ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №6

Дисциплина: Операционные системы

Обрезкова Анастасия Владимировна

Содержание

# 1 Цель работы

Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных. Приобретение практических навыков: по управлению процессами (и заданиями), по проверке использования диска и обслуживанию файловых систем.

# 2 Задание

1. Ознакомиться с инструментами поиска файлов и фильтрации текстовых данных.
2. Приобрсти практических навыков: по управлению процессами

# 3 Теоретическое введение

## 3.1 Перенаправление ввода-вывода

В системе по умолчанию открыто три специальных потока:

– stdin — стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0;

– stdout — стандартный поток вывода (по умолчанию: консоль), файловый дескриптор 1;

– stderr — стандартный поток вывод сообщений об ошибках (по умолчанию: консоль), файловый дескриптор 2.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout. Например, команда ls выводит в стандартный поток вывода (консоль) список файлов в текущей директории. Потоки вывода и ввода можно перенаправлять на другие файлы или устройства. Проще всего это делается с помощью символов >, >>, <, <<.

## 3.2 Конвейер

Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Конвейеры можно группировать в цепочки и выводить с помощью перенаправления в файл.

Чаще всего скрипты на Bash используются в качестве автоматизации каких-то рутинных операций в консоли, отсюда иногда возникает необходимость в обработке stdout одной команды и передача на stdin другой команде, при этом результат выполнения команды должен обработан.

## 3.3 Поиск файла

Команда find используется для поиска и отображения на экран имён файлов, соответствующих заданной строке символов.

# 4 Выполнение лабораторной работы

1. Осуществила вход в систему, используя свое имя пользователя.
2. Записала в файл file.txt названия файлов, содержащихся в каталоге /etc. Дописала в этот же файл названия файлов, содержащихся в моем домашнем каталоге. (рис. [[1](#fig:001)])

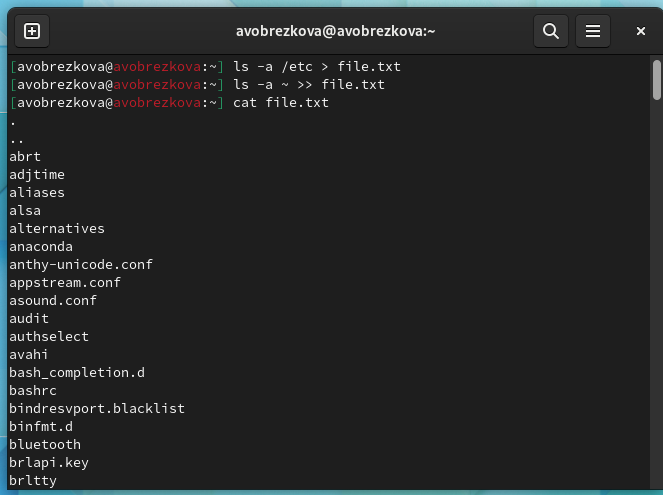


Figure 1: Запись файлов

1. Вывела имена всех файлов из file.txt, имеющих расширение .conf, после чего записала их в новый текстовой файл conf.txt. (рис. [[2](#fig:002)])

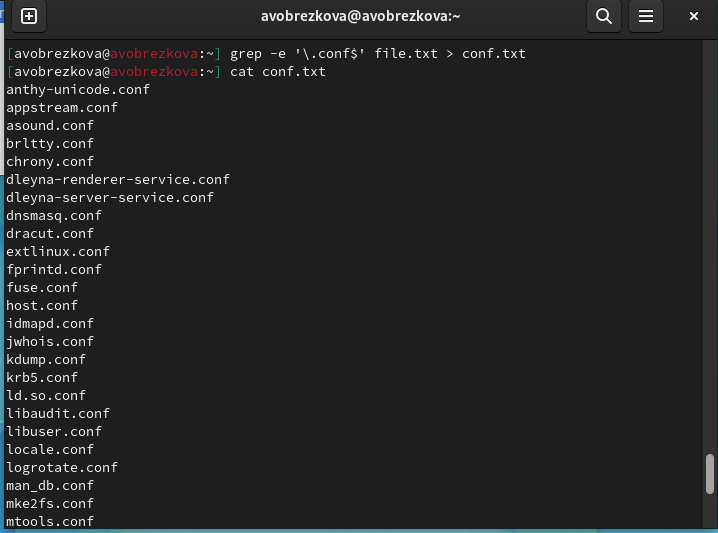


Figure 2: Вывод файлов

1. Определила, какие файлы в моем домашнем каталоге имеют имена, начинавшиеся с символа c. Найти эти файлы можно с помошью команд, указанных на рисунке. (рис. [[3](#fig:003)])

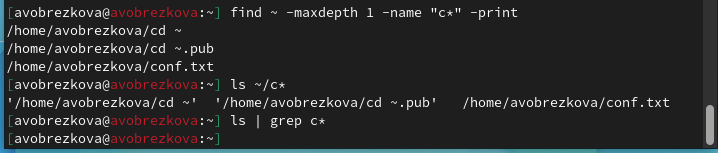


Figure 3: Нахождение файлов по символу

1. Вывела на экран (по странично) имена файлов из каталога /etc, начинающиеся с символа h. (рис. [[4](#fig:004)], рис. [[5](#fig:005)])

Figure 4: Команда для вывода файлов

Figure 4: Команда для вывода файлов

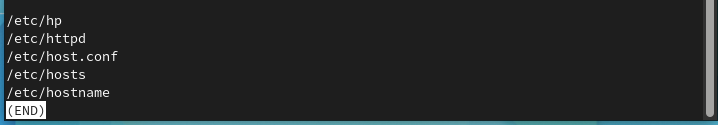


Figure 5: Найденные файлы

1. Запустила в фоновом режиме процесс, который будет записывать в файл ~/logfile файлы, имена которых начинаются с log с помошью команды find / -name “log” > ~/logfile. (рис. [[6](#fig:006)])

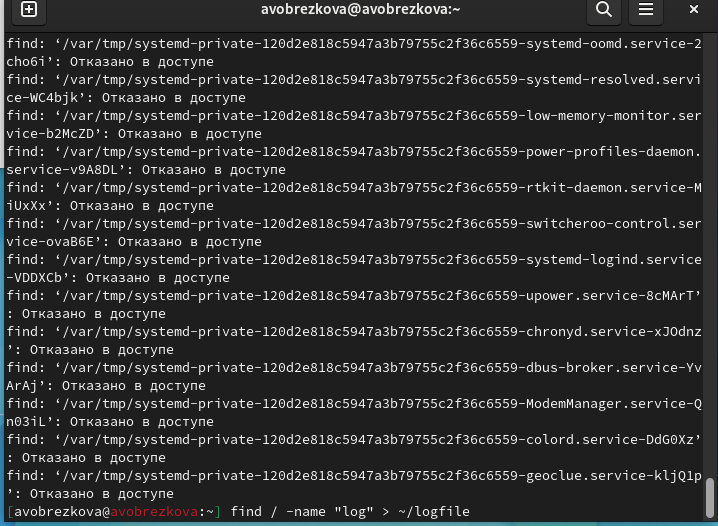


Figure 6: Нахождение файлов по символам

1. Удалила файл ~/logfile. (рис. [[7](#fig:007)])

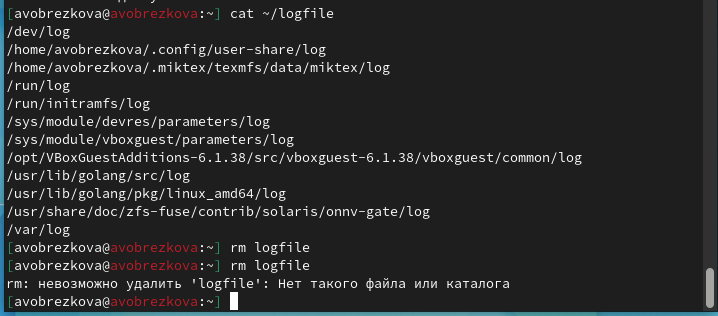


Figure 7: Удаление

1. Запустила из консоли в фоновом режиме редактор gedit. после ввода команды gedit & появляется окно редактора. (рис. [[8](#fig:008)])

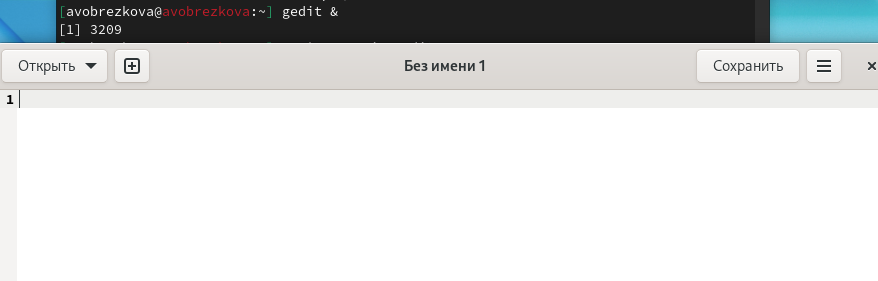


Figure 8: Редактор

1. Определила идентификатор процесса gedit, используя команду ps, конвейер и фильтр grep, с помощью команды ps | grep-i “gedit”. (рис. [[9](#fig:009)])



Figure 9: определение идентификатора процессора

1. Прочитала справку (man) команды kill, после чего использовала её для завершения процесса gedit. (рис. [[10](#fig:010)]; рис. [[11](#fig:011)])

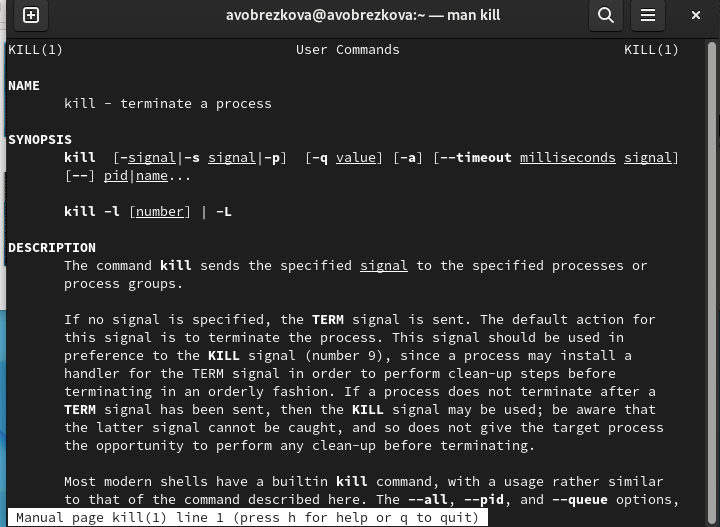


Figure 10: Справка команды kill



Figure 11: Завершение процесса

1. Выполнила команды df и du, предварительно получив более подробную информацию об этих командах, с помощью команды man. (рис. [[12](#fig:012)]; рис. [[13](#fig:013)]; рис. [[14](#fig:014)]; рис. [[15](#fig:015)])

* df - утилита, показывающая список всех файловых систем по именам устройств, сообщает их размер, занятое и свободное пространство и точки монтирования.
* du - утилита, предназначенная для вывода информации об объеме дискового пространства, занятого файлами и директориями. Она принимает путь к элементу файловой системы и выводит информацию о количестве байт дискового пространства или блоков диска, задействованных для его хранения.



Figure 12: Опции команды df

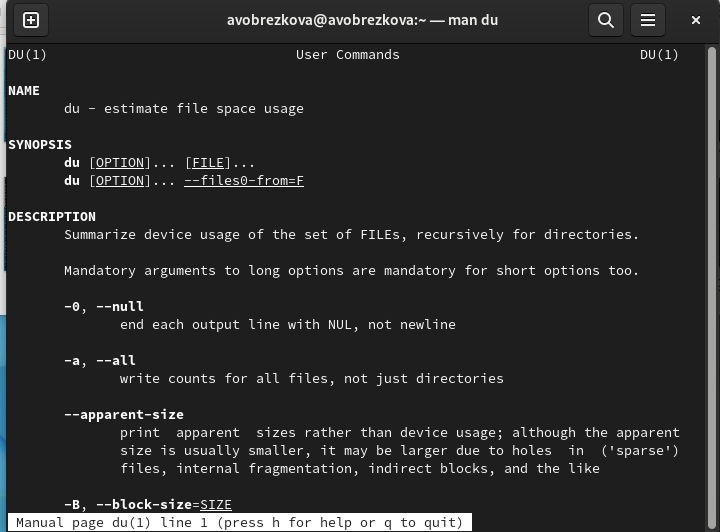


Figure 13: Опции команды du

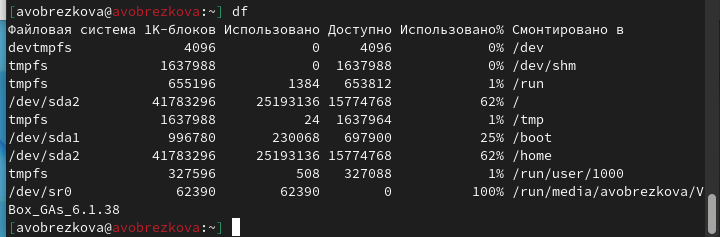


Figure 14: Командв df

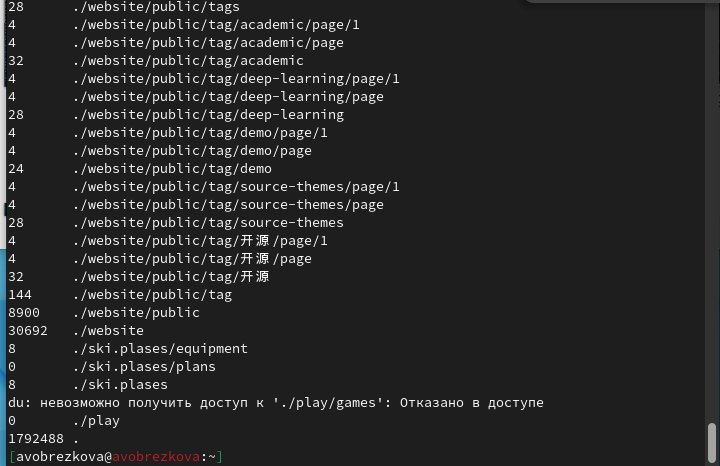


Figure 15: Команда du

1. Воспользовавшись справкой команды find, вывела имена всех директорий, имеющихся в моем домашнем каталоге. (рис. [[16](#fig:016)]; рис. [[17](#fig:017)])

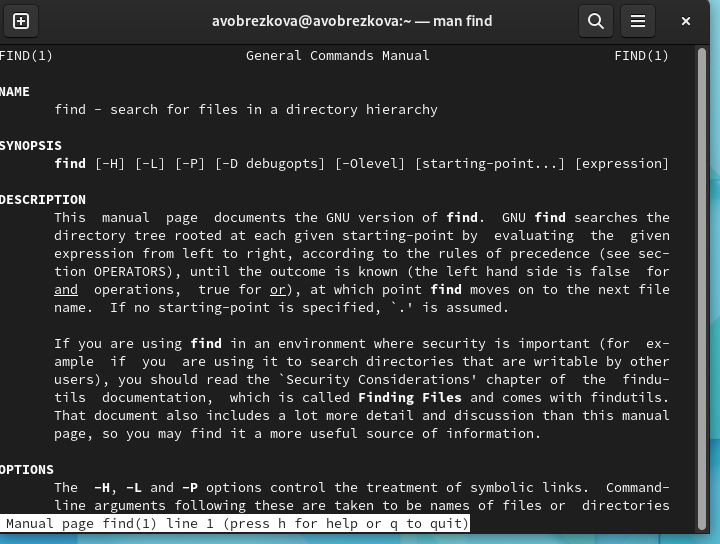


Figure 16: Опции команды find

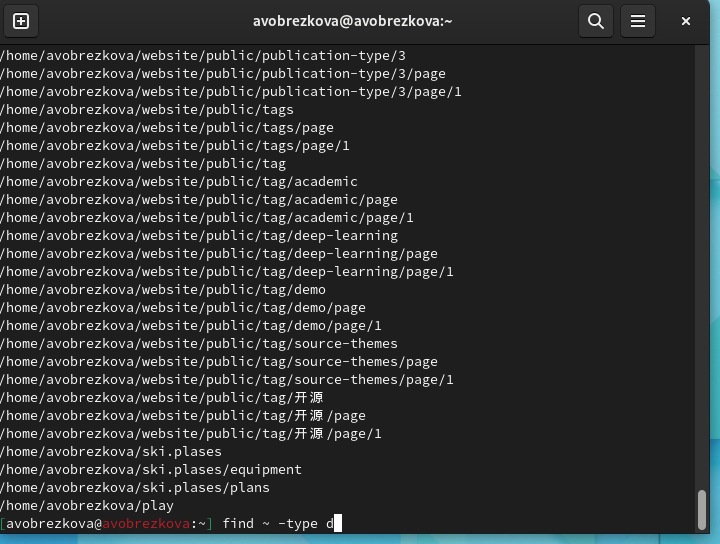


Figure 17: Выполнение команды

Данные изменения можно проверить по ссылке: <https://github.com/avobrezkova/study_2022-2023_os-intro/tree/master/labs/lab06/report>

#Контрольные вопросы

1. Какие потоки ввода вывода вы знаете?

В системе по умолчанию открыто три специальных потока:

* stdin − стандартный поток ввода (по умолчанию: клавиатура), файловый дескриптор 0;
* stdout − стандартный поток вывода (по умолчанию: консоль), файловый де- скриптор 1;
* stderr − стандартный поток вывод сообщений об ошибках (поумолчанию: консоль), файловый дескриптор 2.

Большинство используемых в консоли команд и программ записывают результаты своей работы в стандартный поток вывода stdout.

1. Объясните разницу между операцией > и >>.

‘>’ Перенаправление вывода в файл

‘»’ Перенаправление вывода в файл и открытие файла в режиме добавления (данные добавляются в конец файла)/

1. Что такое конвейер?

Конвейер (pipe) служит для объединения простых команд или утилит в цепочки, в которых результат работы предыдущей команды передаётся последующей.

Синтаксис следующий:

команда1|команда2 (это означает, что вывод команды 1 передастся на ввод команде 2)

1. Что такое процесс? Чем это понятие отличается от программы?

Процесс рассматривается операционной системой как заявка на потребление всех видов ресурсов, кроме одного − процессорного времени. Этот последний важнейший ресурс распределяется операционной системой между другими единицами работы − потоками, которые и получили свое название благодаря тому, что они представляют собой последовательности (потоки выполнения) команд.

Процесс − это выполнение программы. Он считается активной сущностью и реализует действия, указанные в программе.

Программа представляет собой статический набор команд, а процесс это набор ресурсов и данных, использующихся при выполнении программы.

1. Что такое PID и GID?

* pid: идентификатор процесса (PID) процесса (processID), к которому вызывают метод
* gid: идентификатор группы UNIX, в котором работает программа.

1. Что такое задачи и какая команда позволяет ими управлять?

Любую выполняющуюся в консоли команду или внешнюю программу можно запустить в фоновом режиме. Для этого следует в конце имени команды указать знак амперсанда &. Запущенные фоном программы называются задачами (jobs). Ими можно управлять с помощью команды jobs, которая выводит список запущенных в данный момент задач.

1. Найдите информацию об утилитах top и htop. Каковы их функции?

* top − это консольная программа, которая показывает список работающих процессов в системе. Программа в реальном времени отсортирует запущенные процессы по их нагрузке на процессор.
* htop − это продвинутый консольный мониторинг процессов. Утилита выводит постоянно меняющийся список системных процессов, который сортируется в зависимости от нагрузки на ЦПУ. Если делать сравнение сtop, то htop показывает абсолютно все процессы в системе, время их непрерывного использования, загрузку процессоров и расход оперативной памяти.

1. Назовите и дайте характеристику команде поиска файлов. Приведите примеры использования этой команды.

find − это команда для поиска файлов и каталогов на основе специальных условий. Ее можно использовать в различных обстоятельствах, например, для поиска файлов по разрешениям, владельцам, группам, типу, размеру и другим подобным критериям.

Команда find имеет такой синтаксис:

find[папка][параметры] критерий шаблон [действие]

Папка − каталог в котором будем искать

Параметры − дополнительные параметры, например, глубина поиска, и т д.

Критерий − по какому критерию будем искать: имя, дата создания, права, владелец и т д.

Шаблон – непосредственно значение по которому будем отбирать файлы.

Основные параметры:

* -P никогда не открывать символические ссылки
* -L - получает информацию о файлах по символическим ссылкам. Важно для дальнейшей обработки, чтобы обрабатывалась не ссылка, а сам файл.
* -maxdepth - максимальная глубина поиска по подкаталогам,для поиска только в текущем каталоге установите 1.
* -depth - искать сначала в текущем каталоге, а потом в подкаталогах
* -mount искать файлы только в этой файловой системе.
* -version - показать версию утилиты find
* -print - выводить полные имена файлов
* -typef - искать только файлы
* -typed - поиск папки в Linux

Основные критерии:

* -name - поиск файлов по имени
* -perm - поиск файлов в Linux по режиму доступа
* -user - поиск файлов по владельцу
* -group - поиск по группе
* -mtime - поиск по времени модификации файла
* -atime - поиск файлов по дате последнего чтения
* -nogroup - поиск файлов, не принадлежащих ни одной группе
* -nouser - поиск файлов без владельцев
* -newer - найти файлы новее чем указанный
* -size - поиск файлов в Linux по их размеру

Примеры:

find~ -type d поиск директорий в домашнем каталоге

find~ -type f -name ”.\*” поиск скрытых файлов в домашнем каталоге

1. Можно ли по контексту (содержанию) найти файл? Если да, то как?

Файл по его содержимому можно найти с помощью команды grep:

«grep -r”слово/выражение, которое нужно найти”».

1. Как определить объем свободной памяти на жёстком диске?

Утилита df, позволяет проанализировать свободное пространство на всех подключенных к системе разделах.

1. Как определить объем вашего домашнего каталога?

При выполнении команды du (без указания папки и опции) можно получить все файлы и папки текущей директории с их размерами. Для домашнего каталога: du ~/

1. Как удалить зависший процесс?

Основные сигналы (каждый сигнал имеет свой номер), которые используются для завершения процесса:

* SIGINT–самый безобидный сигнал завершения, означает Interrupt. Он отправляется процессу, запущенному из терминала с помощью сочетания клавиш Ctrl+C. Процесс правильно завершает все свои действия и возвращает управление;
* SIGQUIT–это еще один сигнал, который отправляется с помощью сочетания клавиш, программе, запущенной в терминале. Он сообщает ей что нужно завершиться и программа может выполнить корректное завершение или проигнорировать сигнал. В отличие от предыдущего, она генерирует дамп памяти. Сочетание клавиш Ctrl+/;
* SIGHUP–сообщает процессу, что соединение с управляющим терминалом разорвано, отправляется, в основном, системой при разрыве соединения синтернетом;
* SIGTERM–немедленно завершает процесс, но обрабатывается программой, поэтому позволяет ей завершить дочерние процессы и освободить все ресурсы;
* SIGKILL–тоже немедленно завершает процесс, но, в отличие от предыдущего варианта, он не передается самому процессу, а обрабатывается ядром. Поэтому ресурсы и дочерние процессы остаются запущенными. Также для передачи сигналов процессам в Linux используется утилита kill, её синтаксис:

kill [-сигнал] [pid\_процесса]

(PID – уникальный идентификатор процесса). Сигнал представляет собой один из выше перечисленных сигналов для завершения процесса.

Перед тем, как выполнить остановку процесса, нужно определить его PID. Для этого используют команды ps и grep. Команда ps предназначена для вывода списка активных процессов в системе и информации о них. Команда grep запускается одновременно с ps (вканале) и будет выполнять поиск по результатам команды ps.

Утилита pkill – это оболочка для kill, она ведет себя точно так же, и имеет тот же синтаксис, только в качестве идентификатора процесса ей нужно передать ег оимя.

killall работает аналогично двум предыдущим утилитам. Она тоже принимает имя процесса в качестве параметра и ищет его PID в директории /proc. Но эта утилита обнаружит все процессы с таким именем и завершит их.

# 5 Выводы

В ходе выполнения данной лабораторной работы я ознакомилась с инструментами поиска файлов и фильтрации текстовых данных, а также приобрела практические навыки по управлению процессами, по проверке использования диска и обслуживанию файловых систем.

# Список литературы

1. <https://esystem.rudn.ru/pluginfile.php/1976037/mod_resource/content/4/006-lab_proc.pdf>