Отчёт по ДЗ №2

кормилицын владимир, БПИ226

1. В данном скрипте в бесконечном цикле ( в цикле while с условием продолжения true) постепенно выводятся числа Фибоначчи через пробел. Для того, чтобы вывод происходил с некоторой задержкой (5 миллисекунд), используется команда sleep

*#!/bin/bash*

x=0; y=1; i=2

while true; do

    i=$((i + 1))

    z=$((x + y))

    echo -n "$z "

    x=$y

    y=$z

    sleep .5

done

1. В данном скрипте в цикле for генерируется строка вида “../../../../” для переходов в родительские директории. Количество подъёмов LEVEL равно первому аргументу, указанному при запуске скрипта. После этого выполняется команда cd для перехода в новую директорию, и в консоль выводится текущая директория (echo). В конце запускается новый процесс с интерпретатором командной оболочки (exec $sh), чтобы пользователь увидел переход в новую директорию (во время работы в эту директорию переход интерпретатор (т.е. эти переходы делаются в рамках созданного процесса, исполняющего скрипт), но не ‘терминал’ пользователя. Поэтому в новой директории запускается sh / bash (в общем случае, новый интерпретатор), чтобы в консоли пользователя отобразился переход)

*#!/bin/bash*

LEVEL=*$1*

for ((i = 0; i < LEVEL; i++)); do

    echo $i

    CDIR=../$CDIR

done

cd $CDIR

echo "You are in: $PWD"

sh=$(which "$SHELL")

exec $sh

1. В данном скрипте на основе расширения введённого файла вызывается архиватор для распаковки архива. Выбор происходит на основе команды case и указанных паттернов имени файла. Проверка того, что введён аргумент, делается в первом if (если не введён, выводится подсказка). Проверка того, что указанный файл существует, делается во втором if (если не существует, выводится сообщение о не валидном имени файла)

*#!/bin/bash*

if [ $# -lt 1 ]; then

    echo "Usage: $(basename "*$0*") os\_hw2.zip"

    exit 1

fi

if [ -f "*$1*" ]; then

    case *$1* in

        \*.tar.xz)   tar -xvf "*$1*"                         ;;

        \*.tar.bz2)  tar -jxvf "*$1*"                        ;;

        \*.tar.gz)   tar -zxvf "*$1*"                        ;;

        \*.bz2)      bunzip2 "*$1*"                          ;;

        \*.dmg)      hdiutil mount "*$1*"                    ;;

        \*.gz)       gunzip "*$1*"                           ;;

        \*.tar)      tar -xvf "*$1*"                         ;;

        \*.tbz2)     tar -jxvf "*$1*"                        ;;

        \*.tgz)      tar -zxvf "*$1*"                        ;;

        \*.zip)      unzip "*$1*"                            ;;

        \*.pax)      cat "*$1*" | pax -r                     ;;

        \*.pax.z)    uncompress "*$1*" --stdout | pax -r     ;;

        \*.rar)      7z x "*$1*"                             ;;

        \*.z)        uncompress "*$1*"                       ;;

        \*.7z)       7z x "*$1*"                             ;;

        \*)          echo "'*$1*' cannot be extracted/mounted via extract()" ;;

    esac

else

    echo "'*$1*' is not a valid file"

fi

1. В данном скрипте реализована функция быстрого возведения в степень (за O(logn)). Перед вызовом функции проверяется, что передано хотя бы 2 аргумента, делается в

if [ $# -lt 2 ]; Далее вызывается функция bin\_pow с аргументами, указанными при выполнении скрипта, в которой объявляются 3 целочисленные переменные и проверяется, что показатель степени >= 0. Далее, в цикле выполняется алгоритм возведения. Когда показатель становится равным 0, в консоль выводится результат возведения первого аргумента во второй.

*#!/bin/bash*

function bin\_pow {

    declare -i n="*$1*"

    declare -i p="*$2*"

    declare -i res=1

    if ((p < 0)); then

        echo "exponent should be non-negative integer"

        return

    fi

    while true; do

        if ((p & 1)); then

            ((res \*= n))

        fi

        ((p /= 2))

        if ((p == 0)); then

            echo "$res"

            return

        fi

        ((n \*= n))

    done

}

if [ $# -lt 2 ]; then

    echo "Usage: $(basename "*$0*") pow\_base exponent"

    echo "Example: $(basename "*$0*") 10 5"

    exit 1

fi

bin\_pow "*$@*"

1. В данном скрипте реализована функция перестановки бит числа в двоичной записи. Функция интерпретирует входной параметр как целое число, в двоичной записи которого 64 бита, и меняет i-й бит с 65-i ым (считая, что биты нумеруются с 1-ого до 64-ого). Перед вызовом функции проверяется, что при запуске интерпретации скрипта был введён хотя бы один аргумент.

*#!/bin/bash*

function bit\_reverse {

    declare -i n="*$1*"

    declare -i t=0

    ((n = (n << 31) | (n >> 33)))

    ((t = (n ^ (n >> 20)) & 0x00000FFF800007FF))

    ((n = (t | (t << 20)) ^ n))

    ((t = (n ^ (n >> 8)) & 0x00F8000F80700807))

    ((n = (t | (t << 8)) ^ n))

    ((t = (n ^ (n >> 4)) & 0x0808708080807008))

    ((n = (t | (t << 4)) ^ n))

    ((t = (n ^ (n >> 2)) & 0x1111111111111111))

    ((n = (t | (t << 2)) ^ n))

    echo "$n"

}

if [ $# -lt 1 ]; then

    echo "Usage: $(basename "*$0*") number"

    echo "Example: $(basename "*$0*") 2147483648"

    exit 1

fi

bit\_reverse "*$1*"