

Задание на ОС	Разработка ведется в Linux					Разработка ведется в Windows		
	FASM, gcc	GNU as (Intel), gcc	GNU as (AT&T), gcc	YASM (Intel), gcc	YASM (AT&T), gcc	FASM, ms c compiler	YASM (Intel), ms c compiler	YASM (AT&T), ms c compiler
DictOS (Испанский)		1		2			17	
DictOS (Финский)			3			18		19
CalcOS	4	5			7		20	
SolverOS				6	8	21	22	
StringsOS	9	10	11					23
InfoOS	13	12			16		24	
ConvertOS			14	15		25		26

Общие требования к загрузчику:

1. Должен загружать ядро со второго гибкого диска по нужному адресу (0x10000 для плоского ядра, 0x11000+0x12000 для ядра PE).
2. Запрашивать у пользователя (либо BIOS) необходимые параметры и сохранять их в память (по произвольному адресу, определяется студентом).
3. Переводить процессор в защищенный режим и передавать управление ядру.

Общие требования к ядру ОС:

1. ОС должна считать параметры, сохраненные загрузчиком (в п.2), и применить их.
2. Настроить прерывания и включить их.
3. Перейти в интерактивный режим: на экране должна быть строка приветствия. Пользователь вводит в нее команду, а ОС отвечает на нее. Команда состоит из ключевого слова и параметров (для некоторых команд) — указываются через пробел. Перечень поддерживаемых команд указан в варианте задания. Если команда не распознана — должны выводиться сведения об этом (ошибка).
4. ОС не позволяет вводить команды длиннее 40 символов. При попытке ввести команду длиннее лимита — ОС не будет реагировать на лишние нажатия. Должна поддерживаться клавиша Backspace.
5. ОС поддерживает ввод строчных букв латинского алфавита, пробела, цифр, знаков мат. операций + - / *
**** Доп. задание:** ОС поддерживает ввод заглавных букв латинского алфавита.
6. Если все строки текущего экрана заполнены — экран очищается и взаимодействие с пользователем продолжается вновь, на чистом экране.
**** Доп. задание:** Если все строки экрана заполнены, то первая строка исчезает, ее место занимает вторая, место второй — третья и т.д., последняя строка дополняется пустой строкой. Таким образом, экран автоматически прокручивается вверх по мере заполнения, как в консоли Unix.

Пример взаимодействия пользователя и ОС:

Welcome to CalcOS!

info

Calc OS: v.01. Developer: Ivanov Ivan, 23508/1, SpbPU, 2017

Compilers: bootloader: fasm, kernel: gcc

Bootloader parameters: green color.

dictinfo

Error: command not recognized

expr 5+2*3

Result is: 11

expr 5++

Invalid expression

shutdown

Powering off...

При реализации задания следует учесть следующее:

1. Для считывания данных клавиатуры в загрузчике следует применять функцию BIOS int 0x16.
2. При разработке ядра ОС на языке C/C++ недоступны никакие функции стандартной библиотеки Си, в том числе управления памятью, форматированного вывода, работы со строками, обнуления памяти и т.д. Любые нужные библиотечные функции придется реализовать самостоятельно. Допускается использовать какие-либо реализации из GNU-библиотек или ОС Pintos.
3. Не обеспечивается поддержка вещественных чисел. Для получения каких-либо значащих цифр после запятой следует применять только целочисленную арифметику (умножение, деление и деление с остатком).
4. Функции **shutdown** реализуются с помощью записи последовательности байтов в порты ввода-вывода. Номера портов и байтов можно посмотреть в файле `devices/shutdown.c` ОС Pintos: `outw(0x604, 0x2000)`. Реализация функции `outw` выполняется по аналогии с функцией `outb`.
5. Для реализации функции **ticks** следует добавить обработчик прерывания системного таймера 0x08 (подобно обработчику прерываний клавиатуры 0x09), а также разрешить генерацию этого прерывания в маске прерываний контроллера. За разрешение прерываний от системного таймера отвечает бит 0x01 в маске (для клавиатуры — бит 0x02 в маске).

CalcOS

Реализация простого вычислителя выражений.

Функция загрузчика — запрос у пользователя цвета букв консоли, который будет использоваться после загрузки ядра ОС. Должны поддерживаться цвета: green, blue, red, yellow, gray, white. Загрузчик устанавливает каждому цвету в соответствие цифру и распечатывает эту информацию, например: "Press: 1-green,2-blue...". . Пользователь нажимает цифровую клавишу и загрузчик немедленно, без каких либо подтверждений от пользователя, переходит к загрузке ОС.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор) и цифровой кнопке, указанной пользователем при загрузке.

expr выражение

Вычисляет значение выражение с учетом приоритета операций и выводит его на экран.

Выражение содержит числа и знаки +-*/. Должны поддерживаться унарные операции + и -. Деление является целочисленным (не вещественным).

В случае указания некорректного выражения выводится ошибка. В случае попытки делить на 0 — выводится ошибка. В случае возникновения переполнения целочисленного значения int — выводится ошибка. Перед числом может быть несколько унарных минусов.

****Доп. задание:** команда expr поддерживает круглые скобки.

Примеры:

```
# expr ---1+5
Result: 4
# expr 1-2*5
Result: -9
# expr: 5**5
Error: expression is incorrect
# expr: 5555555555555*5555555555555
Error: integer overflow
# expr: 5+71/3/4
Result: 10
# expr: 5/0
Error: division by 0
```

```
** # expr: 10/(3+2)
Result: 2
```

shutdown

Выключение компьютера

SolverOS

Реализация математических алгоритмов.

Во время работы загрузчика пользователем указывается цвет шрифта консоли ОС. Загрузчик выводит поддерживаемые цвета в виде вертикального меню с одним выбранным пунктом. Пользователь может менять выбранный пункт меню с помощью кнопок "вверх", "вниз". Для старта ядра ОС необходимо нажать Enter. Должны поддерживаться цвета: gray, white, yellow, cian, magenta, green.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор), выбранный в загрузчике цвет (текстом, например "green").

```
gcd dig1 dig2
```

Находит НОД чисел dig1 и dig2. Если какое-либо из чисел ≤ 0 - выводит ошибку. В случае целочисленного переполнения — выводит ошибку. В случае некорректной команды — выводит ошибку. Примеры:

[illegible]

solve equation

Решает линейное уравнение вида $ax+b=c$. Если решение вещественное – выводится дробь. Дробь должна быть несокращаемая ($\text{НОД}(\text{числитель}, \text{знаменатель})=1$).

**** Доп. задание:** Если решение вещественное — выводится 5 знаков после запятой.

При неправильном вводе — выводится ошибка. а,с могут содержать унарный минус.

Перед **b** может стоять минус. Примеры:

```
# solve 5x+1=11
Result: x=2
# solve 6x+1=5
Result: x=2/3
# solve -x-1=5
Result: x=-6
** # solve 6x+1=5
** Result: x=0.66667
```

shutdown

Выключение компьютера

****Доп. задание:**

```
lcm dig1 dig2
```

Находит НОК чисел `dig1` и `dig2`. Обработка ошибок — по аналогии с `lcm`.

****Доп. задание:**

div dig1 dig2

Делит dig1 на dig2. Выводит 5 знаков после запятой. Выводит ошибку при попытке делить на 0.

DictOS

Реализация простого словаря-переводчика *с английского языка* на указанный язык.

Загрузчик: взаимодействует с пользователем, позволяя пользователю указать — слова на какие буквы английского языка его будут интересовать после запуска ОС. Загрузчик предлагает пользователю разрешить использовать слова на указанную букву или запретить использовать их путем однократного нажатия на соответствующую клавишу.

При этом на экране указывается какие буквы отмечены, а какие нет:

abcde_____kl_nop_____uv_x_z

При нажатии на клавишу, например 'w' экран обновляется:

abcde_____kl_nop_____uvw_x_z

Для старта ОС пользователь нажимает Enter.

Словарь может храниться в секции данных ядра (можно объявить его прямо в коде как статические данные). Словарь должен быть отсортирован. Допускается хранение словаря в неотсортированном виде, но при этом сортировка выполняется при запуске ОС.

Словарь должен содержать не менее 50 общеупотребительных слов. Артикли в словах указывать не требуется.

****Доп. задание:** Словарь должен содержать не менее 500 общеупотребительных пар слов.

Для поиска в словаре обязательно должен применяться *бинарный поиск*.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор), заданные в загрузчике параметры — перечень букв, слова на которые должны обрабатываться при запросе перевода.

dictinfo

выводит информацию о загруженном словаре (язык, общее кол-во слов, количество доступных для перевода слова — вычислено на основании заданных в загрузчике данных). Пример:

```
#dictinfo
```

```
Dictionary: en -> es
```

```
Number of words: 1121
```

```
Number of loaded words: 780
```

translate слово

Переводит слово с английского языка на другой. Если слово не найдено или не загружено — выводит ошибку. При поиске слова выполняется бинарный поиск в массиве слов словаря.

****Доп. задание:** При выводе перевода некоторые специфические буквы языка должны выводиться корректно (для испанского языка — буква ñ; для финского — буквы ä, ö). В словаре обязательно должны быть слова с такими буквами.

Пример:

```
#translate cat
gato
#translate airport
aeropuerto
# translate moonlight
Error: A word 'moonlight' is unknown
```

wordstat буква

Выводит количество загруженных слов в словаре на указанную букву.

```
#wordstat z
Letter 'z': 57 words loaded.
#wordstat y
Letter 'z': 1 word loaded.
#wordstat d
Letter 'd': 0 words loaded.
```

shutdown

Выключение компьютера.

****Доп задание:**

anyword буква

Выводит случайное слово на указанную букву и его перевод. Буква может быть не указана – тогда выводится случайное слово на любую букву. Алгоритм генерации случайных или псевдослучайных чисел определяется студентом и должен быть описан в отчете.

Примеры:

```
#anyword n
night: noche
#anyword z
Error: no words
#anyword n
nice: bonito
# anyword
boy: muchacho
```

StringsOS

ОС, демонстрирующая возможности работы со строками.

Во время работы загрузчика пользователь должен ввести с клавиатуры "bm" или "std". Под вводом понимается последовательное нажатие двух клавиш: b, потом m; либо трех клавиш — s, t, d. Другие комбинации — не принимаются. Пользователь может ввести aabstbm — такая последовательность будет распознана как "bm" (по первой корректной комбинации). Ввод не должен подтверждаться клавишей "Enter" — как только загрузчик распознает "bm" или "std" — он должен немедленно перейти к загрузке ОС. Указанный пользователем выбор используется в дальнейшем при работе ядра ОС — для определения алгоритма поиска подстроки в строке.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор), параметры загрузчика — указанный алгоритм — bm или std.

upcase строка

Распечатывает указанную строку в верхнем регистре. Пример:

```
# upcase hello
HELLO
# upcase abc def
ABC DEF
```

downcase строка

Распечатывает указанную строку в нижнем регистре. Пример:

```
# upcase hElLo WoRlD
hellow world
```

titlize строка

Распечатывает введенную строку, первую букву в каждом слове выводит как заглавную.

Разделителями слов являются пробелы. Примеры:

```
# titlize hello
Hello
# titlize hello world
Hello World
# titlize abc def
Abc Def
```

template шаблон

Загрузить шаблон для поиска подстрок в память. Если используется алгоритм "bm"

(Бойера-Мура или Бойера-Мура-Хорспула — на выбор студента), то распечатывается таблица сдвигов. Пример для алгоритма "std":

```
# template hel
Template 'hel' loaded.
```

Пример для алгоритма "bm":

```
#template string
Template 'string' loaded. BM info:
s:5 t:4 r:3 i:2 n:1 g:5
```

search строка

Найти в указанной строке первое вхождение шаблона и вывести индекс на экран (индексация — от 0). Пример для загруженного шаблона 'ab':

```
#search ccabcc
Found 'ab' at pos: 2
#search ccbcc
Not found 'ab'
#search ababab
Found 'ab' at pos: 0
```

shutdown

Выключение компьютера

****Доп. задание**

tracesearch строка

Выполняет поиск первого вхождения подстроки, как и search, но при этом на каждом шаге распечатывает отладочную информацию. Следует указать сколько совпадающих символов обнаружено для очередной итерации, а также на сколько символов вперед будет произведен сдвиг.

Пример для 'std':

```
#tracesearch aabbabaababbbb
Searching 'abab' in 'aabbabaababbbb':
Try at pos 0: 1 let. equal, go to pos:1.
Try at pos 1: 1 let. equal, go to pos:2.
Try at pos 2: 0 let. equal, go to pos:3.
Try at pos 3: 0 let. equal, go to pos:4.
Try at pos 4: 3 let. equal, go to pos:5.
... (пропущено)
Try at pos 7: 4 let. equal, found. Result is 7.
```

Пример для bm:

```
template abab
a:1 b:2
tracesearch aabbabaababbbb
Searching 'abab' in 'aabbabaababbbb':
Try at pos 0: 1 let. equal, shift[b]=2, go to pos:2.
Try at pos 2: 3 let. equal, shift[b]=2, go to pos:4.
Try at pos 4: 0 let. equal, shift[a]=1, go to pos:5.
Try at pos 5: 2 let. equal, shift[a]=1, go to pos:6.
Try at pos 6: 0 let. equal, shift[a]=1, go to pos:7.
Try at pos 7: 4 let. equal, found. Result is 7.
```


****Доп. задание**

passgen длина опции

Генерирует случайный пароль указанной длины и распечатывает его на экран. Опции — это одна или несколько букв — указывает допустимые символы в пароле: 'a' — маленькие буквы, 'A' — большие буквы, '0' — цифры, '+' — спец. символы. Если опции не указаны, то это равносильно указанию 'aA0+'. В пароле должен присутствовать как минимум 1 символ из указанных множеств. Если это невозможно — пароль не генерируется.

Алгоритм псевдослучайных или случайных чисел, а также начальное значение seed для него определяется студентом. Примеры:

```
# passgen 5 Aa0
Password: aZ5b0
# passgen 7 Aa0+
Password: $vP78b#
# passgen 3 +Aa0
Error: too short for all symbols
# passgen 4 a
Password: fdid
# passgen 6 aA
Password: uYKJMAp
```

InfoOS

Служебная ОС для сбора и вывода системной информации.

Загрузчик должен собрать необходимую информацию с помощью функций BIOS и сохранить ее в памяти, доступной в дальнейшем ядру. Должно быть собрано:

1. Текущее время. Для получения значения текущего времени в загрузчике следует применять функцию BIOS int 0x1A.
2. ****Доп. задание:** Размер памяти компьютера. Для получения размера доступной памяти следует использовать функцию int 0x15, ax=0xe801. Размер будет возвращен в в регистре cx, также dx (в 64кб-блоках в регистре).
3. ****Доп. задание:** Загрузчик записывает в COM1-порт строку 'Loading OS...' перед загрузкой ядра ОС с диска в память. Вывод в COM можно выполнять с помощью BIOS-функции int 0x14.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор)

help

Выводит перечень поддерживаемых команд

clear

Очистить все введенные пользователем команды и выведенные ОС результаты их выполнения.

ticks

Выводит количество тиков, прошедших с момента старта системы. При каждом вызове распечатываемое значение должно быть больше, чем при предыдущем. Для подсчета тиков в ядре необходимо реализовать обработчик прерывания, подобно обработчику прерываний клавиатуры.

loadtime

Время и дата, считанные с помощью BIOS на момент загрузки ядра ОС.

```
# loadtime
```

```
OS loaded at: 12:43:51
```

curtime

Выводит текущее время. Текущее время можно считать в ядре из порта ввода-вывода 0x71 с помощью inb, предварительно записав в порт 0x70 (outb) интересующий регистр: 0 – секунды, 2 – минуты, 4 – часы. Значение возвращается в формате BCD, т.е. 0x28 для минут – это 28 минут, а не 40.

```
# curtime
```

```
Current time is: 12:45:59
```

uptime

Выводит общее время работы системы в читаемом формате (сколько система работает часов, минут, секунд). Должна быть реализована поддержка единственного и множественного числа.

```
# uptime
Uptime is 1 minute and 15 seconds
# uptime
Uptime is 3 minutes and 1 second
```

****Доп. задание:**

meminfo

Выводит информацию о доступной памяти, в kb, полученной от BIOS.

```
# meminfo
Available memory: 14640kb
```

cpuid

Определяет модель процессора (*Vendor ID*) с помощью инструкции `cpuid` (в ядре) и распечатывает полученную от процессора строку.

```
# cpuid
VendorId: GenuineIntel
```

shutdown

Выключение компьютера

ConvertOS

ОС для выполнения простых операций по конвертации величин:

Задание к загрузчику: указывается цвет шрифта консоли ОС (пользователь выбирает из списка предлагаемых с помощью кнопок "вверх", "вниз" и подтверждает выбор клавишей Enter). Должны поддерживаться цвета: *gray, white, yellow, blue, red, green*. Выбор осуществляется только с помощью кнопок курсора "вверх" и "вниз"! На экран выводится только текущий выбранный цвет. При нажатии кнопки — надпись меняется. После загрузки ОС всех буквы в консоли будут иметь выбранный в загрузчике цвет.

Поддерживаемые ОС команды:

info

Выводит информацию об авторе и средствах разработки ОС (транслятор ассемблера, компилятор), параметры загрузчика — выбранный пользователем цвет шрифта консоли ОС.

clear

Очистить все введенные пользователем команды и выведенные ОС результаты их выполнения.

shutdown

Выключение компьютера

nsconv число исх_система конечн_система

Переводит указанное в параметре число из исходной системы счисления в конечную систему. В случае целочисленного переполнения должна выводиться ошибка. В случае ошибочного знака (некорректный символ для данной СС) выводится ошибка. Должны поддерживаются системы счисления от 2 до 36. Буквенные символы принимаются в нижнем регистре.

Примеры:

```
# nsconv ff 16 10
255
# nsconv 78 8 10
error: invalid symbol: 8
# nsconv 999999999999999999 10 16
error: int overflow
```

posixtime число

Распечатывает дату и время заданные указанным числом, являющимся меткой времени POSIX (время, прошедшее в секундах, начиная с 01.01.1970 00:00:00 (UTC))

```
# posixtime 1111111111
18.03.2005 01:58:31
```

```
# posixtime 1500000000
14.07.2017 02:40:00
```

wintime число

Распечатывает дату и время заданные указанным 64-битным числом, являющимся меткой времени Windows (время, прошедшее в 100-наносекундных интервалах, начиная с 01.01.1601 00:00:00 UTC)

```
# wintime 123456789012345678
21.03.1992 19:15:01
```

**** Доп. задание:**

SZconv число исх_единицы конечн_единицы

Переводит указанное число — размер информации из исходных единиц (например, байт, мегабайт) в другие единицы (например, гигабайты, терабайты). Единицы измерения задаются буквами: b, kb, mb, gb, tb. В случае целочисленного переполнения выводится ошибка. Дробный результат выводится с двумя знаками после запятой. В случае некорректного ввода выводится ошибка.

Примеры:

```
# szconv 1056 kb mb
```

1.03 mb

```
# szconv aa kb b
```

```
error: invalid size: aa
```

```
# szconv 99999999999999999999 tb b
```

```
error: int overflow
```