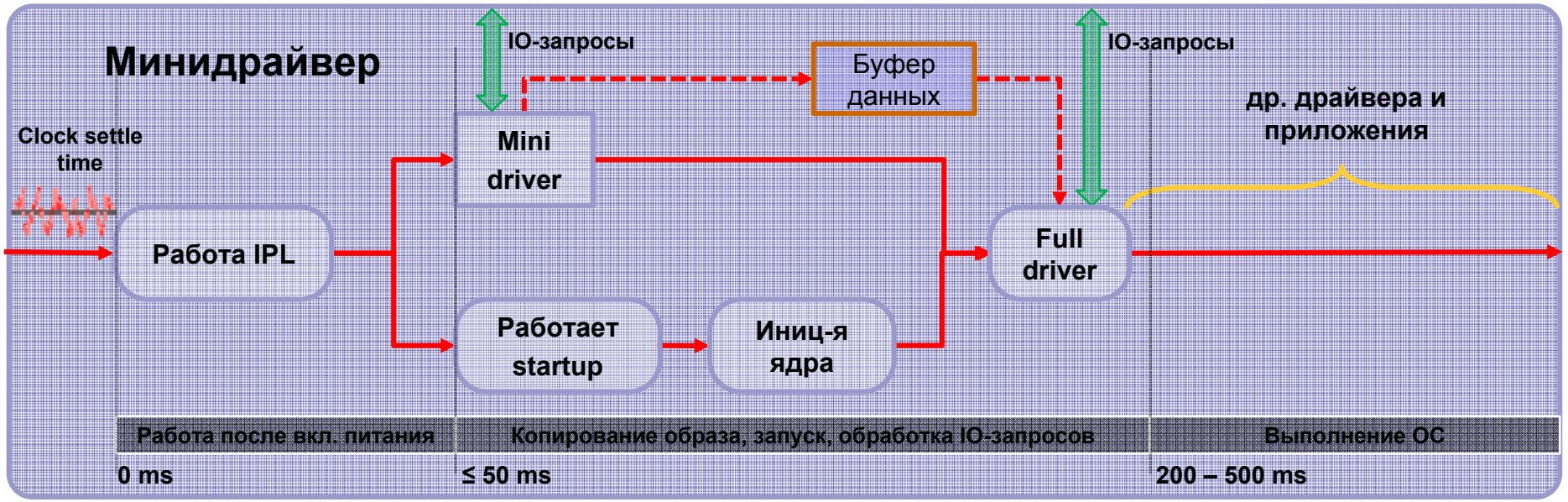


# Восстановление – QNX FastBoot



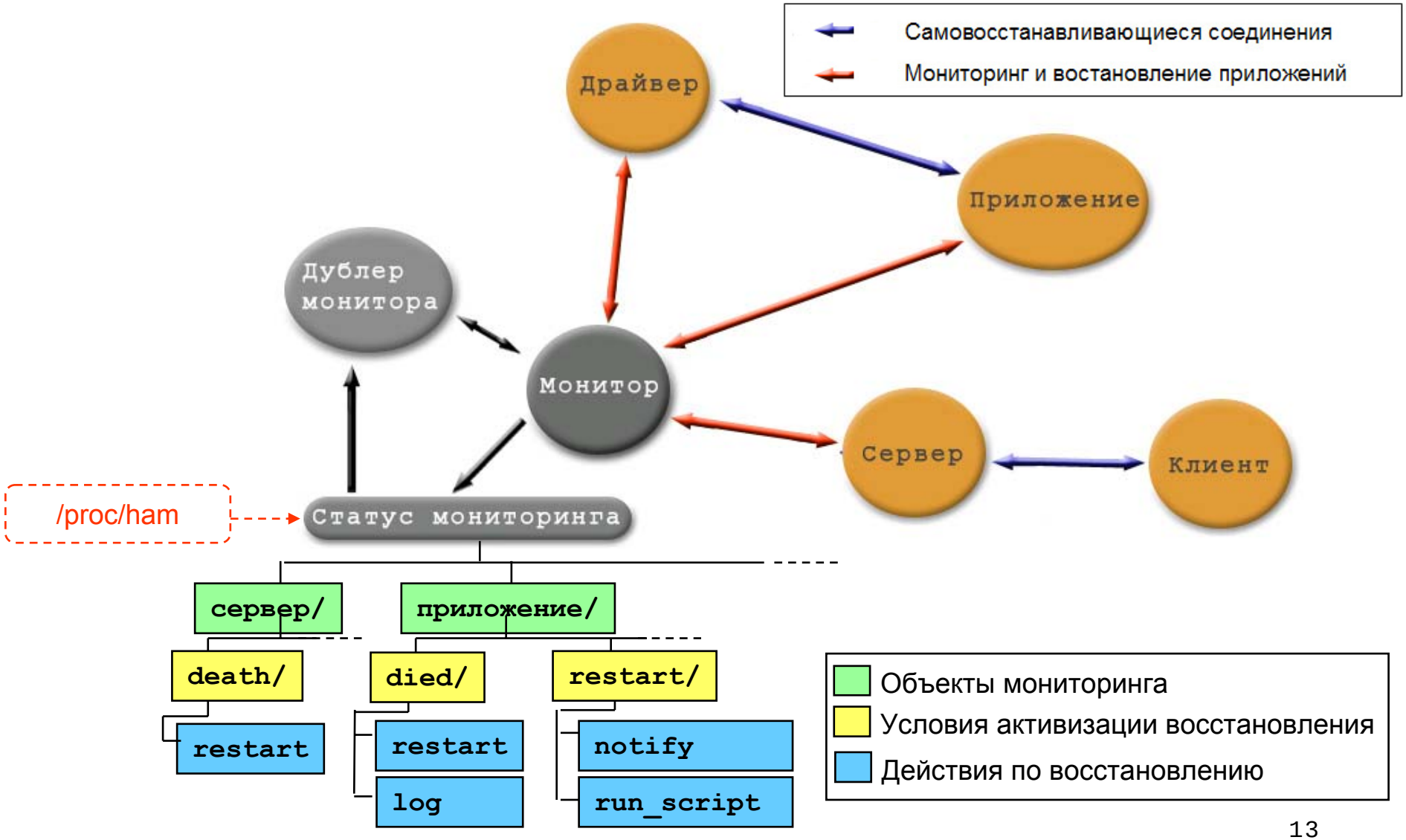

Загрузка полноценной конфигурации QNX Neutrino за 1-2 секунды.

# Восстановление – быстрая активация устройств (IDA)




✓ *Немедленный ответ на внешние запросы*
✓ *Продолжение или завершение работы прозрачным образом после того, как управление передаётся полной версии драйвера без задержек или потерь данных.*

# Восстановление – монитор ключевых процессов



# Восстановление соединений

**Цель механизма – «развязать» логику приложения и логику восстановления соединений**

Пример кода “восстановителя” соединения:

```
fd = ha_open( "/dev/device", O_RDONLY, recover_read_fd, "/dev/device", 0 );
…

int recover_read_fd( int old_fd, void *hdl )
{
  fname = (char *) hdl;
  delay(100); // Здесь выполняются необходимые действия
  new_fd = ha_reopen( old_fd, fname, O_RDONLY );
  return new_fd;
}
```

ВСЕ рассмотренные механизмы реализуются штатными средствами дистрибутива QNX SDP 6.4.1

***Спасибо за внимание!***

ООО «СВД Встраиваемые Системы»

**http://www.kpda.ru**


**Центральный офис:**
196066, г. Санкт-Петербург,
Московский проспект, д. 212 А

тел.: (812) 373-41-17
факс: (812) 373-19-07

**Технический офис:**
191014, г. Санкт-Петербург,
ул. Госпитальная, д.3

тел./факс: (812) 578-02-45