



Министерство образования и науки Российской Федерации

Московский физико-технический институт
(государственный университет)
Факультет радиотехники и кибернетики

**Реализация механизма контрольных точек
для обеспечения надежности исполнения
задач в режиме реального времени
в вычислительных комплексах Эльбрус**

Выполнил студент 318 группы Головкин Михаил Евгеньевич

Научный руководитель Гилязов Салават Састгалиевич

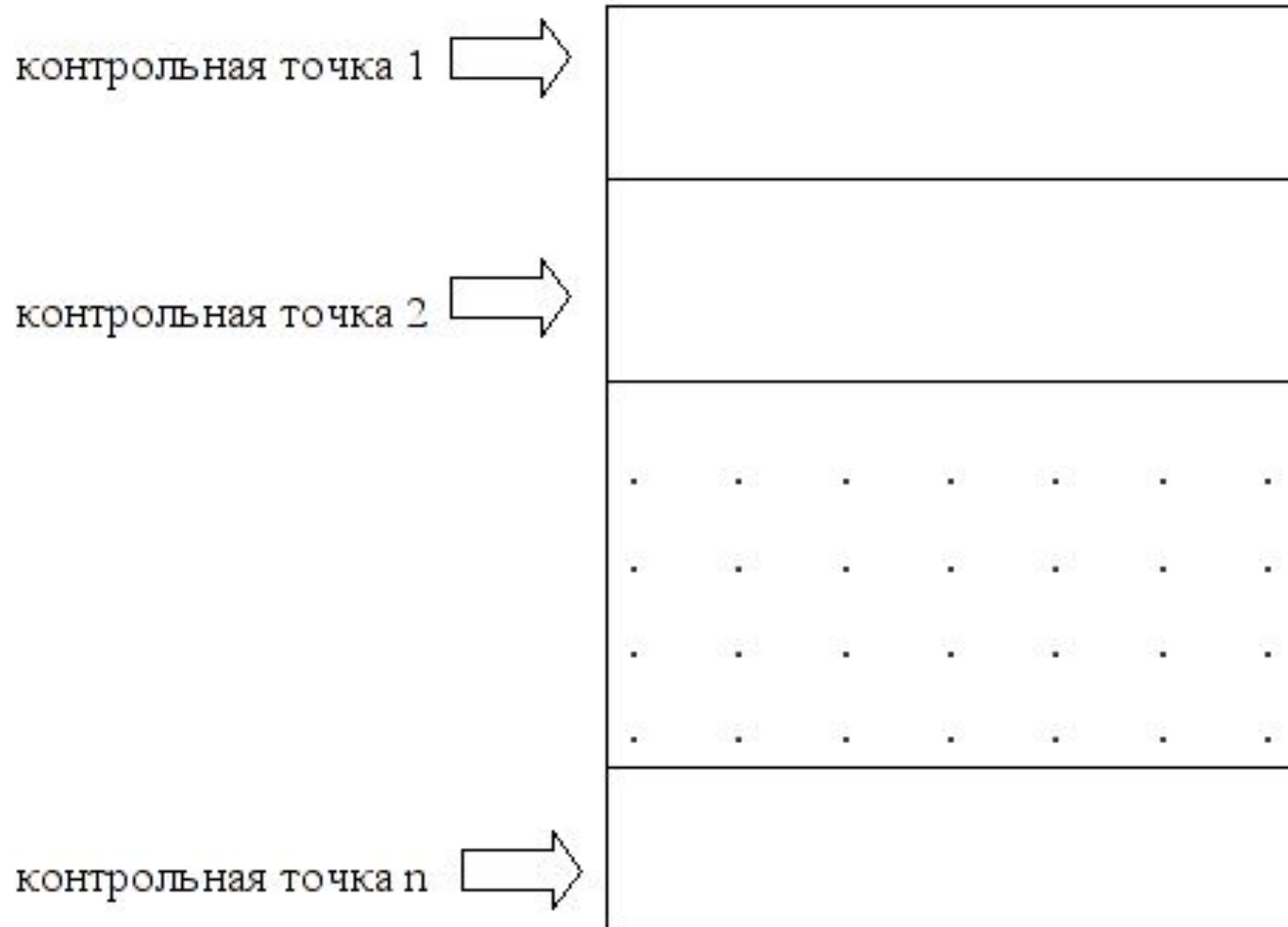
Цель работы

Реализовать механизм контрольных точек в ядре Linux для вычислительных комплексов Эльбрус, удовлетворив следующим требованиям:

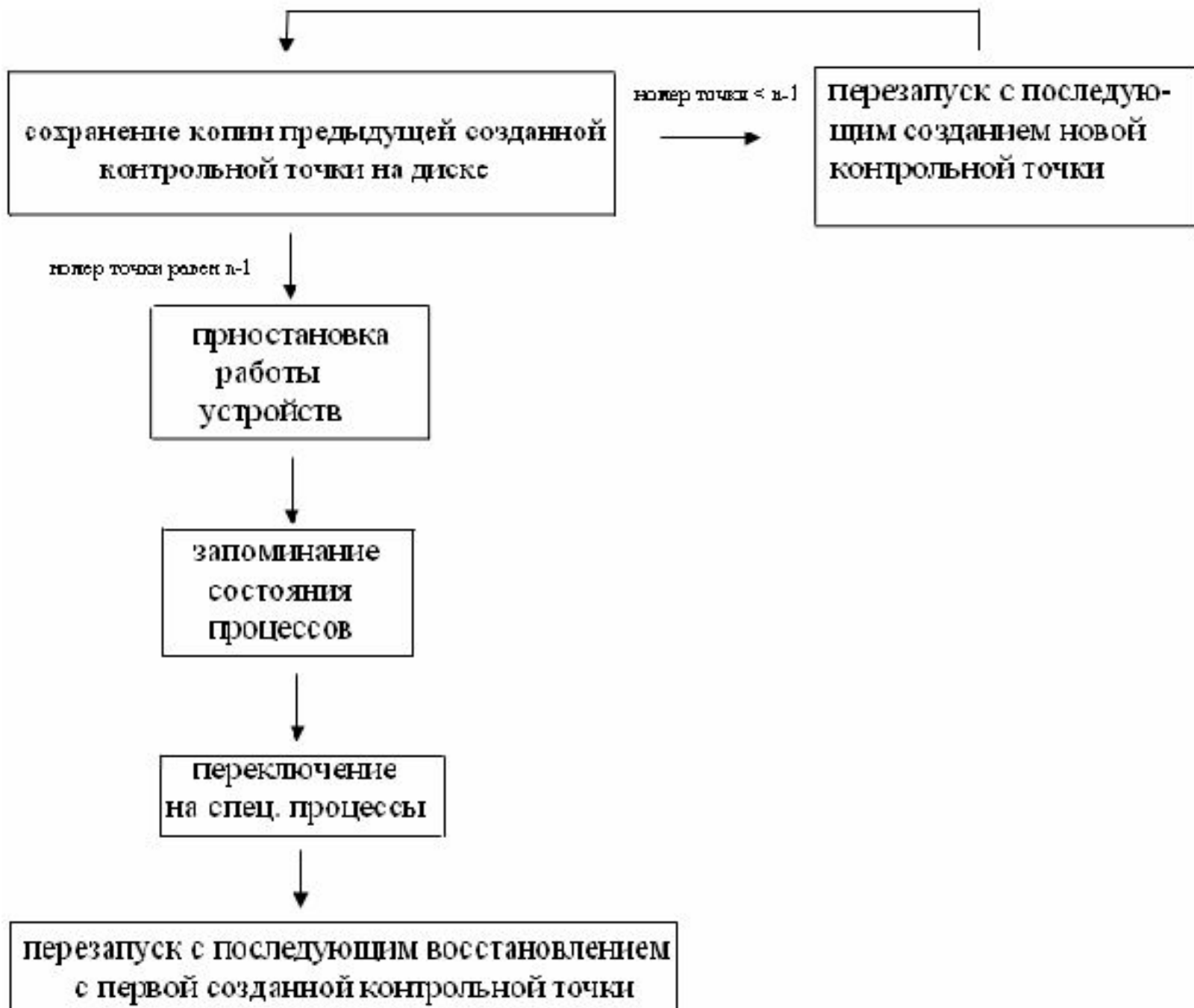
1. Максимальное ускорение работы с контрольными точками.
2. Сохранение на жестком диске резервной копии каждой точки.
3. Возможность восстановления поврежденной либо нерабочей контрольной точки в памяти.

Реализация механизма контрольных точек

Все создаваемые контрольные точки размещаются в основной памяти, что дает существенное ускорение работы механизма. При этом память делится на n частей – по числу точек, которое требуется создать



Этапы создания контрольной



Размещение копий контрольных точек на жестком диске

Для хранения копий и дампа памяти на жестком диске предусмотрен специальный раздел

hda1 /boot	hda2 swap	hda 3 /	hda4 dumpdev
---------------	--------------	------------	-----------------

Структуры данных контрольных точек

Часть файла дампа, отведенная для контрольных точек:

смещение	размер области	максимальное число точек	кол-во рабочих точек (счетчик)	
				дескрипторы
				контрольных
				точек

Все данные, необходимые для работы с контрольными точками, считываются из структуры `sntr_info` при сохранении копий на диске.

физический адрес образа ядра	физический адрес перечня узлов памяти	адрес массива областей памяти, которые не нужно сохранять	число областей не сохра- няемых страниц памяти	корректность контроль- ной точки в памяти	корректность контроль- ной точки на диске
---	--	---	---	--	--

Сохранение копии контрольной точки на диске

Отражение в память части файла дампа, отведённой для сохраняемой точки



Выделение в зонах памяти диапазонов адресов страниц, которые не нужно сохранять (принадлежащих PCI, видеопамяти и т.д.), свободных страниц, а также адресов, соответствующих ”дыркам” в физической памяти



Сохранение на диске областей памяти, занятых контрольной точкой



Снятие отражения в память части файла дампа

Сохранение состояния и перезапуск системы

«Замораживание» состояния устройств, их останов



Переключение на специальные процессы, отвечающие за перезапуск, сохранение состояния системы, восстановление и работу в режиме, когда система неактивна



Сохранение состояния регистров процесса



Перезапуск системы

Восстановление системы и возобновление работы ОС

Восстановление
в реальном
масштабе
времени

Восстановление работы ядра (очереди планирования, работа менеджера памяти, прерываний, таймеров и т.д.)



Переключение на пользовательский процесс, выполнявшийся до момента создания контрольной точки



Восстановление состояния системных устройств на момент останова перед перезапуском



Сохранение на диске копий точек, которые еще не были сохранены



Загрузка (восстановление) повреждённых либо нерабочих точек с диска

«Фоновое»
восстановление

Восстановление контрольной точки с диска

Отражение в память части файла дампа, отведённой для восстанавливаемой контрольной точки



Восстановление областей контрольной точки



Выдача системой статистики восстановленных страниц и областей



Снятие отражения в память части файла дампа

Проблемы реализации

1. Возможность повреждения контрольной точки, находящейся в основной памяти.
2. Расположение структуры `boot_info` в основной памяти.
3. Необходимость выполнить n перезапусков системы для создания n контрольных точек

Использование контрольных сумм

Контрольная сумма S_1 ,

момент времени t_1

контрольная точка 1
контрольная точка 2
.
.
.
.
.
контрольная точка n-1
контрольная точка n

Контрольная сумма $S_2 = S_1$,

момент времени t_2

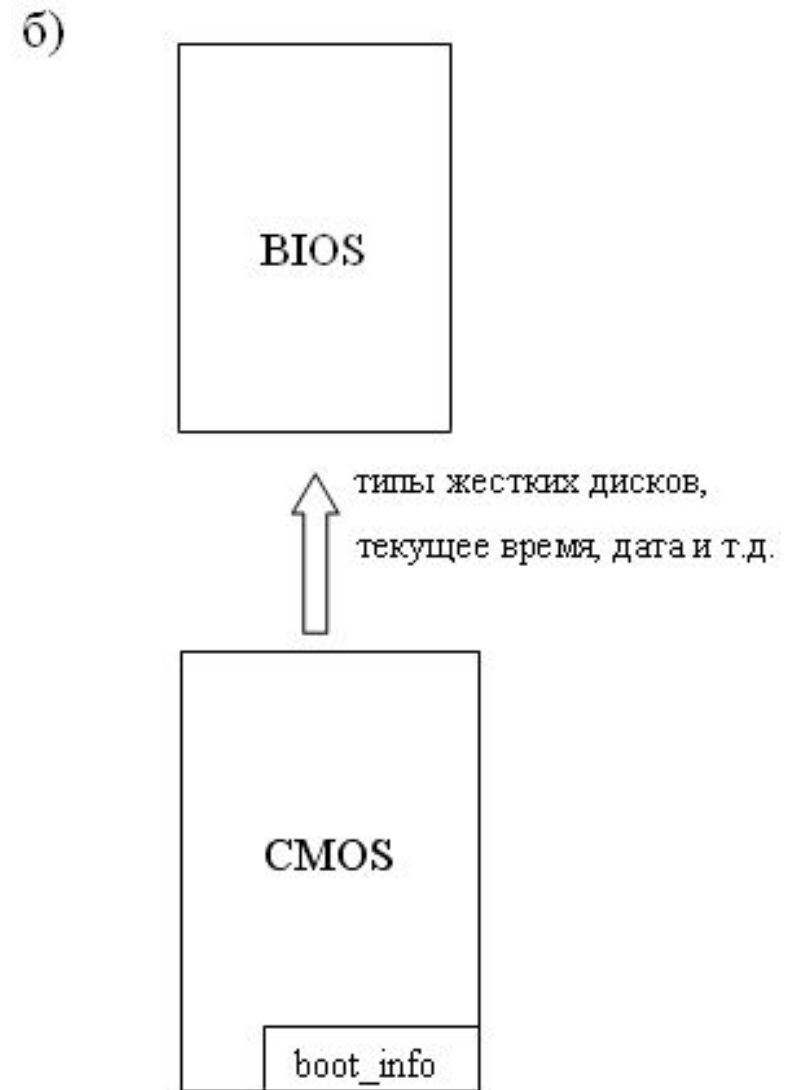
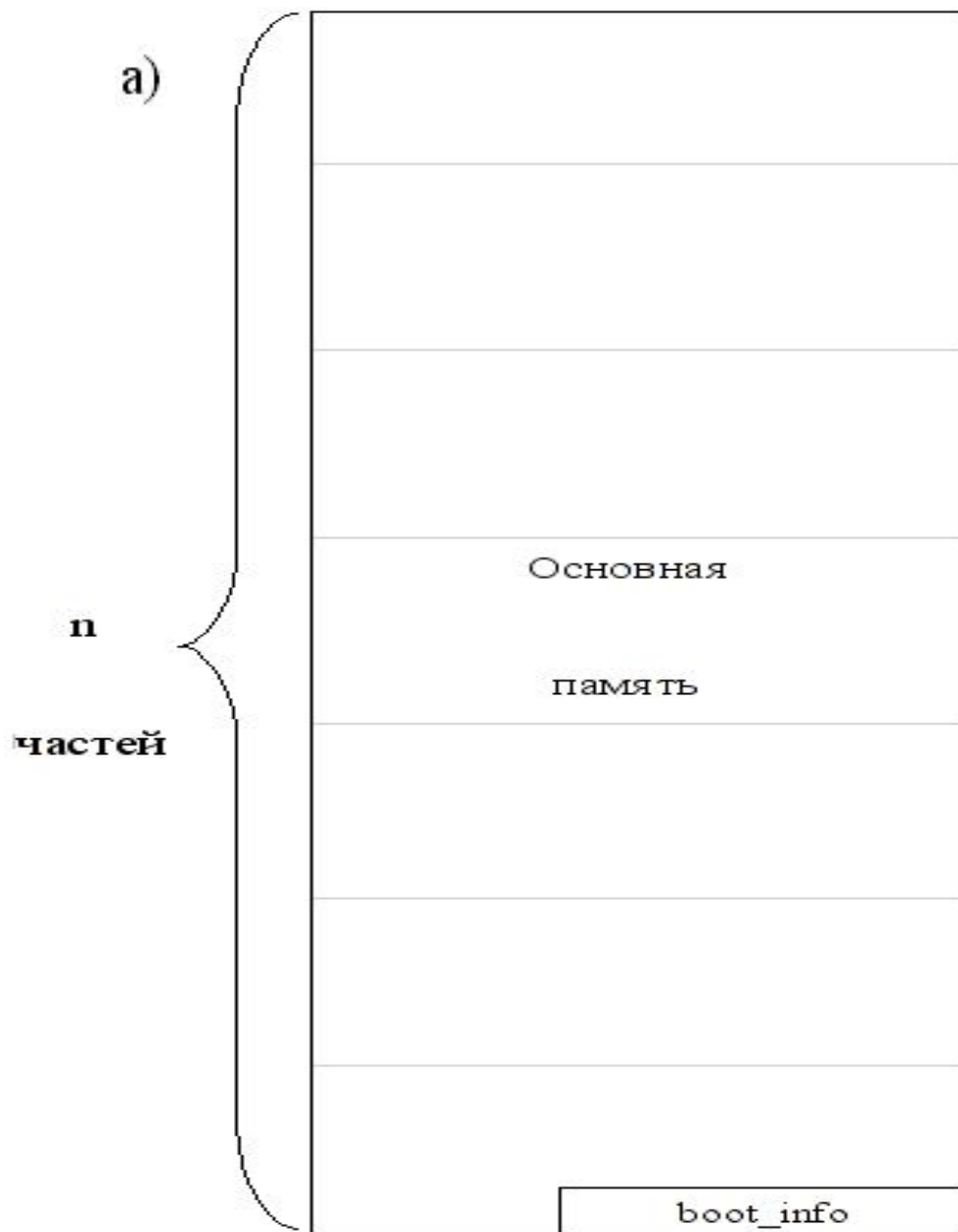
контрольная точка 1
контрольная точка 2
.
.
.
.
.
контрольная точка n-1
контрольная точка n

Контрольная сумма $S_3 \neq S_1$,

момент времени t_3

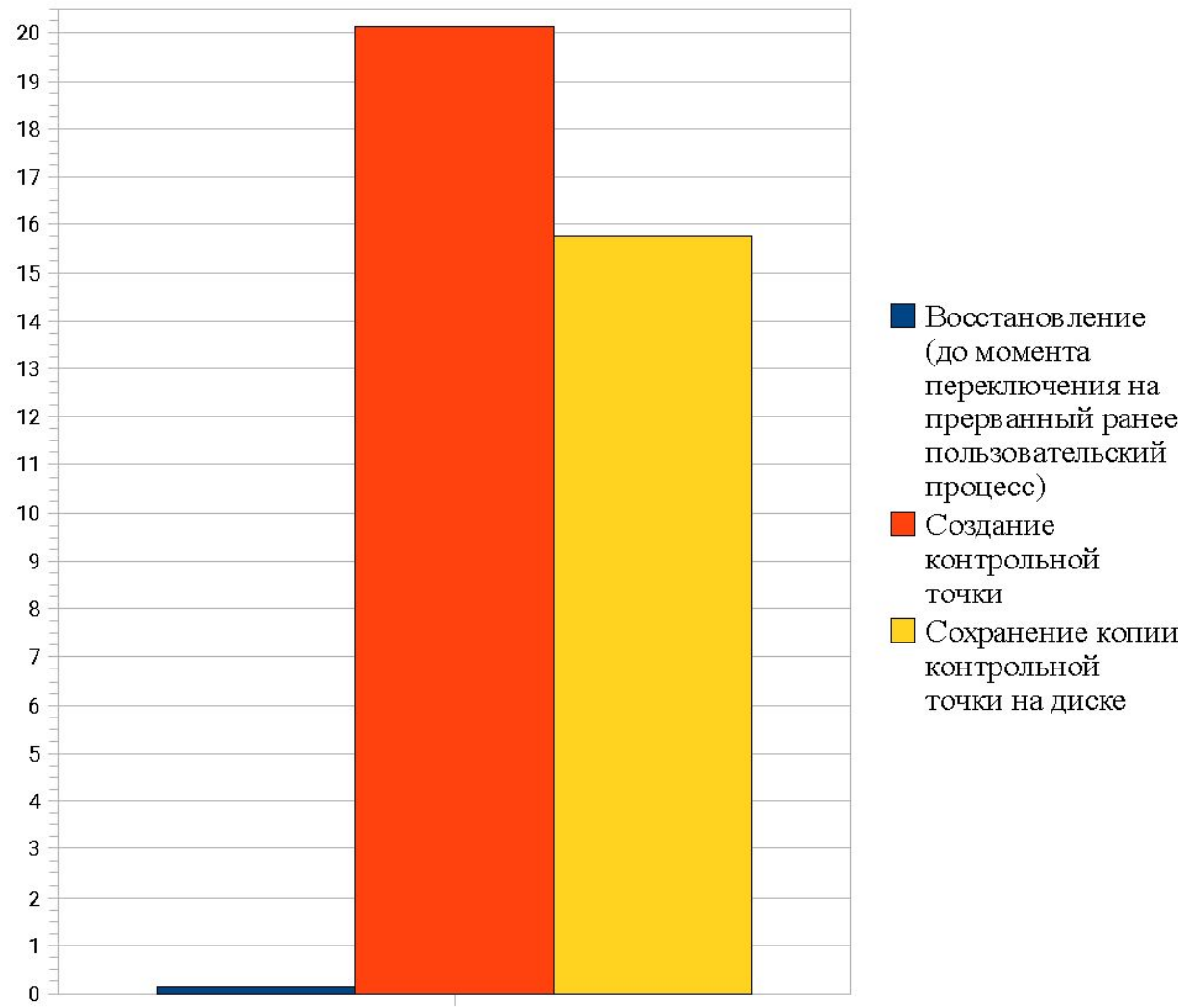
контрольная точка 1
контрольная точка 2
.
.
.
.
.
контрольная точка n-1
контрольная точка n

Размещение boot_info в CMOS



Время восстановления, создания и сохранения копии контрольной точки

Восстановление (до момента переключения на прерванный ранее пользовательский процесс)	0,12 сек
Создание контрольной точки	20,15 сек
Сохранение копии контрольной точки на диске	15,77 сек



Заключение

Для вычислительных комплексов семейства Эльбрус в ядре Linux реализован механизм контрольных точек со следующими характеристиками:

- Высокая производительность в силу расположения контрольных точек в основной памяти.
- Безопасная организация работы механизма за счет сохранения копий контрольных точек на жестком диске.

Предложенная реализация механизма контрольных точек и полученные результаты позволяют сделать вывод о высокой эффективности для дальнейшего использования.