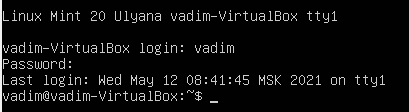
## Утилиты обработки текста и текстовые редакторы

### Цель работы

Ознакомиться со стандартными средствами UNIX-подобных операционных систем, предназначенными для обработки текстовой информации.

### Задания к работе

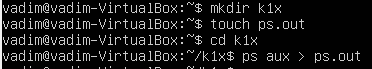
1. Войти в систему с собственной учетной записью.



1. Вывести расширенный список процессов (ps aux) и сохранить эту информацию в файл ps.out

Создадим отдельный каталог k1x:

/home/vadim/k1x



1. Выполнить сортировку файла ps.out по имени пользователя-владельца процесса, сохранить результат в файл sorted.ps



Где команда sed ‘1d’ удаляет первую строку файла ps.out и выводит поток новых данный уже с удаленной первой строкой;

> перенаправляет поток вывода в файл ps.out.temp, так как если мы осуществим перенаправление потока в файл ps.out в виду особенностей работы команды это файл окажется пустым



Переместим файл ps.out.temp с именем ps.out, для того, чтобы изменить исходный файл, удалив в нем первую строку

Затем отсортируем файл:

Так как при выполнении команды ps aux имя пользователя уже указано (1-ая колонка), команда сортировки будет выглядеть так:



Где команда sort сортирует файл,

-k указывает поле по которому нужно сортировать строки, цифра 1 обозначает номер поля

ps.out – это сортируемый файл

> - перенаправляет поток вывода в файл sorted.ps

1. Разделить сортированную информацию из файла sorted.ps по нескольким файлам, в каждом файле - один ключ сортировки (например, файл root.ps - содержит строки, начинающиеся со слова root, user.ps - со слова user и т.д.)



Листинг цикла:

while read line; do echo $line >> $(echo $line | awk ‘{print$1}’).ps; done < sorted.ps

Где while read - заголовок цикла, line переменная в которую будет передаваться по строке в процессе прохода цикла

; - разделитель

do это оператор цикла после которого идут команды

echo $line выводи содержимое переменной окружения $line

>> дозаписывает выведенный предыдущей командой поток в следующий файл с именем

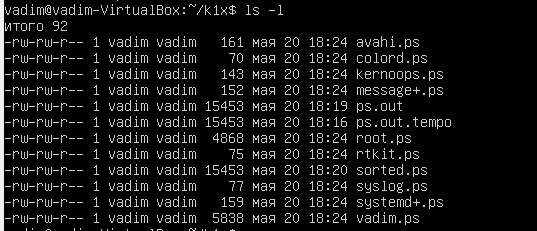
$(echo $line | awk ‘{print$1}’).ps

echo $line выводит содержимое строки, в которой первое слово (имя пользователя) мы выбираем при помощи awk ‘{print$1}’. К полученному имени пользователя добавляем .ps, так как по заданию файл должен иметь вид username.ps

done метка завершения цикла

< sorted.ps указывает передаваемый в цикл файл, который обрабатывается построчно

Результат выполнения цикла:

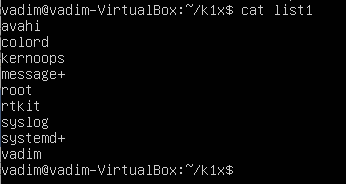


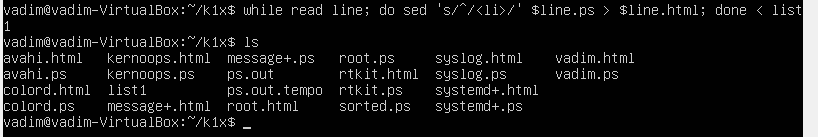
1. Добавить в начало каждой строки созданных файлов текст <li> (тег разметки списков языка HTML), результаты сохранить в файлах исходное\_имя.html (например).



Получим первое слово каждой строки файла sorted.ps (имена пользователей) и оставляем только уникальные имена командой uniq (записывает имена без повторений). Делаем это для того, чтобы имена файлов находились в одном файле, а далее сможем отсортировать эти файлы.

> list1 записывает результат в файл list1





Листинг цикла:

while read line; do sed ‘s/^/<li>/’ $line.ps > $line.html; done < list1

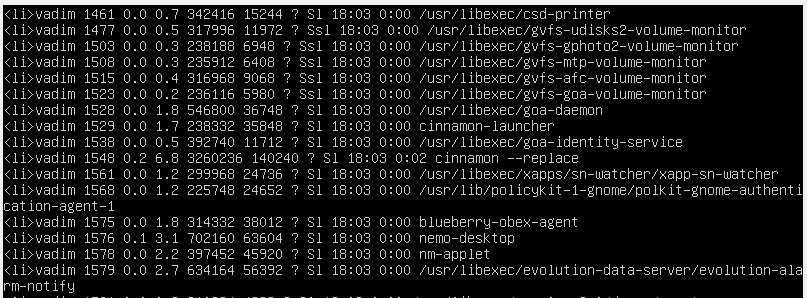
Далее передаем файл list1 в цикл и построчно (то есть для каждого имени пользователя) добавляем в начало каждой строки файла тег <li> командой sed ‘s/^/<li>/’ $line.ps

Где s –заменить ^-начало строки на <li> в файле $line –имя и .sh, то есть в файлах, созданных в предыдущем примере, а результат

> $line.html записать в файл с таким же именем + .html

Результат выполнения цикла:





1. Подсчитать количество строк в каждом из полученных файлов (это значение будет соответствовать числу процессов, запущенных пользователем). Результаты объединить со строкой вида:

<p><b>Итого процессов: NN</b>

где NN - количество процессов, запущенных пользователем и сохранить в отдельные файлы (например root.total, user.total и т.п).

Листинг цикла:

while read line; do count=$(wc $line.html |awk ‘{print$1}’; echo ‘<p><b>Total process: ‘$count’</b>’ > $line.total; done < list1

=$(wc $line.ps.html |awk ‘{print$1}’

Утилитой wc получаем информацию о количестве строк, слов, байт в файле и утилитой awk берем только первое слово – количество строк

И присваиваем это количество строк переменной count=

while read line; do; done < list1

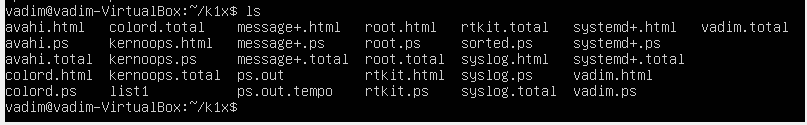
Все это происходит в цикле while куда передался созданный ранее файл list1 с именами пользователей, а в переменной line хранится имя пользователя

echo ‘<p><b>Total process: ‘$count’</b>’

Далее вызываем через echo строку указанного в задании вида, подставляя туда количество процессов из переменной count и записываем (> $line.total) в файл вида username.total

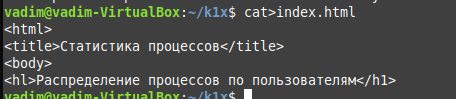
Результат:





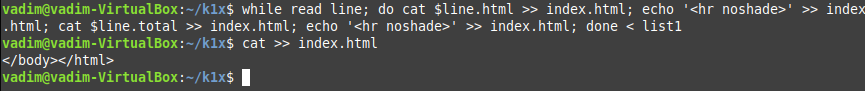
1. Создать файл index.html следующего содержания:

<html>  
<title>Статистика процессов</title>  
<body>  
<h1>Распределение процессов по пользователям</h1>



1. Командами обработки текста последовательно добавить в файл index.html ранее созданные файлы, разделенные строкой "<hr noshade>" (например в таком порядке: файл root.html, строка-разделитель, файл root.totlal, user.html, строка-разделитель, user.total и т.д.). Завершить формирование файла добавлением строки вида:

</body></html>



Листинг цикла:

while read line; do cat $line.html >> index.html; echo ‘<hr noshade>’ >> index.html’ cat $line.total >> index.html; echo ‘<hr noshade>’ >> inde.html; done < list1

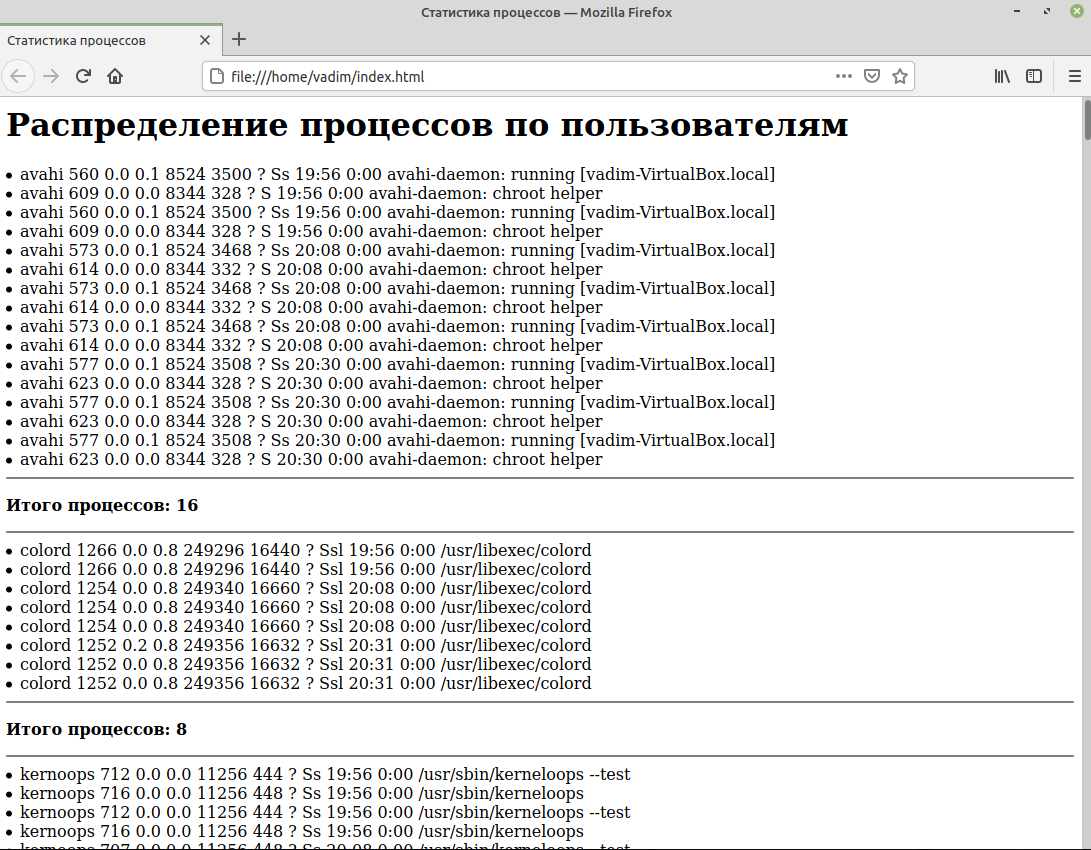
Где цикл while read line; do ;done < list1 также позволяет выполнить действия для каждого имени пользователя как и в предыдущих случаях

Содержимое файла username.html дозаписывается в index.html и записывается разделитель "<hr noshade>". То же самое username.total и разделитель дозаписывается в файл index.html.

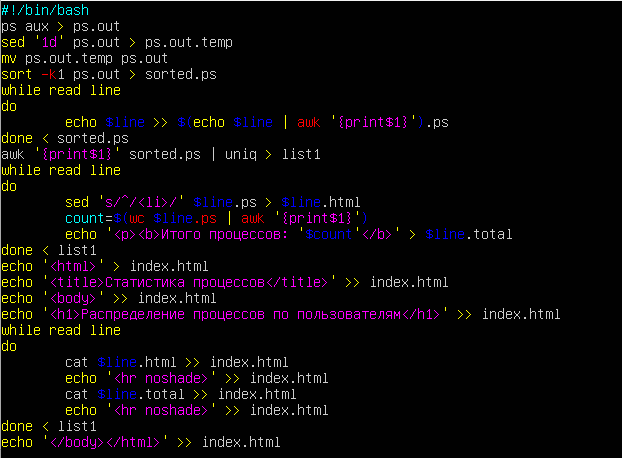
Завершаем создание файла добавлением строки в конец файла:

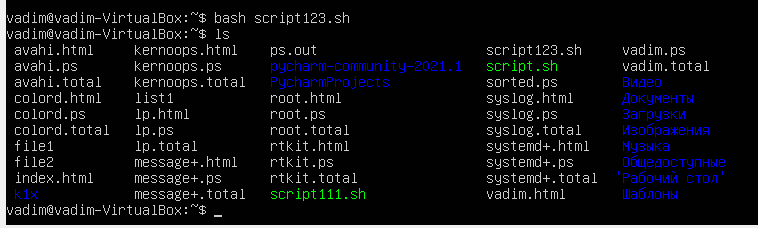


1. Просмотреть полученный файл (index.html) в веб-браузере.



1. Создать скрипт, который автоматизирует проделанные операции.





Листинг скрипта:

#!/bin/bash

ps aux > ps.out

sed '1d' ps.out > ps.out.temp

mv ps.out.temp ps.out

sort -k1 ps.out > sorted.ps

while read line

do

echo $line >> $(echo $line | awk '{print$1}').ps

done < sorted.ps

awk '{print$1}' sorted.ps | uniq > list1

while read line

do

sed 's/^/<li>/' $line.ps > $line.html

count=$(wc $line.ps | awk '{print$1}')

echo '<p><b>Итого процессов: '$count'</b>' > $line.total

done < list1

echo '<html>' > index.html

echo '<title>Статистика процессов</title>' >> index.html

echo '<body>' >> index.html

echo '<h1>Распределение процессов по пользователям</h1>' >> index.html

while read line

do

cat $line.html >> index.html

echo '<hr noshade>' >> index.html

cat $line.total >> index.html

echo '<hr noshade>' >> index.html

done < list1

echo '</body></html>' >> index.html