

```
.TAILER .      <date      >      <#      >      (c)
KRONOS
.HEADER .      КОНФИГУРИРОВАНИЕ СИСТЕМЫ
abs
```

```
abs <имя_файла> {<куски памяти>}
                               {<имя дополнительного модуля>} [-ключи]
```

```
<кусок памяти> = #<имя> <размер в словах>  
absolute linker
```

"Зачем я буду связываться с опасными, - рассуждал он. - Я уж лучше с неопасными свяжусь. Но, честно сказать, и с неопасными связываться порой опасно. Лучше вообще ни с кем не связываться".

Ю. Коваль
Пять похищенных монахов

Осуществляет статическое связывание задачи, создавая образ задачи, готовый к загрузке с абсолютного адреса 0 и к исполнению, и содержащий коды всех модулей, используемых задачей. Здесь 'имя_файла' - имя файла, содержащего головной модуль собираемой задачи. Для модулей задачи порождается код инициализации.

Имеется возможность зарезервировать набор именованных областей памяти. Области памяти располагаются в младших адресах до образа задачи и могут использоваться, например, в организации прямого доступа для контроллеров, умеющих адресовать только младшую часть адресного пространства шины. Доступ к именованным областям памяти осуществляется через модуль Memory. Имя области должно быть не более чем трехбуквенным.

Вместе с основной задачей можно привязывать дополнительный набор модулей, не подлежащих инициализации, как это делается, например, при создании образа системы.

КЛЮЧИ:

- s - показ образа задачи в имитаторе пультовой микропрограммы;
- o - показ порядка инициализации;
- d - вставка DOT'ов в инициализации;
- d=<число> - начало DFT ($\geq 12h$, по умолчанию $80h$);
- s=<число> - размер DFT (по умолчанию $80h$);
- n=<число> - значение NIL;
- p=<номер> - тип процессора (1,2,5,6).

Внимание! Область 00h..10h используется для передачи информации загрузчику. DFT можно располагать с адреса 12h с учетом того факта, что на адресах 0..7Fh располагаются вектора прерываний.

Замечание. DFT - Data Frame Table; DOT и NIL -

команды. См. в томе "Архитектура процессоров семейства КРОНОС" интерпретатор системы команд.

ПРИМЕРЫ:

```
abs booter
Сборка холодного загрузчика.
```

```
abs sys #csh 4096 m2
Сборка системы с областью csh размером 4096 слов и с
модуля-компилятором.
```

```
.PAGE
.HEADER .          УТИЛИТЫ СЕРВИСА
ascii
```

```
ascii [-ключи]
aMERICAN sTANDART cODE FOR iNFORMATION iNTERCHANGE
```

Выдает по нажатию клавиши клавиатуры ее восьмеричную кодировку, в квадратных скобках - шестнадцатеричную и десятичную). Утилита прекращает работу после нажатия ^C.

КЛЮЧИ:

- h (help) - подсказка;
- e (echo) - эхо-режим, полезен для терминалов с программируемой клавиатурой, например, для обработки ESC =, ESC > для VT-52.

ПРИМЕРЫ:

```
ascii -e
На каждое нажатие клавиши утилита не выдает, а сразу
отрабатывает ее кодировку.
```

```
.PAGE
.HEADER .          КОНФИГУРИРОВАНИЕ СИСТЕМЫ
boo
```

```
boo [u=<имя_устройства>|<имя_файла_загрузчика>]
                                     [<имя_файла_системы>] [-ключи]
boot STRAP
```

Загружает в оперативную память образ задачи, например, операционной системы. Для загрузки необходимо указать холодный загрузчик (полученный с помощью утилиты -mkboot-) и образ задачи (полученный с помощью утилиты -abs-). В качестве холодного загрузчика можно указать имя блочного устройства, в нулевой блок которого он записан, или имя файла, содержащего загрузчик. Образ задачи можно взять из файла, а можно указать стандартный файл, связанный с устройством.

boo <имя_файла_загрузчика> <имя_файла_системы> -s

КЛЮЧИ:

- h - подсказка;
- f - загрузка системы из файла, содержащего образ системы; загрузчик при этом находится в нулевом блоке указанного (если не указано, то текущего) устройства;
- s - загрузка производится загрузчиком из указанного файла; система загружается из указанного файла образа системы.

ПРИМЕРЫ:

boo

Пытается загрузить систему холодным загрузчиком с текущего устройства; если устройство оказалось не системным (см. -mkboot), - увы, загружайтесь с самого начала.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

cash

cash [-ключи]
DISK cash

Работа с дисковым кэшем. Дисковый кэш - область оперативной памяти, в которой размещаются блоки внешнего устройства (носителя) с целью уменьшения интенсивности обмена с этим устройством. При этом запись всегда осуществляется на диск, а чтение может производиться из кэша.

С нулевым кэшем система не работает, начальный размер кэша указывается при сборке системы.

КЛЮЧИ:

- s - выдает статистику;
- h - подсказка;
- q - выделяет Qbus-память (только для П2.6);
- a= - добавить к кэшу указанное количество блоков;
- f= l= - добавить к кэшу мультибасовскую память, начиная с адреса f и кончая адресом l.

ПРИМЕРЫ:

cash a=<no_blocks>

Добавляет к кэшу память в количестве -no- блоков.

cash f=<first multiBus paragraph address> l=<last p.a>
Добавляет к кэшу мультибасовскую память, начиная с адреса
-f- и кончая адресом -l-.

cash a=7
Добавить к кэшу 7 блоков.

.PAGE
.HEADER . РАБОТА С ФАЙЛАМИ
cat

cat {<имя_файла>} [-ключи]
CONcatTENATE

Объединяет все перечисленные файлы в указанном
порядке. Результат по умолчанию выводится на терминал, но
может быть стандартным образом направлен в файл.

КЛЮЧИ:

-b - слияние нетекстовых файлов;
-h - подсказка.

ПРИМЕРЫ:

cat a b c >d
запись файлов -a-, -b-, -c- в файл -d-;

cat a b
объединение файлов -a- и -b- выводится на терминал;

cat a
вывод файла -a- на терминал;

cat a >/tty2
вывод файла -a- на терминал номер 2. Если при этом вместо
терминала номер 2 подключено последовательное печатающее
устройство, то файл напечатается на бумаге. Этот способ
может оказаться полезным, если у вас нет соответствующего
драйвера на печатающее устройство.

.PAGE
.HEADER . РАБОТА С ФАЙЛАМИ
ср

ср {образец} <файл-получатель> [-ключи]
сOpY FILE

Физическое копирование файлов. Образец и получатель -
имена файлов. Если файл с именем -получатель- уже
существует, его старое содержимое будет удалено. Имя
получателя имя может содержать метасимволы (* или %).

КЛЮЧИ:

- c (not Create) - попытаться скопировать в уже существующий файл, если такой имеется;
- q (not Query) - не запрашивать подтверждение на копирование;
- l (no Log) - не сообщать имена копируемых файлов;
- r (Remove) - удалить исходные файлы после копирования;
- x (eXcept) - скопировать все файлы, кроме указанных;
- m (Move) - эквивалентно -cr;
- h (Help) - подсказка.

ПРИМЕРЫ:

```
cp first second
копирование файла -first- в файл -second-.
```

```
cp *.m *.d имя_дир/
копирование файлов с расширениями .m и .d на директорию
-имя_дир- .
```

```
cp ./ ../second/*#
скопировать все файлы текущей директории, добавив к имени
каждого '#', на директорию -second-, лежащую на
директории, содержащей текущую.
```

.PAGE
.HEADER . ДРУГИЕ ФАЙЛОВЫЕ СИСТЕМЫ
cpmflx

```
cpmflx
CP/M-86 FILE EXCHANGE UTILITY
```

Переносит файлы из файловой системы CP/M-86 в файловую систему ОС Excelsior и обратно. Вызов утилиты - вход в монитор cpmflx. Утилита ВСЕГДА РАБОТАЕТ с драйвером FD0 и читает и пишет на восьмидюймовые CP/M-диски с размером трека 8 Kbyte. Заметьте, что -path-, -cd-, -mem- - не утилиты системы, а команды -shell-, поэтому в мониторе -cpmflxони не работают.

КЛЮЧИ:

Ключей нет. В мониторе имеются встроенные команды:

- stop - окончание работы;
- help - подсказка;
- dir - распечатка директории CP/M-86;
- ls - распечатка директории Excelsior;
- get - чтение файла из CP/M-86;
- put - запись файла в CP/M-86;
- bin - управление режимом чтения;

si - установка префикса латиницы в ASCII-7;
so - установка префикса кириллицы в ASCII-7;
user - смена User в CPM/86;
exc - переход к выполнению команд ОС Excelsior;
cрт - переход к выполнению команд cртflx.

ПРИМЕРЫ

cртflx

Вход в монитор утилиты.

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

cрт

cрт {<суперобразец>} <получатель>/

cOpY tREE

Копирует дерево, заданное суперобразцом, на директорию <получатель>, создавая на ней все необходимые поддиректории.

КЛЮЧИ:

-h - подсказка;
-q - не запрашивать подтверждения на копирование;
-l - не создавать поддиректорий.

ПРИМЕРЫ:

cрт /ii/doc/vers2//* /doc/version2/ -q

все файлы и поддиректории директории /ii/doc/vers2 на носителе -ii- скопируются на директорию -version2- носителя -doc-.

.PAGE

.HEADER . УТИЛИТЫ РЕДАКТИРОВАНИЯ НОСИТЕЛЯ

cre

cre <имя_файла> [-ключи] [l=size_in_blocks]
creATE FILE

Создает файл на текущей директории с указанным именем, указанной длины и с содержимым из указанных блоков. Утилита полезна для восстановления ошибочно уничтоженных файлов. Нужные блоки ищите с помощью -sc- (см.).

КЛЮЧИ:

-h - подсказка;
 -r - запросить номера блоков, которые займет создаваемый файл;
 -e - запросить размер создаваемого файла в байтах;
 l=number - создать файл размером -number- блоков.

ПРИМЕРЫ:

```
cre lost_program.m -er
```

Создается файл -lost_program.m- с запросом размера в байтах и номеров занимаемых блоков; если все блоки перечислены, нажмите CR.

.PAGE

.HEADER . УТИЛИТЫ РЕДАКТИРОВАНИЯ НОСИТЕЛЯ
 dead

```
dead [<имя носителя>|<имя_устройства>] [-ключи]
dISC eDITOR aND dIAGNOSTIC
```

Дети! Если у вас есть паровоз, - не чините его до тех пор, покуда он не сломается. Если же он будет сломан, то ремонтируйте его умеючи. Не нужно совать внутрь паровоза посторонних вещей, нельзя приклеивать трубу слюнями, потому что слюни клеят не прочно. И, главное, берегите колеса, не делайте их четырехугольными...

А.Н.Толстой. Как ни в чем не бывало

Редактирование и проверка носителя.
 Вызов без ключей - вход в монитор dead.

КЛЮЧИ:

-d - работа с носителем на указанном устройстве;
 -v (verify) - проверка носителя на корректность;
 -c (correct) - проверка с исправлением;
 -e (edit) - редактирование блоков на носителе;
 -h (help) - подсказка.

Вызов без ключей - вход в монитор dead.

Возможны команды:

v, e, h, c - аналогично ключам;
 f (find) - поиск файлов с указанным номером блока;
 l (list) - печать имен файлов с указанным Inod'ом;
 b (block) - печать имен файлов с указанным блоком;
 r (restore) - восстановление Inode;
 i (inode) - вывод информации о файле;
 m (mount) - смена носителя;
 q (quit) - выход.

Замечание. Inode - см. о структуре файловой системы в

томе "Кронос для администратора".

ПРИМЕРЫ:

```
dead -v
```

проверяется текущий (по умолчанию) носитель. Если все в порядке, выдается информация:

```
Disk Editor And Diagnostic v1.0 /01-Mar-87/ (c) KRONOS
```

```
Проверка носителя ii
```

```
Сбор информации о носителе
```

```
Информация о носителе собрана.
```

```
Размер носителя в блоках 4888, из них занято/свободно: 3352/1536
```

```
Максимальное число файлов 3258, из них занято/свободно: 1597/1661
```

Если напечаталось что-нибудь другое, например, указано число блоков, ошибочно считающихся свободными, немедленно прекратите работу с носителем - его нужно чинить.

```
.HEADER . РАБОТА С ФАЙЛАМИ
```

```
diff
```

```
diff <имя1> [<имя2>] [-ключи] [b=номер блока]
```

```
difference
```

Прицеливаясь взглядом то в одного барсука, то в другого, Похититель искал различия.

"Ага, - думал он, - у этого барсука две ноги, а у этого три. Вот вам и различие".

Ю.Коваль. Пять похищенных монахов

Здесь 'имя1' и 'имя2' - имена файлов или имена блочных устройств. Осуществляет посимвольное сравнение двух файлов и выдает таблицу кодов в шестнадцатеричном виде и соответствующих символов в ASCII. Указатель различий - символ '.'. Если указан один файл (одно устройство), на терминал выдается его дамп (шестнадцатеричное представление).

КЛЮЧИ:

```
-h (help) - подсказка;  
-l (log) - выдает все, а не только информацию о различиях;  
-d (dev) - сравнение устройств;  
b= (block)- с какого блока сравнивать.
```

ПРИМЕРЫ:

diff новый_файл старый_файл
сравнение двух файлов;

diff <имя_файла>
выдача дампа.

diff dx0 dx1 -d b=33

сравнение носителей на dx0 и dx1, начиная с 33-го блока.

.PAGE

.HEADER . УТИЛИТЫ СЕРВИСА
dt

dt [-ключи]
dAtA

"What a funny watch!" she remarked. "It
tells the day of the month, and doesn't tell
what o'clock it is!"

"Why should it," muttered the Hatter.
"Does your watch tell you what year it is?"

Lewis Carroll
Alice's adventures in Wonderland

Показывает и редактирует время в системе.

КЛЮЧИ:

-h (hELP) - выдает подсказку;
-e (eDIT) - позволяет отредактировать время;
-r (rEAD) - читает время из файла /\$v/SYSTEM.TIME
(только для систем без таймера);
-c (SEcOND) - секундомер.
Здесь \$v - имя системного носителя.

ПРИМЕРЫ:

dt -er

На экран выдается время, дата и день недели:

17:51.13 01-Jul-1987 /Wednesday/

Их можно редактировать в режиме однострочного экранного редактора.

.PAGE

.HEADER . УСТРОЙСТВА
du

du [<имя носителя>]
dISK uSAGE

Выдает информацию о занятых и свободных блоках на носителе. Если имя носителя не указано, то информация - о текущем носителе.

КЛЮЧИ:

-h - подсказка.

ПРИМЕРЫ:

```
du ii
```

На экран выдается:

Информация о носителе ii

Размер носителя в блоках 4888, из них занято/свободно 3352/1536

Максимальное число файлов 3258, из них занято/свободно 1599/1659

.PAGE

.HEADER . РЕДАКТОРЫ

e2

```
e2 <имя_файла> [-wh]
```

```
eDITOR MODULA-2
```

Компилирующий редактор. При редактировании все ключи и возможности те же, что и у -ex-. При компиляции модуля, текст которого редактировался, происходит позиционирование в тексте на первую ошибку с указанием ее характера.

КЛАВИАТУРА:

SILVER m - запуск компиляции;

SILVER e - показ следующей ошибки.

.PAGE

.HEADER . УТИЛИТЫ СЕРВИСА

echo

```
echo <последовательность_символов>
```

```
echo
```

Отрабатывает указанную последовательность символов. Предназначается для отладки терминала и клавиатуры.

КЛЮЧИ:

Ключей нет и быть не может.

ПРИМЕРЫ:

```
echo word
```

терминал отработает клавиши <w>,<o>,<r>,<d>, то есть на

экран напечатается "word".

echo <ESC>\<ESC>[?1h
утилите -echo- передается последовательность символов,
которая обрабатывается терминалом LABTAM 3000 как выход из
графического режима.

.PAGE

.HEADER . УТИЛИТЫ РЕДАКТИРОВАНИЯ НОСИТЕЛЯ

edi

edi [<блочное_устройство>] { [имя|узел] } [-ключи]
edITOR OF iNODES

Редактор таблицы файлов на блочном устройстве. О
таблице файлов см. в томе "Кронос для администратора",
файловая система. КЛЮЧИ:

-h - подсказка;
-i - для указания номера узла.

При редактировании клавиша SPACE означает: отсутствие
блока - если редактируются номера блоков; инверсию битика
- если редактируются признаки файла. Для передвижения по
меню используйте стрелки.

ПРИМЕРЫ:

edi myfile

Редактировать элементы таблицы файлов, соответствующие
'myfile'.

edi 153 -i

Редактировать элементы таблицы файлов с номером 153.

.PAGE

.HEADER . УСТРОЙСТВА

errlog

errlog <имя группы устройств> [-ключи]
errOR logIC DEVICE

Выдает статистику ошибок для группы устройств.

КЛЮЧИ:

-c (clear) - очищает выданную статистику;
-h (help) - подсказка.

.PAGE

.HEADER . РЕДАКТОРЫ

ex

ex <имя_файла> [-wh]
eDITOR MODULA-2

Экранный редактор текстов -ex- позволяет редактировать файлы шириной до 256 символов. Максимальный размер файла зависит от стека, на котором запущен -ex- (до 16000 строк). Редактор работает на любых терминалах, обслуживаемых ОС Excelsior. Для выхода из редактора нажмите клавиши

CTRL E - выход с записью всех внесенных исправлений;

CTRL C - выход без записи.

Есть возможности:

- позиционирование в файле;
- поиск в файле;
- контекстная замена;
- работа с файлами (переименование и пр.);
- работа с областью текста;
- параллельный просмотр другого файла;
- установка макросов;
- установка параметров редактирования;
- форматирование абзаца текста;
- запуск задач из shell редактора.

РАЗДЕЛИТЕЛИ

Редактор работает с текстом в кодировке КОИ-8. Строки могут быть разделены при чтении (см. библиотеку ASCII):

ASCII.HT (11c)	- заменяется при чтении на пробел;
ASCII.NL (36c)	- формат ОС Excelsior;
ASCII.NULL (00c)	- формат ОС Excelsior;
ASCII.LF (12c)	- Unix-формат;
ASCII.CR&LF (12c,15c)	- RT-11-формат.

Файл пишется с разделителем ASCII.NL. Других контрольных символов в файле НЕТ. Последняя строка завершена NL.

КЛЮЧИ:

-w (NOT wRITE) - разрешить только чтение файла.

.PAGE

КЛАВИАТУРА:

...Информационные работники всегда стремились к работе, наполненной содержанием, смыслом, это составляло их привилегию, выделяло из общей массы лиц наемного труда. Автоматизация же сводит все к нажатию кнопок.

Г.Б.Кочетков

Могущество и бессилие компьютера

Редактор использует три управляющих клавиши GOLD, SILVER и BRONZE, а также набор традиционных управляющих символов - стрелки, дублирование и т.д.. Соответствие клавиш на клавиатуре и функциональных ключей редактора

устанавливает администратор с помощью утилиты -mkboard-.

UP	- переход вверх на строку;
DOWN	- переход вниз на строку;
LEFT	- переход на символ влево;
RIGHT	- переход на символ вправо;
TAB	- табуляция вправо;
BACK TAB	- табуляция влево;
InsCh	- вставка символа с подвижкой хвоста строки вправо;
DelCh	- удаление символа с подвижкой конца строки влево;
BACKSPACE (DEL)	- замена символа слева на пробел с переходом на него курсора;
NEWLINE	- переход на начало следующей строки (line);
PageUp	- переход на 16 (число устанавливается в SETUP) строк вверх;
PageDw	- переход на 16 (число устанавливается в SETUP) строк вниз;
EraseEOL	- удаление хвоста строки;
DelLine	- удаление хвоста строки с приклеиванием хвоста следующей;
InsLine	- вставка строки с отламыванием хвоста на следующую строку (line);
DupLine	- дублирование хвоста строки;
SwapUp	- обмен хвостов вверх от курсора;
SwapDw	- обмен хвостов вниз от курсора;
LF	- стирание хвоста строки (string) и переход на начало следующей строки (line);
CR	- переход на начало текущей строки;
GOLD UP (BRONZE LEFT)	- переход к началу слова;
GOLD DOWN (BRONZE RIGHT)	- переход к концу слова;
GOLD LEFT	- переход к началу строки;
GOLD RIGHT	- переход за конец строки;
GOLD PageDw	- переход на конец текста;
GOLD PageUp	- переход к началу текста;
GOLD DelLn	- склеивание строк;
GOLD InsLn	- разрыв строки;
GOLD DelCh	- удаление слова;
GOLD DupLn	- дублирование хвоста поверх следующей строки;
GOLD BRONZE	- отмена последнего DelLn;
BRONZE GOLD	- возврат к первоначальному содержимому строки;
GOLD GOLD	- командный режим;

GOLD SILVER - установка параметров (SETUP-монитор);
SILVER SILVER - подглядка;
SILVER PgDn - выход в shell редактора (выход - PgUp).

.PAGE

КОМАНДНЫЙ РЕЖИМ:

Переход в командный режим: GOLD GOLD. На экране появится приглашение: >>> . Для возврата нажать CR.
Командная строка ::= {ПРЕФИКС} КОМАНДА [АРГУМЕНТЫ]
ПРЕФИКСЫ ::= {! | \$ | # | =}

"!" - начать поиск с начала файла;
"\$" - искать в диапазоне строк, помеченном маркерами;
"#" - рассматривать диапазон как прямоугольник;
"=" - вставить прямоугольник с вертикальной раздвижкой.

ВЫХОД В SHELL:

Для выхода в shell редактора нажмите SILVER PgDn. Shell редактора полностью аналогичен пользовательской оболочке системы, но не имеет ее команд. Таким образом, из shell редактора можно запускать любые задачи, в том числе все утилиты, компилятор, редактор и т.д.. Например, допустимо запускать один из другого несколько редакторов - только не потеряйтесь в уровнях.

Чтобы вернуться к режиму редактирования, нажмите PgUp.

.PAGE

ПОИСК В ФАЙЛЕ И ЗАМЕНА ОБРАЗЦА

Поиск производится в командном режиме.

[!\$#]"f"<образец> - поиск образца
"n" - найти образец еще раз

В этом случае вместо последовательности GOLD GOLD n CR можно набирать только SILVER n.

[!]"p"<имя_процедуры> - найти процедуру
[!]"e"<имя_процедуры> - найти конец процедуры

Смысл префиксов:

"!" - начать поиск с начала файла;
"\$" - искать в диапазоне строк, помеченном маркерами;
"#" - рассматривать диапазон как прямоугольник.

Например:

>>>fCASE

Редактор будет искать образец -CASE-, начиная с текущей строки.

Замена образца производится в командном режиме.

[!\$#]"r"/"<образец>"/"<замена> - заменить образец
Вместо символа "/" можно употребить любой разделяющий символ.

Например:
>>> #r\$first\$second
В прямоугольнике, помеченном маркерами, 'first' заменится на 'second'.

>>> r/first/second
Везде, начиная с текущей строки, при получении подтверждения, 'first' заменится на 'second'.

ПОЗИЦИОНИРОВАНИЕ :

Позиционирование в файле (N,M - числа) производится в командном режиме.

+N - перейти вперед на N строк
-M - перейти назад на N строк
N - перейти на строку с номером N
N.M - перейти на строку с номером N, колонку с номером M.

.PAGE

РАБОТА С ФАЙЛАМИ:

Производится в командном режиме.

"w" (wRITE)
- записать отредактированный файл, не покидая редактора

"s"<имя_файла> (sET NAME)
- сменить имя файла на указанное, например:
>>>sStdIO.m

Файлу присваивается имя StdIO.m.

"<"имя_донора
Считывается файл -имя_донора-.

Например:
>>> <../etc/myfile

"g"<имя_файла>
Переход к редактированию файла <имя_файла>. При этом производится запрос на сохранение отредактированного текущего файла.

.PAGE

РАБОТА С ОБЛАСТЬЮ ТЕКСТА:

ОБЛАСТЬ всегда маркируется как прямоугольная, но используется, по умолчанию, как диапазон строк. Префиксы #,= заставляют редактор произвести операцию с прямоугольной областью.

Область помечается маркерами:

GOLD GOLD [- установка маркера на начало области
GOLD GOLD] - установка маркера на конец области
GOLD GOLD . - вместо установки двух маркеров, если в
качестве области взята одна строка.

Вместо GOLD GOLD здесь можно использовать SILVER.

После того, как маркеры установлены, можно выполнить действия (в скобках указаны допустимые префиксы):

SILVER UP - спозиционироваться на начало области
SILVER DOWN - спозиционироваться на конец области
Следующие действия производятся в командном режиме
(GOLD GOLD для входа и CR для выхода):
/ - сбросить установленные маркеры
d - delete удалить область (#)
o - over наложить область (#)
i - insert вставить область (#=)
m - move переместить область
c - clear очистить область (#)
>имя_файла - записать область в файл -имя_файла-
>>имя_файла - дописать область в конец файла -имя_файла-

.PAGE

ФОРМАТИРОВАНИЕ

Можно форматировать область текста, помеченную маркерами, можно - абзац.

Параметры форматирования задаются двумя способами - устанавливаются в SETUP-мониторе (см. ex.setup) или явным способом на экране.

В SETUP-мониторе выставляются левый и правый края отформатированного текста и отступ красной строки. Эти параметры SETUP можно менять, не входя в монитор, прямо с экрана:

SILVER | - устанавливает отступ красной строки;
SILVER < - левый край отформатированного текста;
SILVER > - правый край отформатированного текста.

Все форматирование далее будет производиться с этими параметрами до их изменения.

В этом случае для форматирования нажмите SILVER и f.

Другой, экранный способ установки границ форматирования, заключается в следующем: область помечается двумя маркерами. Тогда, как и в первом случае, форматироваться будет группа строк между маркерами по вертикали (включая строки, на которых стоят маркеры), а отформатированный текст будет размещен в ширину между установленными маркерами.

В этом случае для форматирования нажмите SILVER и F.

Форматирование абзаца осуществляется с параметрами SETUP-монитора. Для форматирования нажмите SILVER и z, находясь в любой позиции абзаца.

Редактор дает возможность разместить заголовок симметрично относительно центра текста. Для этого наберите SILVER и с, находясь в любой позиции центрируемой строки.

.PAGE

ПОДГЛЯДКА:

Параллельный показ другого файла.

Для работы с подглядкой нажмите SILVER SILVER; вы попадете в меню подглядки. Движение по меню осуществляется с помощью стрелок или клавиши CR.

Укажите имя файла для просмотра, нажмите CR:

FILE <имя_файла>

Для просмотра DEFINITION MODULE библиотеки укажите ее имя, нажмите CR:

LIBRARY имя_библиотеки

Движение по файлу осуществляется с помощью стрелок и клавиш TABS, PgDw, PgUp. Возвращение в меню после просмотра файла - SILVER. Нажмите пробел при выборе действия:

HELP	- подсказка редактора;
KEYBOARD HELP	- подсказка клавиатуры;
EDIT	- вернуться к редактированию основного файла;
CLOSE	- закончить работу с указанным для просмотра файлом; это действие необязательно как для смены файла, так и для прекращения работы с просматриваемым файлом.

Вернуться к редактированию исходного файла можно и при помощи ^C.

.PAGE

МАКРОСЫ:

Употребление макросов.

Установка BRONZE-макроса:

BRONZE "{<набор_действий>BRONZE}"

На экране появляется запрос:

<<< Кнопка для макроса:

После этого выбранная вами кнопка с литерой послужит для вызова макроса.

Вызов BRONZE-макроса:

BRONZE <кнопка>

Установка и вызов GOLD-макроса производятся аналогично. Некоторые GOLD-макросы являются стандартными для редактора, записаны в файл /ii/bin/.ex.\$\$\$ и устанавливаются при вызове редактора автоматически.

Сохранение макроса:

Чтобы установленные макросы сохранились после выхода из редактора, войдите в монитор установки параметров

(SETUP-монитор) и воспользуйтесь командой SAVE SETUP. Макросы, установленные в редакторе, в том числе стандартные, запишутся в файл .ex.\$\$\$\$. При дальнейших вызовах редактора на текущей директории макросы будут считаны из этого файла, поскольку редактор ищет файл макросов по установленному пути.

.PAGE

SETUP-МОНИТОР:

Установка параметров редактирования (SETUP-монитор).

GOLD SILVER - вход в SETUP-монитор.

В SETUP-мониторе:

Для передвижения по меню используйте стрелки и TABS. После числовых параметров нажмите CR. Переключение on/off, выбор альтернативы осуществляются с помощью клавиши SPACE.

ORIGINAL SETUP	- установить параметры из файла bin/.ex.\$\$\$.
SAVE SETUP	- сохранить установленные параметры при выходе из редактора в файле .ex.\$\$\$\$. При запуске редактора на текущей директории параметры будут взяты из этого файла.
RECALL SETUP	- установить первоначальные параметры (из файла .ex.\$\$\$).
COLUMNS	- число колонок в тексте на экране. Если указанное количество символов не влезет на физический экран, продолжение строки помещается в следующую строку.
LINES	- установить число строк на экране.
PAGE	- на сколько строк вниз/вверх переводит Page Dw/Page Up.
BELL	- включить/отключить звуковой сигнал.
INSERT	- включить/отключить режим вставки.
INFO LINE	- включить/отключить показ информационной строки.
UP MRG	- с какой строчки начинать сдвиг страницы вниз.
DW MRG	- с какой строчки начинать сдвиг страницы вверх.
LEFT MRG	- левая граница отформатированного текста;
FIRST MRG	- отступ красной строки от левой границы текста (отрицательное число, если отступ влево).
RIGHT MRG	- правая граница отформатированного текста.

Для выхода из меню нажмите SILVER.

.PAGE

.HEADER .

ДРУГИЕ ФАЙЛОВЫЕ СИСТЕМЫ

flx

flx
fLExIBLE DISK

Монитор для работы с файловой системой RT-11. В мониторе flx имеются команды:

ini	[имя носителя]	- инициализация носителя;
dir[/f]	[имя носителя]	- DIR носителя;
dir/f		- DIR/FUL носителя;
dump	[имя носителя]	- дамп носителя в терминах RT-11;
del	[имя носителя]:rt11	- удаление файла rt11;
cpf	[имя носителя]:<обр> <пол>	- копирование файла <обр> из RT-11 в файл <пол> ОС Excelsior-II;
cpt/q/a excel	[имя носителя]:rt11	- копирование файла excel в файл rt11 ОС RT-11;
Необязательный ключ /q		- отмена Query;
Необязательный ключ /a		- ASCII-режим (разделитель строк LF) (конец файла ^Z [32b]);
resdx		- инициализация устройства;
help		- вывод на терминал текста подсказки;
bye, ^C		- выход из монитора;
[имя носителя]		- трехбуквенное (!) имя носителя в RT-11, например: dx0, dx1.

.PAGE
.HEADER . РАБОТА С ФАЙЛАМИ
grep

grep <текст> [отрезок] { <образец> } [-ключи]

Ищет вхождения текста в файлах, имена которых сопоставились с образцом. Если задан отрезок, то просматривается не весь файл, а только указанные строки. Отрезок имеет формат: "["начало..конец"]", где в качестве начала и конца отрезка указываются номера строк.

КЛЮЧИ:

-h - подсказка.

ПРИМЕРЫ:

grep MODULE /ii/doc/ut/*
находит слово MODULE во всех файлах директории /ii/doc/ut, печатает имена файлов и строку, в которой найдено слово:

```
cmd.m
1 - MODULE cmd; (* flm 03-Jun-87. (c) KRONOS *)
```

```
sort.m
1 - MODULE yyyy; (* Ned & flm 26-Mar-87. (c) KRONOS *)
```

```
mkut.m
1 - MODULE mkut; (*flm 02-Jun-87. (c) KRONOS *)
```

```
.PAGE
.HEADER .          УТИЛИТЫ СЕРВИСА
help
```

```
help
UTILITY help
```

Осуществляет службу подсказки в системе. При входе в монитор печатаются имена всех утилит в режиме меню; в зависимости от выбора выдается описание утилиты, в которое входит порядок вызова, этимология названия, краткое описание работы, или ключи, или примеры использования утилиты, или другие подразделы (топики), если они есть.

КЛЮЧИ:

Ключей нет. В мониторе команды:

h	- подсказка;
PgUp	- выход из нижнего уровня в верхний (и из монитора);
PgDn	- переход из верхнего уровня в нижний;
Space	- выбор нужной альтернативы;
+	- подчеркнуть топик для печати;
-	- убрать подчеркивание;
GOLD+	- подчеркнуть все топики;
GOLD-	- убрать все подчеркивание;
GOLD GOLD	- выход в командный режим.

В командном режиме:
dos <имя_файла> - оформить подчеркнутые топики в файл, готовый к печати.

```
.PAGE
.HEADER .          ВИЗУАЛИЗАЦИЯ
hy
```

```
hy
hISTORY
```

Выдает историю последнего неудавшегося процесса: номер процесса, причина прерывания, последовательность вызова процедур с указанием имен модулей, которым принадлежат процедуры, номеров строк и номеров операторов в строке (нумерация с нуля).

ПРИМЕРЫ:

hy

Выдается на экран:

```
#процесс: 1A383h причина: процесс прерван (^C) [0050h]
```

```
cmd.          | MakeNapkin:54.1 <- one:67 <-  
Files.        | LsOne:202.1 <-  
FsDnode.     | Ls0:75.1 <-  
Files.       | LsDir:207.1 <-  
cmd.         | BEGIN:82 <-  
Tasks.       | TaskStarter:318.1 <-
```

В этом примере показана история процесса, прерванного ^C, посланным с терминала. Процесс прервался в момент выполнения второго в 54-й строке оператора процедуры -MakeNapkin- из модуля -cmd-, которая, в свою очередь, была вызвана в 67 строке процедуры -one-.

В случаях, когда утилите неизвестно имя процедуры, указывается ее номер и предваряется знаком '?'. Если утилите неизвестен номер строки, выдается ее позиция в коде относительно начала процедуры.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

ins

```
ins <имя_устр> <файл_драйвера> [-sh|m=<размер памяти>]  
insTALL
```

Устанавливает драйвер устройства. В качестве имени файла утилите должно быть указано полное имя кодового файла драйвера. Если соответствующий boot-драйвер системы загружен, то все смонтированные на указанном устройстве носители демонтируются, старый драйвер удаляется, новый загружается, и затем носители монтируются снова с новым драйвером. Если драйвер не бутовый, то его нужно сначала удалить, предварительно демонтировав носители с соответствующего устройства, затем установить новый драйвер. С помощью утилит 'mou' и 'rm' можно в дальнейшем монтировать и демонтировать носители на соответствующем устройстве (см.).

```
ins {имя_устройства} -r [-sh]
```

Удаляет драйвер устройства.

КЛЮЧИ:

-r (remove) - удаление драйвера;

- s (statistic) - выдается информация о размерах П-стека драйвера, установлен ли драйвер и сколько под него отведено памяти;
- h (help) - подсказка.

ПРИМЕРЫ:

ins dx dp -r
драйверы, соответствующие устройствам dx и dp, удаляются.

ins lp /ii/etc/D180
установлен драйвер печатающего устройства.

.PAGE

.HEADER . УСТРОЙСТВА

iv

iv <устройство> <метка>
[<max. число файлов>] [-ключи] [f=<N>] [t=<M>]

INITIALIZATION & VERIFICATION

Инициализация и проверка носителей. Инициализация требуется для разметки носителя в соответствии со стандартом файловой системы ОС Excelsior. Перед тем, как проинициализировать носитель, подумайте, не осталось ли на нем какой-либо ценной информации - она будет утрачена.

Метка, указанная при инициализации носителя, будет служить именем этого носителя.

iv <устройство> -i[l]

Проверка носителя без инициализации.

КЛЮЧИ:

- h - подсказка;
- f (Formatting) - с форматированием;
- f=<N> t=<M> - форматирование части носителя;
- v (not Verify) - без проверки;
- l (not Lock) - без 'закрытия' плохих блоков;
- i - проверка (без инициализации);
- il - позволяет проверять любые носители.

ПРИМЕРЫ:

iv dx0 shu 56
Инициализация и проверка флоппи-диска; после инициализации имя носителя - -shu-, максимальное число файлов - 56.

iv dx0 -il

Проверяет носитель на устройстве dx0 без инициализации и "закрытия" испорченных блоков.

.PAGE

.HEADER . LABTAM 3000

labex

```
labex <имя_файла_шрифта> [-ключи]
labTAM eDITOR
```

Редактирует алфавит. Предназначается для создания и редактирования файла шрифта для терминала Labtam 3000. После вызова файла редактором слева на экране появляется увеличенное изображение редактируемого символа, справа - его код. После нажатия клавиши <NumLock> изображение символа можно редактировать с помощью клавиш 7,8,9,6,3,2,1, которые стирают или выставляют символ "заштрихованный прямоугольник" в соответствующей нажатой клавише позиции (вверх, вниз, вправо, влево, наискосок). Переключение "стирание- выствление" происходит после нажатия пробела. Сообщение о переключении - в правой части экрана. После редактирования символа его либо записывают (^E), либо нет (^C). Выход из редактора с записью и без записи файла шрифта осуществляется так же. Файл шрифта указывается при запуске утилиты -labtam- для инициализации терминала.

КЛЮЧИ:

```
-h      - подсказка;
-c      - создать новый файл шрифта;
-l      - загружает алфавит из указанного файла на
          терминал.
```

ПРИМЕРЫ:

```
labex /ii/bin/labtam.dfnt -l >/tty2
```

Загрузить шрифт из файла -labtam.dfnt- на терминал номер 2.

.PAGE

.HEADER . LABTAM 3000

labtam

```
labtam { <номер_терминала> [<файл_шрифта>] } [-i]
Labtam 3000
```

Инициализирует терминал LABTAM 3000, устанавливает требуемый шрифт. По умолчанию файл шрифта labtam.dfnt. Файл шрифта создается и редактируется с помощью редактора шрифта -labex- (см. labex). Терминал Labtam нуждается в инициализации после каждого выключения-включения.

```
labtam [<файл_курсора>] -c
```

Устанавливает требуемый курсор.

КЛЮЧИ:

- i - безусловная инициализация;
- b1 - установить стандартный немигающий курсор с инициализацией;
- b2 - установить стандартный мигающий курсор с инициализацией;
- o - инициализация старой клавиатуры;
- k - инициализация клавиатуры;
- s - оставить стандартный курсор;
- c - программирование курсора.

ПРИМЕРЫ:

```
labtam -b1
```

Установить немигающий курсор.

```
labtam 3 /ii/users/my/star.dfnt
```

Загрузится шрифт из файла -star.dfnt-, созданного с помощью утилиты -labex- (см. -labex-).

```
labtam crs -c
```

Установить курсор из файла курсора -crs-. Файл курсора представляет собой прямоугольник высотой 12 символов и шириной 8 символов, заполненный пробелами и другими символами по вкусу.

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

ln

```
ln <образец> <новое_имя>
```

```
Link file
```

Присваивает файлу еще одно имя. При этом физического копирования файла не происходит, лишь число связей файла увеличивается на одну, отсюда термин "привязать файл". Все привязанные к файлу имена совершенно равноправны.

КЛЮЧИ:

- h (Help) - подсказка.

ПРИМЕРЫ:

```
ln /vv/we/sem/Model.m Model.m
```


привязать файл Model.m, лежащий на директории /vv/we/sem/, к текущей директории под тем же именем.

```
ln /vv/we/sem/Model.m /ii/sem/Schem.m
```

привязать файл Model.m, лежащий на директории /vv/we/sem/, к директории sem на носителе ii под именем Schem.m.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

load

```
load {<имя_задачи>} [-ключи]
load PROGRAM IN MEMORY
```

Загружает в оперативную память программу (например, утилиту). В качестве имени задачи указывается полное имя кодофайла.

Задача останется в оперативной памяти до перезагрузки системы.

КЛЮЧИ:

-h - подсказка;
-s - выдает статистику;
m= - размер памяти в словах, в которую загружается задача. После загрузки задачи и выделения стека лишняя память освобождается.

ПРИМЕРЫ:

```
load /ii/bin/help
```

Загрузить в оперативную память утилиту -help-.

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

ls

```
ls {образец} [-ключи]
```

```
llst FILE NAMES
```

Выдает на экран имена файлов, удачно сопоставившихся с образцом. Имена файлов с расширителями .cod и .sum, а также начинающиеся с литеры "." (точка) печатаются только по ключам. Запуск -ls- без аргументов выводит на экран имена файлов текущей директории.

```
ls /dev/
```

Выдает список устройств и информацию о них.

КЛЮЧИ:

- x (eXcept) - о файлах, кроме указанных;
- a (all) - о ВСЕХ файлах, в том числе .*, *.cod, *.sym;
- r (ref) - о занимаемых блоках на диске;
- l (long) - длинный формат;
- v (volume) - имя носителя, на котором физически размещен файл;
- c (create) - время создания файла;
- w (write) - время последней записи в файл;
- 1 - печать имен файлов в одну колонку;
- 2 - печать имен файлов в две колонки;
- 4 - печать имен файлов в четыре колонки;
- h (help) - подсказка;
- ls / -l - выдает информацию о числе открытых файлов на носителях;
- ls - эквивалентно ls *.

Замечание. Ключи -l, -r автоматически выставляют ключ -a.

ПРИМЕРЫ:

ls

выдается список имен всех файлов на текущей директории;

ls /dev/

выдается список устройств и информация о них (имена носителей), например:

Устройства:

t2 #: {2}	последовательное
mw #: {0..2}	блочное
tt #: {0}	последовательное

Носители:

ii [mw0] блоков: 4888 max файлов: 3258

.PAGE

.HEADER . КОМПИЛЯТОРЫ

m2

m2 {<имя_файла>} [-ключи]

MODULA-2 COMPILER

Модуль-2 компилятор. В результате компиляции DEFINITION MODULE получается симфайл, IMPLEMENTATION MODULE и MODULE - ref-файл и кодофайл одновременно. Подробнее об именах файлов читайте в разделе 'ИМЕНА'. В компиляторе реализованы некоторые расширения Модуль-2 (см. в томе "Кронос для программистов"). Об ограничениях реализации см. раздел 'ОГРАНИЧЕНИЯ'.

КЛЮЧИ:

- e - запрет расширений языка;
- a - печать всех ошибок (по умолчанию первые 6);
- t - не проверять границы массивов;
- b - не проверять границы отрезков;
- f - прерваться по первой ошибке;
- p=<число> - генерация кода для KRONOS-P2.<число>.

ИМЕНА:

Имена симфайлов и кодофайлов получаются из имени модуля (не из имени файла) добавлением расширителей '.sym' и '.cod' соответственно.

ПРИМЕР:

```
DEFINITION MODULE xxx;      => xxx.sym
IMPLEMENTATION MODULE xxx; => xxx.cod
```

ОГРАНИЧЕНИЯ:

Текущая реализация компилятора допускает не более:

- 16 выходов EXIT из LOOP-цикла;
- 256 символов в строчной константе;
- 64 имен в списке имен;
- 7 параметров у процедуры (ARRAY OF и SEQ передаются двумя параметрами).
- 256 байт размах значений меток в операторе CASE. Метки только положительные.

ПРИМЕРЫ:

```
m2 exf.m
```

Если в процессе компиляции не обнаружено ошибок, на экран выдается сообщение:

```
Modula 2.5 [P2.5] (c) KRONOS. v1.5 /15-Nov-87/ <1> "exf.m"
  строк: 11 время: 24с код: 239w   exf.cod
```

В результате компиляции на текущей директории появятся файлы exf.sym и exf.cod.

Если в тексте модуля были обнаружены ошибки, о них выдается сообщение:

```
Ожидался символ ';'
104:  PROCEDURE Next(i,j: INTEGER$,A: ARRAY OF CHAR);
```

```
Число ошибок: 1
```

Выдается сообщение о характере ошибки, приводится номер строки и сама строка, в которой допущена ошибка, а также помечается предположительно место в строке, где надо

искать ошибку (символом '\$').

.PAGE

.HEADER . УТИЛИТЫ СЕРВИСА

mail

```
mail { <номер терминала> } <сообщение>
mail
```

Посылает сообщение на терминал с указанным номером.

КЛЮЧИ:

Ключей нет.

ПРИМЕРЫ:

```
mail 2 3 Бойся падения метеоритов!
```

На терминалы 2 и 3 выдается указанное сообщение.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

mkboard

```
mkboard
mAKE kEYboard
```

Устанавливает функциональные ключи на клавиши терминала. Работает в режиме монитора.

При установке функциональных ключей они сразу начинают обрабатываться на всех терминалах того типа, на котором запускается -mkboard-, и сохраняются до перезагрузки системы или до нового переопределения утилитой -mkboard-. Для того, чтобы установленные ключи сохранились при перезагрузке, необходимо выйти из монитора с записью. В этом случае установленные функциональные ключи запишутся в файл 'kb(N).keys', где N - тип терминала. Этот файл используется утилитой -stty- для инициализации клавиатуры при загрузке.

КЛЮЧИ:

Ключей нет. В мониторе имеются команды:

e	- выход с записью;
q	- выход без записи;
m	- показать другие меню;
Arrows,CR	- выбрать раздел;
n	- создать числовую последовательность;
SPACE BAR	- создать символьную последовательность;
LF	- перейти на следующую последовательность.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

mkboot

```
mkboot [<имя_устр>] <имя_файла> [-ключи] [a=0|a=1]
mAKE boot VOLUME
```

Позволяет ассоциировать с устройством холодный загрузчик и образ системы.

Замечание. После -mkboot- нельзя удалять или перемещать файл с образом системы.

КЛЮЧИ:

- h - подсказка;
- c - записывает файл, как холодный (cold) загрузчик на указанное устройство. Имя устройства указывать обязательно;
- d - записывает ссылку на файл, содержащий образ системы, в область холодного загрузчика тома на указанное устройство. Имя устройства указывать обязательно;
- a=0/1 - autorun - отключить/включить автозагрузку.

ПРИМЕРЫ:

```
mkboot dx0 bootRK0 -c
Записать в нулевой блок устройства dx0 файл bootRK0 как загрузчик с RK0.
```

```
mkboot dx0 /sys/etc/ExcII -d
При загрузке с dx0 будет прочитан образ системы из файла /sys/etc/ExcII на RK0.
```

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

mkdir

```
mkdir {имя_директории}
mAKE dirECTORY
```

Создает новую директорию.

КЛЮЧИ:

Ключей нет.

ПРИМЕРЫ:

```
mkdir /ii/we/user
создать на директории -we- директорию -user-.
```

```
mkdir user
```

создать на текущей директории директорию -user-.

.PAGE

.HEADER . УСТРОЙСТВА

моу

```
моу {<имя_устройства>} [-hr]
моуNT
```

Монтирует носитель на блочном устройстве 'имя_устройства'. Если на устройстве уже смонтирован носитель, производится попытка его демонтировать. Посмотреть, какие носители смонтированы в системе, можно с помощью утилиты -ls- (ls /dev/).

```
моу имя_устройства [текстовый_параметр] [-adsr]
```

Монтирует логическое устройство, передавая драйверу в качестве текстового параметра полное имя файла, в котором будет размещен псевдодиск. Такой файл можно создать с помощью утилиты -cre-. Созданный псевдодиск можно пометить, запустив -iv-. Утилиту удобно использовать, например, для получения загрузочных носителей, создавая их на псевдодиске и копируя затем псевдодиск на накопитель 'один в один' (см. -volc-).

КЛЮЧИ:

```
-h - подсказка;
-r - только демонтирует носитель с устройства;
-a - assign привязывание псевдодиска;
-d - deassign отвязывание псевдодиска;
-s - show показ статистики.
```

ПРИМЕРЫ:

Пример 1.

```
моу dx0 dx1
```

На устройствах dx0, dx1 монтируются носители. Имена выдаются на экран.

Пример 2. Известны драйверы логических устройств:

LD для ОС Excelsior-II. Например:

```
моу ld3 /ii/sys/SYS.DISK -a
моу ld -s
```

RN для RTM Network DL K/C ОС RT-11. Например:

```
моу rn3 VD1:KRON3.USK#PASSWORD -a
```

Пример 3. Порядок создания псевдодиска:

```
cre /ii/squ/sys.dsk l=62
```

На директории /ii/squ создается файл длиной 62 блока.

```
моу ld7 /ii/squ/sys.dsk
```

В созданном файле размещается псевдодиск ld7.

iv ld7 sys -v

Псевдодиск инициализируется с меткой -sys- (см. -iv-).

.PAGE

.HEADER . ВИЗУАЛИЗАЦИЯ

ms

ms {<образец>} [-ключи]

MODULA-2 sTATIC

Исследует модули, имена которых удачно сопоставились с образцом. Выдает суммарную информацию об этих модулях.

КЛЮЧИ:

-h (hELP)	- подсказка;
-d (dEFINITION)	- без определяющих модулей;
-m (ImPLEMENTATION)	- без реализующих модулей;
-c (cODEFILE)	- без кодофайлов;
-i	- статистика команд.

ПРИМЕРЫ:

ms /ii/util/ex/ex*

Выдается следующая информация:

MODULEs:	DEF	MOD	
exIO	6	196	
exCmd	13	235	
exMain	133	1363	
exSetUp	28	317	
exMem	45	424	
exHead	22	67	
exMacro	25	177	
exPeep	9	307	
exTex	14	128	
exm	-	47	
ex	-	142	
TOTAL:	295	3403	in 20 modules.

CODEs:	CODE	GLO	STRINGS
exMain	7784	976	424
exPeep	2324	484	276

TOTAL: 10108 code 1460 globals 700 strings in 2 modules.

T O T A L: 3698 lines 1460 globals 700 strings 10108 code
12396 bytes MIN MEMORY for the program

.PAGE
.HEADER . РАБОТА С ФАЙЛАМИ
mv

```
mv {<образец>} <новое_имя> [-ключи]  
mOvE FILE
```

Заменяет имя файла на новое. Действие утилиты такое же, как у `-ln-`, но с отвязыванием старого имени.

Имя директории можно заменить только в пределах родительской директории.

КЛЮЧИ:

- h - подсказка;
- q - не запрашивать подтверждение на переименование;
- l - не печатать имя файла на экран.

ПРИМЕРЫ:

```
mv файл1 файл2  
переименовать файл1 в файл2;
```

```
mv {<имя_файла>} другая/  
перемещение файла 'имя_файла' с текущей директории в  
директорию 'другая'.
```

.PAGE
.HEADER . УТИЛИТЫ СЕРВИСА
pr

```
pr {<имя файла>} [-ключи] {<числ_ключ>=N}  
prINT FILE
```

Командный файл вывода текстов на печатающее устройство. Командный файл содержит в себе запуск утилиты печати на конкретное устройство, в зависимости от конфигурации системы. Помимо дополнительных возможностей, утилита печати выполняет ряд обязательных функций, обрабатываемых с помощью запуска стандартных для всех утилит ключей. Перед запуском печати должен быть установлен соответствующий драйвер (см. `-ins-`).

КЛЮЧИ:

- m - выделение ключевых слов для Модуля-2 файлов;
- r - при печати на рулонную бумагу;
- H/T - без печати header/tail;
- f - без оформления страницы заголовками;
- t - вывод на терминал;
- z - сквозная нумерация страниц;
- s - печать через один интервал;

p= - номер на первой странице файла;
f= - номер на первой напечатанной странице;
l= - номер на последней напечатанной странице;
P= - размер страницы;
m= - размер левого поля в символах.

ПРИМЕРЫ:

```
ins lp /ii/etc/LP5rb6329
Установлен драйвер печатающего устройства Robotron CM
6329.01-M.
```

```
pr myfile p=2 f=13
Страницы будут пронумерованы с номера 2, печать
начнется с 13-й страницы.
```

```
pr /ii/util/m2/mc* -z
Напечатаются все файлы, начинающиеся с литер -mc-, с
нумерацией страниц. Нумерация сквозная.
```

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

rm

```
rm {образец} [-ключи]
rEmOVE FILE
```

Удаляет с указанных директорий файлы, имена которых удачно сопоставились с образцами. На самом деле -rm- только отвязывает файлы (см. -ln-, "привязать/отвязать файл"). Файл удаляется физически, если он не привязан ни к одной директории.

Чтобы удалить директорию, необходимо сначала удалить все файлы на этой директории.

```
rm "/"<имя_носителя>
```

Демонтирует носитель. Носитель демонтируется, если на нем нет открытых файлов. На носителе есть открытые файлы, если на нем находится чья-нибудь текущая директория, или файлы с него есть в чем-либо пути, или кто-то работает с файлами этого носителя.

КЛЮЧИ:

-x - удалить все файлы, кроме указанных;
-q - не запрашивать подтверждение на удаление;
-l - не сообщать имена удаляемых файлов;
-h - подсказка.

ПРИМЕРЫ:

```
rm {<имя_файла>}
удаление файла 'имя_файла' с текущей директории;
```

```
rm * -q
удаление всех файлов с текущей директории;
```

```
rm *.cod
удаление всех кодовых файлов с текущей директории;
```

```
rm /ABC/Test
удаление файла Test на носителе ABC.
```

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

run

```
run <имя_задачи> <параметры_задачи> [-ключи]
run
```

Запускает задачу без ожидания ее завершения.
Внимание! Отведенная память не освобождается! Задача
запускается навсегда.
Замечание. Запускаемой задаче нельзя передавать ключи.

КЛЮЧИ:

-s (statistic)	- выдает оценку размеров П-стека и размер памяти, отведенной под задачу;
-h (help)	- подсказка;
s=	- размер стека задачи. По умолчанию 1024 слова (4K);
m=	- размер памяти, в которую загружается задача. После загрузки задачи и выделения стека лишняя память освобождается.

ПРИМЕРЫ:

```
run sh 2 ii s=512
Запускает -sh- на стеке 512 слов, передавая ей два параметра - номер терминала и имя системного диска.
```

.PAGE

.HEADER . УТИЛИТЫ РЕДАКТИРОВАНИЯ НОСИТЕЛЯ

sc

```
sc -<ключи> [f=<номер>] [t=<номер>]
scANNER
```

Ищет указанные блоки на текущем носителе. Удобна для нахождения ошибочно удаленных файлов. Если известны блоки, которые занимал отвязанный файл, с помощью утилиты -sc- можно восстановить его.

КЛЮЧИ:

-h - подсказка;
-f <char> - поиск блоков, начинающихся с указанной
литеры;
-s { no } - показ первой строки блоков с номерами -no-;
-r - показ первой строки блоков;
f=<number>, t=<number> - диапазон блоков from..to.

ПРИМЕРЫ:

```
sc -r f=5
```

показать первые строки блоков, начиная с пятого.

```
sc -f D
```

показать первые строки блоков, начинающихся с 'D', в том
числе все "DEFINITION MODULE".

.PAGE

.HEADER . SHELL
sh

```
sh {номер_терминала}  
shELL
```

Утилита -sh- предназначена для ведения диалога с
системой. После запуска -sh- на терминале доступны ее
команды path, cd, set, us, mem, quit, bye, hi, news.
Подробное описание команд приведено в томе "Кронос для
начинающих".

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ
size

```
size {<образец>} [-ключи]  
FILE size
```

Подсчитывает размер файлов и поддеревьев, удачно
сопоставившихся с образцом. Если в качестве образца
указано имя директории, в процессе подсчета на экран
печатаются имена файлов и поддиректорий этой директории.

КЛЮЧИ:

-v - отключает печать имен промежуточных файлов;
-h - подсказка.

ПРИМЕРЫ:

```
size ooo
```

выдается размер файла -ooo- в байтах и число занимаемых блоков:

```
ooo          |   байтов:   21096   блоков:   6
.PAGE
.HEADER .      КОНФИГУРИРОВАНИЕ СИСТЕМЫ
sp
```

```
sp  [-ключи]
sET pARAMETERS
```

Изменяет параметры системы и показывает их.

КЛЮЧИ:

- d - вывод системной DFT;
- i - включение-выключение демонстрации порядка инициализации модулей после загрузки задачи;
- l - включение-выключение демонстрации порядка загрузки модулей при загрузке задачи;
- e - включение-выключение показа стека внешних;
- g - включение-выключение игнорирования конфликтов версий;
- c - вывод конфигурации;

```
sp <module name> -off  исключения модуля из конфигурации;
sp <module name> -on   включение исключенного модуля.
```

ПРИМЕРЫ:

```
sp -g
После этого во всей системе до перезагрузки или до
следующего вызова этой утилиты будет игнорироваться
конфликт версий.
```

```
sp exPublic -off
sp exPublic# -on
```

Модуль редактора -exPublic- заменяется на отладочный.

```
.PAGE
.HEADER .      РАБОТА С ФАЙЛАМИ
squ
```

```
squ {образец} <файл_пакет> [-ключи]
squASH & SMOOTH FILES
```

Упаковывает файлы, имена которых удачно сопоставились с образцом, записывая их в указанном порядке в файл-пакет. К указанному имени файла-пакета добавляет расширитель .squ.

```
squ <файл_пакет> [-q]
```

Распаковывает пакет. Поскольку каждый файл пакета распаковывается с запросом на распаковку, можно распаковать лишь часть пакета (если, например, для всех файлов заведомо не хватит места на носителе). При этом сам пакет остается без изменений.

КЛЮЧИ:

-h - подсказка;
-q (query) - не запрашивать подтверждение на распаковку/упаковку;
-p (pathnames)- записывать полные имена файлов.

ПРИМЕРЫ:

```
squ /ii/bin/*.cod bin -q
```

Создать файл-пакет -bin.squ-, состоящий из всех кодофайлов директории /ii/bin.

```
squ bin
```

Распаковать пакет -bin.squ-. При получении подтверждения распаковываются соответствующие файлы.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

stty

```
stty [ttyno] [ t=<terminal type> ] [-ключи]
sET tERMINAL tYPE
t=1 - фрящичек
t=2 - LABTAM 3000 v85.1
.....
```

...Чтоб свистнуть телевизор, надо
быть немножко художником. Вещь хрупкая,
громоздкая, задел экраном за дверную ручку
- и привет.

Ю.Коваль. Пять похищенных монахов

Устанавливает тип терминала, программирует клавиатуру, используя файл 'kb(N).keys', где N - тип (!) терминала. Такой файл получается с помощью утилиты -mkboard-. Таким образом, вы можете установить функциональные клавиши на вашем терминале на удобные для вас кнопки. Файл 'kb(N).keys' находится по установленному пути.

КЛЮЧИ:

-h - подсказка;
-k - перепрограммировать клавиатуру с помощью файла

'kb(N).keys', где N - ТИП терминала;
-v - показать функциональные клавиши, установленные для терминала.

ПРИМЕРЫ:

stty 3 t=2
Сообщить системе, что у 3-го терминала тип 2 (Labtam 3000).

stty 3 t=2 -k

Перепрограммировать 3-ий терминал с помощью файла 'kb(2).keys'.

.PAGE

.HEADER . УСТРОЙСТВА

try

try <устройство> <число попыток>
try TO DO

Устанавливает в системе число повторных попыток при ошибке на устройстве.

КЛЮЧИ:

Ключей нет.

ПРИМЕРЫ:

try dx 30

При ошибке чтения/записи дискеты будет совершено 30 попыток прочитать/записать.

.PAGE

.HEADER . КОНФИГУРИРОВАНИЕ СИСТЕМЫ

up

up
STARTup

Утилита -up- предназначена для настройки системы на конкретное оборудование (загрузка драйверов), а также для загрузки и запуска различных утилит и командных файлов (например, запуск утилиты sh - shell). -up- умеет только загружать и запускать программы и выполнять командные файлы. После того, как система инициализирована, пользоваться -up- нельзя.

Если установлен режим автозапуска системы (autorun), то утилита -up- выполняет только запуск файла конфигурации системы (/etc/config).

КЛЮЧИ:

Ключей нет. Утилита имеет команды:

@ <имя_файла>	- исполняет командный файл 'имя_файла';
\$\$ [префикс]	- устанавливает префикс к именам утилит (например, путь до директории, где находятся утилиты, от корневой директории). Если [префикс] опущен, на экран выдается ранее установленный префикс.
<имя_файла> <параметры>	- выполняется утилита '/\$\$/имя_файла', где '\$\$' - ранее установленный префикс;
/полное_имя_файла <параметры>	- исполнение утилиты 'полное_имя_файла';
stk [целое число]	- устанавливает размер памяти в словах для утилиты, которая будет запущена после этой команды;
mem	- выдает на экран карту оперативной памяти;
\$буква=текст	- установка макроса: везде далее '\$буква' будет пониматься как 'текст';
\$v	- везде заменяется на имя стартового носителя системы;
quit	- переход в пульт (FHO - for hackers only);
bye	- окончание работы;
%	- комментарии в командном файле;
? команда	- появляется запрос на подтверждение команды; ответы: CR, LF, "y", "д" - исполнять; SPACE, "n", "н" - не исполнять;
@ /\$v/etc/config	- запуск командного файла для конфигурации системы по умолчанию;

Начально установлено: $\$\$=/\$v/bin$, $\$b=/\v/bin , $\$e=/\v/etc .

.PAGE

.HEADER . ВИЗУАЛИЗАЦИЯ

v2

v2 {<имя_файла>} [-ключи]
vISUALIZER OF MODULA-2 CODEFILE

Выдает информацию о кодофайле программы. При запуске утилиты без указания имени файла происходит вход в монитор v2; появляется приглашение 'Vis>', после чего можно пользоваться командами

- m - появляется приглашение 'Имя модуля', после чего необходимо указать имя файла, о котором требуется информация;
- q - окончание работы с монитором.

В мониторе работают как команды все ключи v2.

КЛЮЧИ:

- t - показывает процедурную таблицу: шестнадцатеричные номера процедур и соответствующие имена;
- v - выдает шестнадцатеричные номера переменных модуля и их имена;
- i - выдает общую информацию о модуле: версия, размер кодофайла, количество процедур и переменных и т.д.;
- e - выдает список импортируемых модулей;
- s - выдает строковый пул модуля;
- h,? - подсказка;
- c - протокол записывается в файл с расширением .vs;
- x - отключается выдача протокола на терминал;
- p= - тип процессора.

.PAGE

.HEADER . УСТРОЙСТВА

volc

volc <устр_источник> <устр_приемник> [-ключи]
vOLUME COPY

Копирование носителей 'один в один'. Используется при создания загрузочных (системных) носителей.

КЛЮЧИ:

- h - подсказка;
- s=<целое> (Source) - задает начальный блок источника;
- d=<целое> (Destinator) - задает начальный блок приемника;
- l=<целое> (Length) - задает число копируемых блоков.

ПРИМЕРЫ:

```
volc ld7 dx0 l=41
```

Копируется 41 блок с логического устройства ld7 на устройство dx0.

.PAGE

.HEADER . РАБОТА С ФАЙЛАМИ

where

```
where <суперобразец>} [-h]
```

```
where FILE?
```

Ищет файл -образец- на указанной директории. Будет искать и на дочерних директориях, если перед именем файла стоит '//'. В случае нахождения такого файла выдает его полное имя.

КЛЮЧИ:

-h - подсказка;
-d - показать дату последней модификации;
-t - показать время последней модификации;
a=DDMMYY - искать только файлы, созданные после DD дня
MM месяца YY года (без разделителей).

ПРИМЕРЫ:

```
where /ii/doc/util/file/* a=151287 -d
```

на директории /ii/doc/util/file ищет все файлы, созданные после 15.12.87, с показом даты создания.

```
where /ii/doc/util/file/*.doc
```

на директории /ii/doc/util/file ищет все файлы с расширителем -doc- .

```
where /ii/doc/util//ls.doc
```

ищет файл -ls.doc- на директории /ii/doc/util и по всем директориям, лежащим на директории -util-. Будут найдены, например, такие файлы:

```
/ii/doc/util/ls.doc
```

```
/ii/doc/util/file/ls.doc.
```