UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

EVALUACIÓN	Obligatorio	**GRUPO**	TODOS	**FECHA**	17/10/2019
MATERIA	Programación para DevOps				
CARRERA	AII - ASA				
CONDICIONES	- Puntos: Máximo: 40 - Fecha máxima de entrega: **09/12/2019 antes de las 21 hs** LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP, RAR O PDF. **IMPORTANTE:** - Inscribirse - Formar grupos de hasta dos personas. - Subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO"				

Modalidad y puntaje

El obligatorio será hecho por dos estudiantes (como máximo, siendo 2 el número recomendado).

Si bien es en grupo, la defensa es individual y la calificación también. Se consultará a cada participante, **por separado, sobre cualquier aspecto de la entrega del grupo**.

El obligatorio tiene un puntaje máximo de **40** puntos (esto a su vez es el 40% del puntaje total del curso) y se divide en dos ejercicios. El primero de **20** puntos (un shellscript en bash) y el segundo de **20** puntos (un script desarrollado en Python).

Características generales

El objetivo básico de este obligatorio es hacer un script que genere un listado de los caminos absolutos, o relativos, de los archivos de un directorio pasado como parámetro, precedidos de la cantidad de caracteres del largo de cada camino (con opciones para listar todos los archivos del directorio en forma recursiva a partir de él, ordenar su salida por longitud de los caminos de mayor a menor, etc.) y hacer un script en Python que lo utilice apropiadamente para agregarle más funcionalidades y flexibilidad en la interpretación de sus parámetros.

El obligatorio consiste en la creación de un script en bash y un script en Python.

Ejercicio 1: Script. Valor 20 puntos máximo.

Realizar un shellscript que sea de la forma:

ej1_largo_caminos.sh [-r] [-f] [-v] [-l] [-n] Directorio

Por defecto, el script deberá listar todos los caminos absolutos a los archivos de un directorio pasado como parámetro, sin importar su tipo ni si son o no ocultos, precedidos de la cantidad de caracteres que tiene cada camino. La cantidad de caracteres del largo de cada camino deberá ser rellenada con ceos a la izquierda para que el valor siempre tenga 3 dígitos (se puede asumir que los caminos nunca tendrán un largo de más de 999 caracteres).

El script tendrá varios modificadores, que trabajaran en forma complementaria (sin interferirse unos a otros, ya que cada uno de ellos controlará aspectos independientes de la información a desplegarse).

En caso de recibirse el modificador ***-r***, se listaran todos los archivos contenidos a partir del directorio pasado como parámetro, buscando en forma recursiva y en las mismas condiciones (el listado será recursivo en la estructura de directorios partiendo del parámetro ***Directorio***). Si el script no recibe el modificador ***-r***, solo se tomarán en cuenta los archivos del directorio pasado como parámetro y no los contenidos en los de los subdirectorios (no será recursivo).

En caso de hacerse un listado recursivo (modificador ***-r***), los subdirectorios (dentro del directorio pasado como parámetro) donde el script no tenga permisos de acceso para realizar la búsqueda de archivos (para incluirlos en el listado), simplemente se ignorarán. Este problema de acceso no se considerará que cause un error en la ejecución global del script, por lo que el código de retorno no será distinto de cero por esta razón.

Si el script recibe el modificador ***-f***, solo listará archivos regulares.

Si el script recibe el modificador ***-v***, solo se desplegarán los archivos no ocultos (que podrían o no estar en directorios ocultos, no debiéndose descartar la búsqueda en directorios ocultos).

Si el script recibe el modificador ***-l*** (ele minúscula), los caminos a mostrarse serán relativos al directorio que se ha pasado como parámetro (no al directorio corriente de trabajo del script). La cantidad de caracteres a calcularse será la de los caminos relativos correspondientes.

En caso de recibirse el modificador ***-n***, se listara la información ordenada por el largo de los caminos, en orden decreciente (primero se listarán los caminos más largos para finalizar con los más cortos).

El script aceptará tanto un camino absoluto como relativo al directorio, aunque todos los caminos desplegados siempre serán relativos o absolutos según si recibe o no el modificador ***-l***, sin importar el tipo de camino que ha recibido. De esta manera, se podría recibir un camino absoluto al directorio pero desplegarse caminos relativos a ese mismo directorio que se ha recibido (si el usuario ha ingresado el modificador ***-l***) o viceversa.

El script deberá validar la existencia del directorio pasado como parámetro, verificando que sea un directorio y que se tengan los permisos necesarios para acceder a él. Si el directorio ingresado no existe en el sistema de archivos, el script deberá desplegar el mensaje "***El directorio < camino absoluto al directorio pasado como parámetro, aunque se ingrese un camino relativo > no existe, por favor ingrese un directorio válido.***" por su salida estándar de errores, devolviéndose un ***1*** como código de retorno del script. Si el directorio pasado como parámetro no es un directorio (es un archivo de otro tipo), deberá desplegarse el mensaje "***El parámetro < camino absoluto al directorio pasado como parámetro, aunque se ingrese un camino relativo > no es un directorio válido, sino otro tipo de archivo.***" por su salida estándar de errores, devolviéndose un ***2*** como código de retorno del script. Si no se tienen todos los permisos adecuados en el directorio pasado como parámetro para obtener los datos necesarios y poder hacer el listado, deberá desplegarse el mensaje "***No se tienen los permisos necesarios para acceder al directorio < camino absoluto al directorio pasado como parámetro, aunque se ingrese un camino relativo > y hacer listado.***" por la salida estándar de errores y deberá devolverse un ***3*** como código de retorno del script.

Si el script recibe un numero incorrecto de parámetros, se deberá desplegar un mensaje de error que diga "***Cantidad de parámetros incorrecta, solo se reciben los modificadores -r, -f, -v, -l y -n y un directorio válido***" por la salida estándar de errores y se deberá devolver un ***4*** como resultado (como código de retorno del script).

Si se recibe un modificador inválido, se deberá desplegar el mensaje de error "***Modificador < modificador pasado como parámetro > inexistente, solo se aceptan -r, -f, -v, -l y -n, y en ese orden en caso de estar varios presentes.***" por la salida estándar de errores y se devolverá un ***5*** como código de retorno del script.

En caso de cualquier otro posible error (si es que lo hay), se desplegara un mensaje de texto apropiado por la salida estándar de errores y se deberá retornar un ***6*** como resultado (como código de retorno del script).

Observación: los modificadores pueden estar presentes o no cuando el usuario utilice el script, pero se puede asumir que, en caso de recibirse modificadores, se ingresarán en el orden ***-r***, ***-f***, ***-v***, ***-l*** y ***-n***. Esta suposición es opcional, pudiéndose desarrollar, si se quiere, un script que acepte los modificadores en cualquier orden.

Si no hay errores, se desplegará por la salida estándar el listado de los caminos de los archivos, precedidos de la cantidad de caracteres que tiene cada camino. El numero de caracteres que tiene cada camino se separará del camino propiamente dicho usándose el carácter "***:***" y se devolverá un ***0*** como código de retorno del script.

Si tampoco se producen errores, pero no hay archivos que listar, se desplegará por la salida estándar de errores el texto "***No hay archivos para listar en el directorio < camino absoluto al directorio pasado como parámetro, aunque se ingrese un camino relativo > pasado como parámetro.***" En este caso, también se retornará un ***0*** como código de retorno del script (no se considera un error si no hay nada para listar).

Ejemplos de uso:

Suponiendo que el directorio *Escritorio/Directorio_inexistente* no existe y que el directorio corriente de trabajo es */root*, entonces, la salida del siguiente comando:

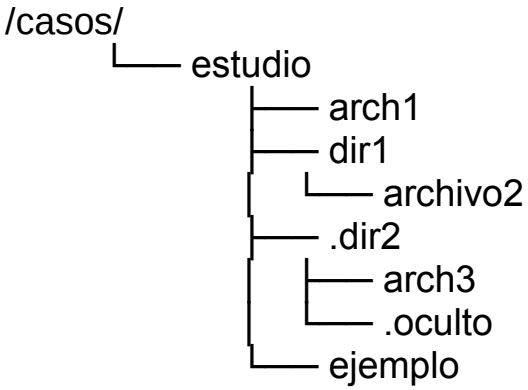
ej1_largo_caminos.sh Escritorio/Directorio_inexistente

Deberá desplegar (por su salida estándar de errores):

El directorio /root/Escritorio/Directorio_inexistente no existe, por favor ingrese un directorio válido.

o

Suponiendo que el directorio corriente de trabajo es */casos*, y no hay ningún problema para acceder al directorio */casos/estudio/* ni a sus subdirectorios, y si se tiene la siguiente estructura en el filesystem a partir del directorio /*casos/estudio/:*



Entonces, el comando:

ej1_listado_largo_caminos -r -v -n estudio

Deberá desplegar, por su salida estándar (considerando el anterior contenido del filesystem):

028:/casos/estudio/dir1/archivo2
*026:/casos/estudio/**.**dir2/arch3*
022:/casos/estudio/ejemplo
020:/casos/estudio/arch1
019:/casos/estudio/dir1
014:/casos/estudio

Observaciones:

La anterior salida será igual si el script se llama con un camino absoluto, por lo que el comando *ej1_listado_largo_caminos -r -n -v /casos/estudio* tendrá la misma salida antes mencionada. Los caminos mostrados son absolutos porque el script no ha recibido el modificador *-l*.

El archivo */casos/estudio/**.**dir2/**.**oculto* no se ha mostrado en el listado anterior porque se ha indicado el modificador ***-v***, que hace que solo se muestren los archivos no ocultos. En el listado anterior, si se ve el archivo */casos/estudio/**.**dir2/arch3* porque no es oculto, aunque se encuentre en un directorio oculto.

Ejercicio 2: script en Python. Valor 20 puntos máximo.

El objetivo del ejercicio 2 es hacer un script de Python que despliegue información producida en base a la salida del script creado en el ejercicio 1, agregándole funcionalidades y siendo más flexible en la interpretación de los modificadores (no requiriendo un orden especifico para poder reconocerlos).

El ejercicio 2 consiste en realizar un script en Python que sea de la forma:

ej2_largo_caminos_expandido.py [-h] [-r] [-f] [-v] [-l] [-i] [-t] [-o {l,d}] directorio

El script de Python recibirá las indicaciones según lo ingresado por el usuario en la línea de comandos (modificadores y directorio) y desplegará los caminos, sus largos y la cantidad de

directorios de cada camino, de los archivos del directorio pasado como parámetro. El script Python deberá aceptar los modificadores en cualquier orden, pudiendo estar o no presentes.

Además de desplegarse el largo de los caminos a los archivos listados, el script de Python agregará un campo que contenga la cantidad de directorios que hay en el camino de cada archivo (sin contar el propio archivo, si es que es un directorio también).

La cantidad de directorios, que tiene cada camino, estará después de la cantidad de caracteres que tiene cada camino (también se usará el carácter ":" para separar los campos), quedando el formato de la salida como sigue:

<largo del camino, rellenado con ceros>:<cantidad de directorios que tiene el camino>:<camino>

Varios de los modificadores del script de Python son iguales a las del ejercicio 1. Esas opciones serán tratadas de igual forma a ese script (***-r*** para que busque en forma recursiva, ***-f*** para archivos regulares, ***-v*** para archivos no ocultos, ***-l*** para que se listen caminos relativos al directorio pasado como parámetro). Además, la opción ***-o l*** (modificador ***-o*** con parámetro ***l***) deberá comportarse análogamente al modificador ***-n*** del script del ejercicio 1.

A las mismas opciones que tiene el script en bash del ejercicio 1, el script de Python le agrega más funcionalidades dadas por los siguientes modificadores:

Si el script en Python recibe el modificador ***-h***, desplegará una ayuda (por la salida estándar) explicando para qué sirven los diferentes modificadores y la sintaxis general del comando.

Si el script en Python recibe el modificador ***-o***, ordenará la salida por uno de dos posibles criterios. Si se recibe ***d*** como parámetro de la opción ***-o*** (el usuario ingresa en la línea de comandos ***-o d***), entonces se ordenarán los caminos por la cantidad de directorios que tiene cada camino en forma descendente (archivos con mayor cantidad de directorios en su camino primero y los que tienen la cantidad menor de directorios al final). Si se recibe ***l*** como parámetro de la opción ***-o*** (el usuario ingresa en la línea de comandos ***-o l***), entonces se listará la información ordenada por el largo de los caminos, en orden decreciente (primero se listarán los caminos más largos para finalizar con los más cortos). Esta opción es equivalente al modificador ***-n*** del script del ejercicio 1 (el script en Python deberá llamar apropiadamente al script del ejercicio 1, con ese modificador, para conseguir este ordenamiento). Las opciones ***-o d*** y ***-o l*** son excluyentes entre sí. Si se recibieran ambas, se considerará una cualquiera de ellas (por ejemplo, la última de ellas, es decir, la que se ingreso último en la línea de comandos al ejecutarse el script Python) y se ignorará la otra.

Si el script en Python recibe el modificador ***-i***, debe invertir el orden de los datos a desplegarse. Por ejemplo, si el script en Python recibe los modificadores ***-o l*** (modificador ***-o*** con parámetro ***l***) y ***-i***, entonces la salida saldrá ordenada por el largo de los caminos, según su cantidad de caracteres, en forma creciente (y no decreciente, como es el orden por defecto de la opción ***-o l***).

Si el script en Python recibe el modificador ***-t***, desplegará, al final del listado, un total indicando la cantidad total de archivos listados, con un mensaje que diga:

Se han desplegado <cantidad de archivos listados > archivos

El script de Python deberá utilizar apropiadamente el script del ejercicio 1, no resolviendo por sí mismo (en ningún caso) la información a obtener del filesystem. El script en Python deberá llamar al

script del ejercicio 1 con los parámetros adecuados para producir la salida a mostrar como resultado (haciendo el procesamiento necesario de la información brindada por el script del ejercicio 1 para adaptarla a los pedidos del usuario). Observe el paralelismo que hay entre la mayoría de los modificadores y directorio del script en Python (modificadores ***-r***, ***-f***, ***-v***, ***-l*** y ***-o l*** relacionado con ***-n***) y los modificadores y directorio del script del ejercicio 1.

El script en Python utilizará la verificación de errores del script del ejercicio 1 (si se producen errores al ejecutarse ese script) y reutilizará también los mensajes que envía a su salida estándar de errores (los mensajes de error, que correspondan, deben ser generados por el script del ejercicio 1, haciendo el script en Python un aprovechamiento de los mismos, reenvidándolos a su propia salida estándar de errores). En particular, esto se dará para el caso que el directorio no exista, o no sea un directorio sino otro tipo de archivo, o no se tengan los permisos necesarios para acceder a él. El código de retorno del script en Python será también el que retorne el script del ejercicio 1 en estos casos.

En caso que existan errores en la interpretación de los argumentos recibidos (cantidad incorrecta de parámetros, modificadores inválidos, parámetros incorrectos), se retornara un ***10*** como código de retorno y se desplegará, por la salida estándar de errores, una descripción resumida de la sintaxis general del comando (del script en Python) por la salida estándar de errores.

Si no se producen errores (se ha pasado un camino a un directorio válido, no hay problemas de acceso a él y no hay modificadores incorrectos ni otros errores en el ingreso de los parámetros), se desplegará la información que retorna el script del ejercicio 1 por su salida estándar, agregándosele la cantidad de directorios que tiene cada camino, en el orden que corresponda y con las características apropiadas según los modificadores indicados por el usuario. En este caso, el código de retorno deberá ser 0.

Si no hay errores y además no hay archivos para listar, también se aprovechará el mensaje que envía el script del ejercicio 1 en ese caso (por su salida estándar de errores), que será reenviado por la salida estándar de errores del script en Python. Como no es un error que no existan archivos, el código de retorno del script de Python también será cero en este caso (como se da también con el script del ejercicio 1 en este mismo caso).

Por ejemplo, si se ejecuta:

ej2_largo_caminos_expandido.py -r -i -o d -t /casos/estudio/

entonces la salida del script en Python deberá ser algo como (suponiendo el mismo contenido del sistema de archivos mencionado en el ejemplo del ejercicio 1):

014:2:/casos/estudio
020:3:/casos/estudio/.dir2
022:3:/casos/estudio/ejemplo
020:3:/casos/estudio/arch1
019:3:/casos/estudio/dir1
028:4:/casos/estudio/.dir2/.oculto
026:4:/casos/estudio/.dir2/arch3
028:4:/casos/estudio/dir1/archivo2

Se han desplegado 8 archivos

Observaciones:

Los ejemplos mostrados en este texto son solo ilustrativos.

Se ha desplegado la cantidad total de archivos listados por el parámetro ***-t***.

El listado anterior tiene dos archivos más que el listado del ejemplo del ejercicio 1 porque no se le ha dado al script de Python del ejercicio 2 el modificador ***-v***, y por eso se muestran también los archivos ocultos.

El listado esta ordenado por la cantidad de directorios, por el modificador ***-o*** con parámetro ***d***. El orden es inverso al de esa opción porque se ha indicado también el modificador ***-i***, que invierte el orden de la salida.

Elementos a entregar por cada grupo (tanto impresos como en formato electrónico)

Se deberán entregar los siguientes ***5*** elementos en el informe a subir online, siguiendo el procedimiento detallado en la última página de este documento, con titulo "**RECORDATORIO: IMPORTANTE PARA LA ENTREGA**". Además, estos ***5*** elementos deberán ser entregarlos impresos, junto con ***1*** elemento adicional en formato electrónico, en el momento de la defensa. Las impresiones solicitadas deben tener tamaño de letra fácilmente **legible**, preferentemente sin engramparlos ni ensobrar las hojas individualmente (para una mayor comodidad en el momento de la defensa), y en el siguiente orden:

Generales del obligatorio

1- Una página indicando **claramente** la siguiente información: los **nombres de los integrantes** del grupo (junto con sus **respectivas fotos**), sus números de estudiantes y los **caminos absolutos** donde se encuentra el script del ejercicio 1 y el script en Python del ejercicio 2.

(**1** elemento para la parte general del obligatorio).

Del ejercicio 1

2- El **código del script del ejercicio 1**.
3- Un **ejemplo de la salida del script**, considerando un caso en que ha ejecutado sin errores.

(tenemos en total **2** elementos distintos para el ejercicio 1).

Del ejercicio 2

4- El **código del script en Python del ejercicio 2**.
5- Un **ejemplo de la salida del script en Python**, considerando un caso en que ha ejecutado sin errores.

(tenemos en total **2** elementos distintos para el ejercicio 2).

TODOS LOS ELEMENTOS ANTEIORES TMBIEN DEBEN ESTAR IMPRESOS EN DIA DE LA DEFENSA.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

IMPORTANTE: SIN TODOS ESTOS ELEMENTOS IMPRESOS, NO SE PUEDE TOMAR LA DEFENSA Y POR TANTO, NO SE PUEDE CONSIDERAR ENTREGADO EL OBLIGATOIRO, AUNQUE SE ENTREGUE EN LINEA CORRECTAMENTE.

Observación: También debe ser entregada la maquina virtual en que se ha desarrollado el obligatorio, en un DVD.

OBSERVACION:

En total, son **5 elementos distintos** a entregar en forma online y también impresos obligatoriamente entre los dos ejercicios y la información general. La perdida mínima por cada elemento no entregado, tanto en forma electrónica como impresa, es de 3 puntos (pudiendo ser muchos más, como en el caso de faltar el **código del script en bash o del script en Python impreso y/o también en formato electrónico**, lo que implicaría la perdida de al menos **20 puntos**), aunque se puedan obtener por otros medios o estén en formato electrónico en el DVD entregado. Tenga en cuenta también que se pierden puntos si no está disponible **toda la información** pedida en cada uno de esos **5** elementos distintos.

Maquina virtual y directorio con archivos

Se entregará la maquina virtual donde se ha desarrollado el obligatorio (en el momento de la defensa, ya que no se puede subir online), que deberá contener todo lo pedido. La contraseña de **root** de la maquina virtual deberá ser la misma que se usa en el curso (**root01**). Además se entregará un directorio conteniendo toda la estructura de archivos involucrada en la creación del obligatorio, incluyendo también una versión de todos los archivos entregados en forma online. Este directorio no estará dentro de la maquina virtual (estará aparte de ella, pudiendo colocarse en el mismo DVD, en un directorio aparte).

Observación: EN CASO DE EXISTIR ALGUNA DISCREPANCIA ENTRE LA INFORMACION ENTREGADA ONLINE Y LA IMPRESA EN EL MOMENTO DE LA DEFENSA, SE TOMARÁ COMO VALIDA LA QUE IMPLIQUE EL MENOR PUNTAJE PARA EL GRUPO, por lo que se recomienda imprimir la misma información entregada online para la defensa, sin hacerle cambios, que solo pueden implicar pérdida de puntos y no un beneficio adicional.

Para entregar online los elementos solicitados en el formato electrónico (excepto la máquina virtual), se debe seguir el procedimiento detallado en el anexo "RECORDATORIO: IMPORTANTE PARA LA ENTREGA", en la siguiente página.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios** (Cap.IV.1, Doc. 220)

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de la misma.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. La entrega se realizará desde **gestion.ort.edu.uy**
2. Previo a la conformación de grupos cada estudiante deberá estar inscripto a la evaluación. S**ugerimos realizarlo con anticipación.**
3. **Uno de los integrantes del grupo de obligatorio será el administrador del mismo** y es quien formará el equipo y subirá la entrega
4. Cada equipo (2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar)
5. El archivo a subir debe tener **un tamaño máximo de 40mb**
6. Les sugerimos **realicen una 'prueba de subida' al menos un día antes**, donde conformarán el **'grupo de obligatorio'.**
7. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
8. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc)
9. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor pasar por la oficina del Coordinador o por Coordinación adjunta **antes de las 20:00hs.** del día de la entrega

Si tuvieras una situación particular de fuerza mayor, debes dirigirte con suficiente antelación al plazo de entrega, al Coordinador de Cursos o Secretario Docente.