Оперативни системи

Вежби

Задача 1

Да се напише програма во C која работи со процеси и нитки. Главната програма (родител процесот) треба да креира дете процес, на кого ќе му прати низа од 100 цели броеви. Дете процесот најпрво треба да ја пополни низата од 100 цели броеви со нули. Потоа, дете процесот треба да креира N нитки (N се внесува од тастатура во родител процесот), притоа на секоја нитка дете процесот и испраќа (како аргумент) случаен позитивен цел број „К“ (најмногу 500). Секоја нитка прави „К“ промени во низата и потоа завршува со работа. Секоја една промена во низата значи случајно одбирање на еден елемент од низата и менување на неговата вредност. Првата половина од нитките ја менуваат вредноста на елементот со зголемување на неговата вредност за 1, додека пак втората половина на нитките ја намалуваат вредноста на елементот за 1. Откако ќе завршат со работа сите нитки, главната нитка (дете процесот) печати на екран колку елементи од низата ја имаат променето својата вредност (т.е. не се повеќе нула). Родител процесот завршува откако дете процесот ќе заврши. Генерирањето на случајни броеви се прави со помош на функцијата rand( ).

БОНУС: Проверката колку елементи од низата ја имаат променето својата вредност да ја прави родител процесот.

Задача 2

Да се напише PERL скрипта која како аргументи добива имиња на две датотеки. Скриптата треба да ја измине втората датотека, да ги детектира сите наредби кои што се наоѓаат во втората датотека, и секоја наредба, заедно со аргументите, да ја испише во посебна линија на екран. Секоја наредба започнува со името на наредбата, после кое следат низа аргументи (или нема ниту еден аргумент). Имињата на наредбите се сместени во првата датотека т.е. датотеката што се испраќа како прв аргумент. Секоја наредба е сместена во нов ред во првата датотека.

Пример:

./perlscript.plx naredbi.txt vlez.txt

naredbi.txt

cat
cmd
cntrl
pop

На екран:

cat –l –p
cmd ppl.ls –l
cntrl pap –o ppp.exe
pop
cmd empty
cat *.*

vlez.txt
cat –l –p cmd ppl.ls –l cntrl pap –o ppp.exe
pop cmd empty cat *.*

Задача 3

Да се напише Shell скрипта која што најпрво ќе ги најде датотеките „naredbi.txt“ и „vlez.txt“ во системот и ќе ги ископира во тековниот директориум. Потоа, ќе ја повика perl скриптата и ќе ги изврши една по една наредбите што се враќаат како одговор од PERL скриптата. Резултатот од извршување на секоја една од наредбите се печати во излезна датотека „izlez.txt“, во која што преку прилепување (append) го додава излезот на секоја една од наредбите.

Задача 4

Да се напише C програма за кодирање на дадена слика од страна на „N“ нитки. Секоја нитка, како аргументи добива: <koj_del_od_slikata_go_kodira> < kolku pikseli ima toj del>. Сликата е запишана во текстуална датотека, така што секој знак е пиксел кој има код на боја од 0 до 255. Секоја нитка паралелно врши кодирање на сликата, притоа секоја си знае кој дел е нејзин и колкаво парче (ги добива горните леви координати на подматрицата која треба да ја кодира и димензијата на подматрицата). Кодирањето се прави така што се зема пиксел по пиксел и се проверува дали неговите соседни пиксели (околу него во подматрицата) имаат иста вредност или се разликуваат најмногу за 1. Доколку е тој услов исполнет, тогаш на екран процесот ги печати координатите на пикселот и бојата со која се кодира. Доколку не е исполнет условот, на екран не се печати ништо. Сликата е со димензии 1096x1080. Главната нитка ја пополнува матрицата од датотеката и ги започнува останатите нитки. Бројот на нитки со кои се кодира се внесува како аргумент од командна линија, додека пак, податоците што се праќаат на секоја нитка (како аргументи) се внесуваат од тастатура од главната нитка.

Оперативни системи

Вежби 2021

Задача 1

Да се напише PERL скрипта која како аргумент добива име на влезна датотека. Скриптата треба да креира излезна датотека со исто име како и влезната, само со наставка “_out”. Скриптата треба да ја измине влезната датотека линија по линија, и во излезната датотека да ја запише линијата, така што доколку во линијата текст се наоѓа датум, датумот да дојде прв во линијата, па потоа, без менување на редоследот, да дојде и останатиот текст од линијата. Форматот на датумот е “DD-MM-YYYY”, притоа, доколку има повеќе од еден датум, се запишуваат во излезната датотека онолку линии колку што има различни датуми во влезната линија, притоа текстот што следи е без датуми. Доколку скриптата се повика со втор аргумент и доколку вториот аргумент е валиден датум, тогаш дополнително и на екран се печатат оние линии каде што го има тој датум. Зборовите во датотеката се одделени со празно место.

Пример:

Vlez.txt

Prv den od godinata 01-01-2021 dodeka pak posleden den e 31-12-2021
Posleden den za Fevruari e 28.02.2021 godina.

Vlez.txt_out

01-01-2021 Prv den od godinata dodeka pak posleden den e
31-12-2021 Prv den od godinata dodeka pak posleden den e
Posleden den za Fevruari e 28.02.2021 godina.

Задача 2

Да се напише SHELL скрипта која како аргумент добива листа од имиња од датотеки. Скриптата, за секоја датотека (име од датотека) да провери дали ја има како датотека во тековниот директориум, или под-директориуми. Доколку ја има, проверува дали датотеката (во нејзината содржина) има барем еден валиден датум (DD-MM-YYYY), и доколку нема, ја игнорира. Доколку има валиден датум, тогаш ја повикува PERL скриптата и како прв аргумент го праќа името на датотеката, и т.н. со сите датотеки што се пратени како аргументи од командна линија. За секоја датотека што ќе биде пратена на PERL скриптата, се печати на екран нејзиното име и колку различни датуми се пронајдени.

Задача 3

Да се напише програма во C која работи со процеси и нитки. Главната програма (родител процесот) добива низа од наредби кои треба да ги изврши како аргументи од командна линија. Родител процесот треба, најпрво, да креира онолку деца процеси колку што има наредби наведено од командна линија (наредбите се без аргументи). Пота, треба да креира онолку нитки, колку што има наредби, така што, секоја нитка ќе чека и ќе брои колку секунди му било потребно на соодветниот процес да заврши. Тоа значи дека, првата нитка ќе биде задолжена за првата наредба т.е. за првиот процес, втората за вториот и т.н. Секоја нитка брои колку време се извршувал нејзиниот процес (наредба) и кога ќе заврши кажува колку вкупно секунди му требало да заврши, а потоа и самата нитка завршува. Откако ќе завршат сите процеси/нитки, тогаш главниот процес/нитка печати на екран колку време и требало на секоја наредба да се изврши.

БОНУС: главниот процес/нитка печати на екран колку време и требало на секоја наредба да се изврши во растечки редослед според времето на завршување, најпрво да се отпечати која наредба и требало најмалку време да заврши, па потоа следната наредба и т.н.

Оперативни системи

Вежби 2022

Задача 1

Да се напише PERL скрипта која како аргументи добива бројка и листа од имиња на датотеки. Првиот аргумент (бројката) означува колку од датотеките се влезни, останатите се излезни датотеки. Скриптата треба да провери дали има барем една излезна датотека според бројката. Потоа, треба да направи статистика за секоја излезна датотека, дали името на излезната датотека (без екстензија) се појавува како збор во содржината на влезните датотеки. Во излезните датотеки, во секоја редица одделно, се запишува колку пати е пронајдено името на излезната датотека, за секоја влезна датотека, притоа доколку името на излезната датотека не се наоѓа во содржината на некоја влезна датотека, се запишува вредност 0.

Пример:
./perl.plx 4 dat1.txt dat2.txt dat3.txt dat4.txt dat5.txt dat6.txt
Резултат:
cat dat5.txt
dat1.txt 1
dat2.txt 0
dat3.txt 7
dat4.txt 2

cat dat6.txt
dat1.txt 0
dat2.txt 0
dat3.txt 1
dat4.txt 5

Задача 2

1. Да се напише SHELL скрипта која како аргумент добива бројка X. Скриптата треба да провери дали во тековниот директориум ја има датотеката „files.txt“. Доколку ја има, тогаш продолжува да се извршува скриптата. Во датотеката, во секоја линија одделно, има имиња на датотеки. Скриптата треба да провери дали има најмалку онолку имиња на датотеки колку што е бројката X што се праќа како аргумент на скриптата, и доколку има најмалку онолку имиња на датотеки колку што е X, продолжува да се извршува скриптата, со тоа што исто така проверува дали првите X датотеки постојат во тековниот директориум. Потоа, скриптата ја повикува PERL скриптата од првата задача, со тоа што како прв аргумент на скриптата испраќа вредност X/2, а во продолжение ги испраќа првите X датотеки од „files.txt“.

   БОНУС: откако ќе заврши PERL скриптата, на екран да се отпечати содржината на излезните датотеки т.е. на вторите X/2 датотеки што се испраќаат на PERL скриптата.

Задача 3

Да се напише програма во C која работи со процеси. Програмата (главниот процес) треба да дозволи да се внесе целобројна вредност од тастатура K (најмногу 100). Потоа, главниот процес треба да дозволи да се внесат K наредби од тастатура, притоа секоја наредба што се внесува има име на наредбата и три аргументи. Главниот процес, за секоја наредба внесена од тастатура треба да креира дете процес, на кое ја испраќа наредбата (заедно со трите аргументи), притоа, дете процесот треба да ја изврши таа наредба. Секое од деца процесите, пред да ја изврши наредбата, одбира случаен број од 1 до 20 (со наредбата rand()%20+1) и чека толку секунди пред да почне да ја извршува наредбата. Дете процесот печати колку секунди ќе чека пред да почне со чекање.

Главниот процес, ги чека деца процесите да завршат со извршување, потоа печати на екран дека и тој завршува.
БОНУС: Главниот процес да измери колку секунди им требало на сите деца да завршат со извршување.