Parte de teorie

• Scrieti comanda ls care afiseaza toate numele de directoare din directorul ./proc ce reprezinta imaginea unui process inchis pe disc (numele directorului contine doar cifre)

(sa construim expresia numelor de directoare formate doar din cifre -expresii regulate

Cu ls si grep)

• Care va fi rezultatul urmatoarei comenzi shellscript echo {00...99} – e o multime

Pot fi intrebari din zona comenzilor simple sau inlantuiri de comenzi (pipe)

• Descrieri efectul urmatoarei directive macro:

$ (EXE):$(OBJS) //$(EXE) contine numele executabilului (este tinta) , $(OBJS) contine lista de obiecte

$(LD)-o $@$(OBS)// $@- contine numele tintei curente (se suprapune cu $(EXE) )

• Prezentati pe scurt apelul de sistem fork:

Vom primi o paralela intre 2 apeluri de sistem diferite (gen fork si exe, signal si sigaction – de ce a fost nevoie de signal si sigaction istoric)

• Completati urmatoarea secventa de cod astfel incat P1 trimite un semnal SIGUSR1 lui P2, la primirea semnalului P2 va trimite un mesaj prin pipe catre P3:

Main(){

If(fork())

//P1 luam valoarea lui fork si trimitem un SIGUSR1 catre P2, putem utiliza wait (pid)

Else

If(fork())

//P2 signal

Else // folosim wait

//P3

}

• Fire de executie

Care va fi valoarea variabilei i la finalul executiei urmatorei secvente de cod:

Int main(void){

Int i=0;

pthread\_create(...); //

++i;} //i are valoarea 1, dar putea avea orice valoare daca era utilizat in firul de executie si era declarat global

SAU afisare text din firele de executie. Fiecare fir sa afiseze cate un cuvant (SISTEME DE OPERARE) pthread\_join

**Parte** **de shellscript**

Sa se realizeze un shellscript ce primeste ca si argument un director D. Scriptul va parcurge recursiv directorul D si va cauta si afisa calea catre toate subdirectoarele ce nu contin niciun fisier (sunt goale sau contin doar subdirectoare).

Cel mai simplu e sa ne punem un flag pt fisierul curent (daca contine ceva)

Sau functie separata care verifica daca fisierul curent e gol sau nu si sa nu uitam sa facem si pentru fisierul radacina

Alte exemple:

Sa facem comparatii intre structuri de directoare

Timestamps

**Parte de C**

Sa se scrie un program C care contine doua procese (parinte+fiu), procesul parinte primeste ca si argument un nume de fisier F si un cuvant C. Parintele citeste fisierul text F si-l trimite printr-un pipe procesului fiu care va cauta cuvantul C in fisierul primit. Daca procesul fiu gaseste cuvantul C, va anunta procesul parinte printr-un semnal la alegere. La primirea semnalului procesul parinte va afisa „gasit”. Daca cautarea se incheie fara success, atunci parintele va afisa mesajul „inexistent”. Cautarea va avea o limita de timp de 5s, dupa care procesul parinte va termina executia procesului fiu, ii va prelua starea si isi va incheia la randul lui executia indiferent de starea cautarii.

Rezolvare:

!Important: Mai intai pipe si apoi fork.

Cuvantul e salvat intr-o variabila

Cautare cu strstr in bufferul a ce am citit din pipe

Inainte sa pornim procesul de citire sa folosim signal sau sigaction

Cum ne dam seama daca cautarea se termina fara success: nu am primit semnalul SIGUSR1 si procesul copil si-a incheiat executia.

Folosim combinatia cu alarm pe care il definim in cadrul proc parinte

Sigkill

**Punctele se primesc pe subcomponentele problemei. E bine sa ne atingem de toate subiectele.**