

parcial2.pdf



DonOreo



Fundamentos del Software



1º Grado en Ingeniería Informática



Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Para entregar el examen, cree un fichero comprimido en formato tar.gz con todos los ficheros del examen, para ello ejecute la órden:

tar -czvf apellidosNombreDelAlumno.tar.gz todos_los_ficheros_del_examen (entre espacios en blanco)

y súbalo a la plataforma antes de las 13:20. Consejo no esperen al tiempo límite de entrega, comiencen a subirlo a las 13:15. No se aceptan entregas por email.

1.(3 ptos.) Cree un guión (script) llamado **examen1.sh** en bash con el siguiente funcionamiento:

1. **./examen1.sh --c argumentos.** Los argumentos pueden ser:

-fichero_fuente fichero_destino . Copia el fichero fuente en el destino ó

-solo directorio_destino (si no existe se crea). Copia los ficheros planos del directorio de trabajo en el directorio_destino ó

2. **./examen1.sh --h** muestra esta ayuda en pantalla, ó

3. **./examen1.sh --sh argumento.** El guión nos deberá mostrar los usuarios que están usando el tipo de Shell especificado como argumento. El argumento podrá ser: csh, o sh o bash..

Si el guión (script) se ejecuta sin argumentos o con un número erróneo de argumentos o un tipo de argumento no esperado (si se esperan archivos y son directorios o viceversa) dependiendo de la opciones seleccionadas, el script mostrará el error y la ayuda.

2.(5 ptos.) Escriba un archivo llamado Makefile para compilar con la orden **make** los archivos (.cpp) y (.h) del directorio que se le ha facilitado en esta tarea. El directorio dado deberá ser tomado como el directorio de trabajo. El nombre del programa generado deberá llamarse examen2. El archivo Makefile debe cumplir las siguientes reglas:

1. Cuando se invoque con la orden make deberá generar los archivos objetos, la librería llamada mateslib.a que contendrá todas las funciones y el ejecutable examen2 en el directorio de trabajo.

2. Deberá disponer de regla llamada limpiar que borre todos los archivos objetos, la librería mateslib.a y el ejecutable examen2.

3. Disponer de la regla instalar que deberá realizar la siguientes acciones: generar el ejecutable examen2 y la librería mateslib.a; crear los directorios ./bin, ./SOURCE, ./CABECERAS y ./LIB sino existen; mover el ejecutable examen2 al directorio ./bin; mover todos los archivos (.cpp) al directorio ./SOURCE, mover los archivos (.h) al directorio ./CABECERAS, mover mateslib.a al directorio ./LIB y por último en el directorio de trabajo solo debe quedar el Makefile y los directorios ./bin, ./SOURCE, ./CABECERAS y ./LIB, el resto de archivos deben ser borrados.

4. También deberá disponer de la regla desinstalar condicionada a que la regla instalar haya sido invocada con anterioridad. La regla desinstalar deberá restituir el directorio de trabajo a la situación inicial. En concreto, en el directorio de trabajo **solo** deberán aparecer los archivos (.cpp) y (.h) que han sido proporcionados inicialmente, junto con el archivo makefile realizado, borrando el resto de archivos o directorios que puedan existir.

3.(2 ptos.) Modifique el fichero Makefile para depurar el programa compilado en el ejercicio anterior. Realice un script de gdb llamado examen3.gdb y que realice las siguientes actividades de depuración al programa del ejercicio anterior:

1. Defina un punto de ruptura en la función veuclidea(vector,vector).

2. Defina un punto de ruptura condicional en la función garea que detendrá la ejecución cuando el triángulo tenga una altura mayor o igual a 5.

3. Realice 3 iteraciones del bucle y muestre la variable i en todas las iteraciones.

Los archivos del examen(sin resolver) los he subido aquí:

http://bit.ly/Archivo_examen_II_B3_tar_gz

```
#!/bin/bash
# Título: examen.sh
# Fecha: 08/02/2021
# Autor: DON OREO

ayuda="Ayuda:
1- ./examen1.sh --c argumentos. Los argumentos pueden ser:
    -fichero_fuente fichero_destino . Copia el fichero fuente en el destino ó
    -solo directorio_destino (si no existe se crea). Copia los ficheros planos del
    directorio
    de trabajo en el directorio_destino

2- ./examen1.sh --h muestra esta ayuda en pantalla, ó

3- ./examen1.sh -sh argumento. El guión nos deberá mostrar los usuarios que están
    usando
    el tipo de Shell especificado como argumento. El argumento podrá ser: csh, o sh o
    bash..\n"

case $1 in
    --c) if [ $# == 3 ];then
            if [ -f $2 ] && [ -f $3 ];then
                cat < $2 >> $3
                printf "Archivo '$2' copiado a '$3'"
            else
                printf "'$2' y '$3' no son archivos planos o no existen\n"
            fi
        elif [ $# == 2 ];then
            cp /* $2
            printf "Archivos copiados a '$2' "
        else
            echo Numero incorrecto de argumentos
        fi
    ;;
    --h) printf "$ayuda"
    ;;
    --sh) if [ $# != 2 ];then
            echo "Uso: ./examen --sh <shells> donde 'shells' son 'bash' 'sh' 'csh'"
            exit 1
        fi

        cat /etc/shells | grep -x /*/$2
        if [ $? == 1 ];then
            echo Debe introducir una de las siguientes shells:
            cat /etc/shells | cut -d '/' -f 3,4
        else
            echo Los usuarios con la shell $2 son:
            cat /etc/passwd | grep $2 | cut -d ":" -f1 | sort
        fi
    ;;
    *) printf "Numero de argumentos incorrectos
$ayuda"
    ;;
esac
```

```
# Nombre archivo: Makefile
# Uso: make
# Autor: DON OREO

CC = g++
GDB = -g
CPPFLAGS = -Wall
SCRS = pr.cpp complex.cpp geom.cpp vector.cpp dist.cpp
OBS = pr.o complex.o geom.o vector.o dist.o
HDRS = complex.h geom.h vector.h dist.h

examen2: pr.o complex.o geom.o vector.o dist.o mateslib.a
    $(CC) $(GDB) $(OBS) libmates.a -o $@

pr.o: pr.cpp $(HDRS)
    $(CC) $(GDB) -c $(CPPFLAGS) $<

mateslib.a: complex.o geom.o vector.o dist.o
    ar -rvs libmates.a $^

complex.o: complex.cpp complex.h
    $(CC) $(GDB) -c $(CPPFLAGS) $<

geom.o: geom.cpp geom.h
    $(CC) $(GDB) -c $(CPPFLAGS) $<

vector.o: vector.cpp vector.h
    $(CC) $(GDB) -c $(CPPFLAGS) $<

dist.o: dist.cpp dist.h
    $(CC) $(GDB) -c $(CPPFLAGS) $<

instalar: examen2 libmates.a bin SOURCE CABECERAS LIB cleanObs

desinstalar:
    mv CABECERAS/*.h ./
    mv SOURCE/*.cpp ./
    rm bin/* LIB/*
    rmdir bin CABECERAS LIB SOURCE

limpiar:
    rm *.o mateslib.a examen2

bin:
    mkdir $@
    mv examen2 $@

SOURCE:
    mkdir $@
    mv *.cpp $@

CABECERAS:
    mkdir $@
    mv *.h $@

LIB:
    mkdir $@
    mv libmates.a $@

cleanObs:
    rm *.o
```

```
break veuclidea  
  
break garea(triangulo) if m.h >=5  
  
# Parte 3 NO FUNCIONA. Borrar esta linea  
  
display i  
continue  
display i  
continue  
display i
```