

Sisteme eragileak

Batch Sistemak

Ez eraginkorre S/I-ren zein

Mono programazio

Soluzioa

Multitask/Multiprogramazio

S/I-ren zein beste atazle agin andoren

Programazio exekutio tarteketue

Programazio Memoriak gore

Elikarrelintza (Denboraz-partekatutako sisteme)

Quantomendi (denboraz-korte txikilek)

Erauzten-denboraz txikilek

Multi-programazio sentzadie
(unix - ez jabealuna)

Ezabiltzale aritzeko sistemeak

- Segurtasun

Interfaze grafikoak

Sozializazio → Linux

Administrazio-atazle errestu

Sarea (TCP/IP)

Bederaz-zerbitzari-estekia

Sisteme autonomoak

Gailu Mogillorak

Sistema eragileko murriztu (Android)

S/I-gailu berride eta interakazio berride

Sarea konektatu noudhi.

Sisteme deiekh Bufferrek betetzean egiten dira, hau sistema berean egon daitezken abiadure differentziak eruditela. Audiorioz urritarre bufferre eta pitcher-area es dira berdinak eta es dute modo a berean idatzitako momentu

Aparteke 2c

Prozessele de Escektu

Exekutizile 5 modo de egora

↳ Ranking → mortzu.

Sleeping → le edo gain

Stuck → trabektu.

Zombie → Zombi

Stopped → Blokeatu gelditik.

Programa guztiek esin memoria egora berabat erregiteko
ku kompartzela.

1. Prozesue sortu → fork() → id bieltu

2. Exekutatu → Hainbat prozesu
ezin

↳ programen berregatu eta
testuinguru esarrir.

- Scheduler → politika bat erabili, ze prozesu aukeratu
erabili.

- Dispatcher → Exekutu osoa, osoa progresu ordetzailea.

3. Blokeatu

↳ Sistema dientzen bidez

↳ prozesue blokeatu → Stopped egora,
utzi

4. Esentu

↳ Sistema errotua esikorra

↳ blokeatzeko prozesue-pista

Lock beate
Programa bat
exekutatu.

5. Blokeatu

Stop

egoz jerrri corribit, d
~~o aldean ematen~~

5. Buleku → exit(0) sistema deia erabilitzen da buletzaleo,
eta horren bidez, prozesu erabilitzen ari den bali bide guztiek
askeztzen dira. Horrela, horregoz prozesu exekutatzera utzi.

Zu gorditxo de gizara unea bihio ikar bilaketa?

Los unen uinburtxo gorditxo de et. init prozesuak hiribileko beregean.

Prozesuak

↳ Dependientek edo independentek

Proseso independentek \rightarrow hau bi prosesu desberdineri argumentu berdinak pista da. Idiotzago, erantzue berdinak daude, leize artean eta baitago dependientziak eta adoriora exklusio bidezki bere odatik egingo du. Ezinez bere lortuz.

Proseso dependientes \rightarrow hau horrek bi prosesuen artean dependentzia gogo da, adoriora erantzue desberdinak bitarte geritzeko. Beste, biak baino gehiagozko exkultatzen esteroko.

Bi edo du mota desberdin beraizten dira.

Elastiko - Matematizale eredu. Hau da, elastikoak tankeen funtsezko buffer batetan zehaztu daitezke elementuak batez berriz hartzteko.

Berez-Berbitzari eredu: Elastikoak zahar direnean periodikoki koste prozesu batetan sa elementurik beotu. dega. Konprobatzear do.

Komunikazioen

↳ prozesua arteko informazio trukelek

Sinbromakzioa

↳ Funtzioa erabiliz \Rightarrow Explizituki

↳ Implizituki \rightarrow Komunikazio-mekanismoa bidez

Selezio kritikoa

↳ CPU bidera ohi bere la egin dezakea selazia

↳ Konkurrentzia exekutiboa denean gertatzen da, CPU biderak i behio desberdinak ondorioz edo diri bi bider exekutiboak ibili

↳ Hainbat prozesu erabiltzen dituzte ~~punt~~ kude-zatieri denitz

Fitxa komunikazioak erabiltzen denei fitxategia de, PIP-enkin
alderatuta, ez dene aronmoa eta bi inturnetako atxigerririk del
era desberduna. Besteak idetzai edo izateko derrigorretarik de bi
hainbat mortxu egingo. Eta, norbit besterak idetzeari oni hain
biderak izaten egingo biderak du, hauela anitz testu

Maza-trukearen biderako komunikazioa sotetean zego, adoreago
metoduan dudela.

↳ Buzoiak FIFO (First in First Out): Buzoi mota hau
hainbat funtzionarioak bideratzen sortu ditzake:

multimed(): Pentsatzailekin fitxategia bat sortuko da,
bestean de puntuak bidera bidera begiratzen da.

int uMed (char *path, int mode, int disp)

Tito bat sortu nahi badu erabilera ondoren gauza d.

Id = multimed (str, ~~0~~ S_FIFO | 0666, 0);

Beste funzio bbe, ~~intelligenza~~
ull fifo) usun lauter Fito izeneko fitadegi
beresia sortzen du, potz-ean jarritako izenekin, "node" edo
ezorri ueni dire bilwendu definituen dura.

Bent gabelo bidoile ere badele, Pipe deritez.

int pipe() \rightarrow pipe bat sortido du \rightarrow non fitategi batere
bidatzeko den puntuera, eta beraren ideia edo irakurri egun
dantzaera, normalean pfd horri o balioc sartea zio irakurketa
egiteko eta beraren ideabideko, eragileko egun ondoren
informazio gaixotekoa da.

PPG-a

PfP6-a
La Norvegia balestelor de pi prozessuer esteia
Pipe bateu, pipe-ec sortu due prozessuri ~~balestelor~~
Synchronizatu cu idago du kate prozessuerin baine oldetariz
Cinelele idago da Baldin etc, os kate sortear kate pipe bat
Strategi berari begjider.

Prozesse im bestehenden System - die ich

Identifikasjø

`getpid()`: Prozessore identifizieren und töten sie

`getppid()`: Profesuror, processu gurezaren identifikatzailea bultzatzen du.

getroffen). Prozesse sind, Erblichkeiten identifizieren und daraus

Proben Sorte

Foro U: Sistema dei lavori processu ~~ambiori~~ bat sortee du han
processu - unen deritao. Processu omache gurasoien erregatzi gehieke berak
ditu, ~~vez~~ identifikatu dena izen edite, labeak o belinek berak du eta
gurasoak omacheak.

except: permuta bedeute position Zeichen \neq "program", konnndo...
exklusiv los der Bullezen null beliebig jemi beliebter Zeichen bestim. Arbeit
de jehin beliebter dale posetklos Zeichnien argumentu nopen
Zein den.

excep: kuso bantuan bi ~~agama~~ soalnya pastinya disebut
parametro

exofitisch wohin der Prozess des Louche d'iver organisiert,
Adonitz hervorhebt, dass diese Arbeit es die betriebsspezifische
Zielsetzung organische Bopun zulässt.

Prozessdiagramme erläutern

~~Potiti~~ blektasvi-algoritmoval dende komstado, kri' schedlerd definitzen ditu. Honek adierazten baitu enetakoak dira ordene. hainbat irizpide dende alorretako ordene.

CPU-darbore caii kienh beste prezelvci.

First-Come First-Serve \rightarrow FIFO bilden bilden implementieren da
~~prozeß~~ dekorate ggü. defensie prozeß exklusivität der Dekorat. Es ist
Ergebnisse bei es habe ~~ergänzt~~ ggü. defensie prozeß exklusivitätsbesch.

Shortest Job First (SJF): Lan getrie gegen bular die proseso
ekelitasa de lcheago posisioen, oso ofrisitasa de yerkde lkeledeko
piso gelice diter proseso ihoiz ea ekelitasa.

Darbora hudo diceva che teste prozessari

Ronald Rubin → processus bekostigde denbare kerne bestrijding
do bore exclusione gevulddele de "quantum", denbare denbare-kern
kern, denbare kerne horrelen beletten en beide exclusione,
processus-lichaam bekeer jstzen de als lungeno processus
exclataten hesciba de.

Prozesse lebenswerten oder ausgrenzen
Prozess hat andere Lernziele, schulische Ziele hat
es anderen die lebenswerten oder es kann diese den
Prozess beeinflussen geben.

Lebenswerte: Wollen wir es da absetzen
Lebenswerte machen was denn eigentlich die Prozesse
erwartet haben da einen.

Andererseits kann da jeder zu Prozess erwartet das Lebenswerte
es bedeuten zu politisch erzielt zu sein schulisch.



Fitxategi sistemek ezaugarriak

Ikuspuntu funtzional baten arabera, fitxategi bat, ezaugarri batzuk dituen informazio multzo bat dela esan daiteke. Fitxategien ezaugarriak hauek dira:

- Informazioa era iraunkorrean gordetzen da, programek erabiltzen dituzten beste objektu batzuk ez bezala (adibidez aldagaiak).
- Karaktere kate batez irudikatzen den izen bidez identifikatzen dira. Izenak esanahi berezia duen zati bat izan dezake (luzapen bat) edo ez, Linux-en bezala. Luzapena fitxategiaren mota zehazteko erabil daiteke.
- Fitxategiak direktorioetan multzokatzen dira. Direktorioek antolaketa egitura hierarkikoa dute (zuhaitza). Horrela Fitxategi Sistema osatzen dute. Fitxategi baten izen absolutuak (edo zuhaitzaren erronko bidea) fitxategi sistemako fitxategi bakar bat egiten dio erreferentzia.
- Batzuetan, fitxategi bat direktorio bat baino gehiagotik elkarbana daiteke (loturen bidez).
- Fitxategien gainean eragiketa anitz egin daiteke (sortu, irakurri, idatzi, ezabatu, aldatu, ezaugarriak aldatu ...).
- Erabiltzaile anitzeko sistemetan atzipen baimenen bidez babesten dira, erabiltzaile bakoitzak atzipena duen edo ez adieraziz, eta atzipena izanez gero, egin ditzakeen eragiketak zehatzuz.
- Fitxategi baten informaziorako atzipen unitatea erregistro logikoa da. Sistema eragiletan normalena, sistema deitarako atzipen unitate gisa karakterea izatea da. Erregistro logikoak gehienetan sekuentzialki atzitzen dira, baina era zuzenean atzitzeko aukera ere izan ohi da.

Gure konputagailuan gordetako fitxategiei erreparatuz gero, eduki, tamaina eta erabilpen oso anitzak dituzten fitxategiak aurkituko ditugu. Adibidez, sistema eragileak eta aplikazioek konfiguraziorako byte gutxi batzuetako testu fitxategi asko erabiltzen dituzte. Bestalde, multimedia fitxategiak oso handiak izan ohi dira. Azken hauek normalean aldaketa gutxi jasaten dute, aplikazioetako kode fitxategiek bezala, halere, badira aplikazioak (testu editoreak edo *gcc*) biziraupen laburreko aldi baterako fitxategiak erabiltzen dituztenak.

Fitxategi sistema batek ezaugarri zehatz batzuk dituen euskarri-gailu baten beharra du. Funtsean, gailuak biltegiratze iraunkorra (ez-hegazkor) eskaini behar du. Gainera, atzipen kopuru mugaezina baimendu behar du, irakurketan zein idazketan. Halere, badaude sistemak non irakurri bakarrik egin daitekeen, eta sistemak non nahikoa den, baita hobea ere, behin baino ez idaztea (adibidez back-up -etan).

1. Bai, izen absolutuak fitxategi batzuk soili egiten dialetiko erreferentzia.

2. Funtsean biltegiratze iraunkorra eskaini behar du.



SISTEMA ERAGILEEN GARAPENA

2 Konputagailu pertsonalen sistema eragileak

1970ko hamarkadaren amaieran hardwarea merkatu zen eta mikroprozesadoreak agertu ziren, ondorioz posible zen erabiltzaile bakar bat sistema dedikatu bat eskaintza, ezaugarri hau funtsezkoa da sistema pertsonal batean. Hasiera batean, konputagailu pertsonalen sistema eragileak, adibidez MS-DOS (1981), **erabiltzaile bakarrekoak** ziren, (ez zuten babes-mekanismorik) eta **ataza bakarrekoak** ziren; hau da, ez ziren 30 urte lehenagoko jatorrizko monitorean oinarritutako sistemenean oso desberdinak, desberdintasun nagusia terminal baten bidez zuzenean jardun daitekeela da. Baina 1990ko hamarkadatik aurrera, konputagailu pertsonal merkeetan erabilgarri dagoen hardwareak multi-ataza sistemak onartzen dituzte (Mac OS, Windows, LINUX) eta **erabiltzailearen interfaze grafiko** aurreratuak ere onartzen dituzte. Gainera, gaur egungo konputagailu pertsonalak erabiltzaile anitzekoak dira. Konputagailu pertsonalen ezaugarrietako bat erabiltzaileak sistemaren administratzalea izan behar duela da, ondorioz beharrezkoa da administrazio-atazak ahalik eta gehien erraztea.

Informatika hedatzeaz gain, konputagailu pertsonalen sorrerak eragin gehigarri bat du. Ordura arte, enpresa handien multiprogramatutako sistema zentralak urrunetik, terminal "tonetoetatik", atzi zitezkeen telefono-modemen bidez. Konputagailu pertsonalek terminal hauek ordezkatu dituzte eta orain sistema autonomoak dira, hauek konputazio-ataza batzuk burutu ditzaketen sistema autonomoak dira eta sistema zentralera deskarga dezakete. Bereziki edozein komunikazio-protokolo exekuta dezakete era autonomoan. Protokolo estandarra adostu ondoren (adibidez TCP/IP) konputagailu pertsonalak beraien artean ere zuzenean komunika daitezke. Konputagailu zentralaren kontzeptua desagertzen doa; gaur egun beraien artean sare-azpiegitura baten bidez komunikatzen den konputagailu-multzo bat ematen da. Baliabide zehatz batera sarbidea eskaintzen eta kudeatzen duen sareko makina izango da baliabide horren **zerbitzaria**. **Bezeroek** baliabidea atzitzen dute **bezero-zerbitzari eskema** baten bidez. Konputagailu-sareen agerpen eta hedapenak asko korapilatu du ez bakarrik sistema eragilea, baita ere bertan implementatzen diren zerbitzuak, (*middleware* deritzona). Ondorioz Interneteko esparruan hedatzen diren **sistema banatuak** sortu dira eta zerbitzuen eskema eta kontzeptu oso elaboratuak sortu dituzte, adibidez *web-zerbitzuak* eta *cloud computing* zerbitzuak. Ikastaro honetan sistema zentralizatuak aztertzera mugatuko gara, baina ez dugu ahaztu behar gaur egungo egoera askoz ere konplexuagoa dela.

LINUX (1991) sistema eragilea UNIX sistema-familiako kidea da eta konputagailu pertsonaletan erabiltzera zuzenduta dago, baino beste ingurunetan ere erabil daiteke. LINUX nabarmena da **software libreko¹** produktu bat delako, fenomeno honek softwarearen merkatua irauli du, bereziki sistema eragileenarena. Software librearen ezaugarri nagusia da ez duela kopia eta banaketa mugarik, **jabedun softwareak** ez bezala. Internet beharrezko baliabide bat izan da ideiak eta kodea komunitate batean partekatu eta trukatzeko software librea garatzen duten programatzaleen artean. Ondorioz eta frogatu den bezala, softwarearen egokitzea eta produktu berrien garapena errazten da, gainera erroreak zuzentzen dira eta bertsioak erraztasunez doitzen dira. GNU² bezalako erakundeek software librearen kopia, eraldaketa eta birbanaketa lizenziak ematen dituzte baldintza bakarrarekin, distribuzio berriak iturburu-kodea barneratu behar du³. LINUX da gaur egun eta askogatik UNIX familiako gehien erabiltzen den sistema.

¹Free software. Ez nahastu *freeware*-arekin. Ez da ezta ere doako software bezala ulertu behar.

²<http://www.gnu.org>

³Lizenzi honi *Copyleft* deitzen zaio.

Häufige Anwendungen

↳ Erhältbare Ballone etc. über Internet

90er Jahre

↳ Multi-plateforme Interfaz gestaltet

Ganz egal → Erhältbare Anz.

↳ Administrativ - nutzbar → erreichbar.

Komplett - personalisiert → terminielle Ordnung

↳ System autonom

↳ Zweck verbindet. ist zu verbinden.

Zeitalter

↳ Serbien etc. Nutzbarer erheblich erhöht

Zeitverlust

↳ Bezeichnung → Bereich verbindet entnommen
Von Zeit

91 → Linux Software libre

GNU ↳ Kopie etc. berechtigt
↳ Lizenzrecht einer

↳ bedingt

↳ Distributionen berücksichtigen

Modus benutzt

```

/*
 *----- temporizadoreak.c -----*/
/*----- definizioak.h -----*/
#include <nds.h>
#include <stdio.h>
#include "definizioak.h"

void DenbEtenBainendu01()
{
    /* Denboragailu batzen etenak baimendu (Timer0)
     * Horretarako lehenengo eten guztiak galarazi eta bukaeran berriro baimendu
    */
    void DenbEtenGalarazi01()

    /* Denboragailu batzen etenak galarazi (Timer0)
     * Horretarako lehenengo eten guztiak galarazi eta bukaeran berriro baimendu
    */
    void konfiguratuTemporizadorea01()

    /* Temporizadorearen konfigurazioa bere SI erregistroak aldatuz
     */
    void DenbEten010()

    /* Denboragailuaren (Timer0) etenaren tratamendurako zerbitzu errutina
     */
    etenkuTeaketa.c
    /*----- */

    /* añadir los includes que sean necesarios
     */
    #include <nds.h>
    #include <stdio.h>
    #include "teklatua01.h"
    #include "temporizadorea01.h"
    #include "definizioak.h"

    /* Procedura honek eten taula hasieratu egiten du. Horrela, eten kudeatzaitaleal jakin ahal izango du
     * eten eskaera bat iristen zaionean zein den exekutatu behar duen zerbitzu errutina.*/
    void etenak010()
    {
        /*...
         * Teklekintegridad gauza bera egin daiteko egoerekin
         */
        #define HASTERIA 0
        //...
        // aldagai orokorrak
        extern int EGOERA;
    }
    irqSet(IRQ_TIMER0, DenbEten01);
}

```

```

/*
jokoac - Programa Nagusia
*/
----- teklatura.c -----
/*
Kode hau garatu da do votoren "Simple sprite demo" adibidea eta Jaeden Ameroneren beste
adibide batean oinarrituta.
----- */

#include <nds.h>
#include <stdio.h>
#include "definizioak.h"
#include "grafikoak.h"
#include "spriteak01.h" //Teklatuarekin egiten den zerbaitean ondorioz spriteekin lan egin behar
izanez gero.

#include <nds.h> //nds-rako garatua dagoen liburutegia
#include <stdio.h> //c-ko liburutegi estandarra sarrera eta irteerako funtzoak definitzen
dituena
#include <stdlib.h> //c-ko liburutegi estandarra memoria erreserbak eta zenbakik
konbertsioak egiteko
#include <unistd.h> //Sistema eragileen arteko kompatibilitatea zurratzeko liburutegia

//Geuk garatutako fitxategiak

#include "grafikoak.h"
#include "fondoak01.h"
#include "spriteak01.h"
#include "definizioak.h"
#include "jokoac01.h"
#include "jokoac01.h"

touchPosition pos_pantaila;
void jokoac01()
{
    erakutsiAtea01(); //fondoak.c fitxategian definituta
    initSpriteMem01();
    SprteakMemorianGorde01();
    PaletaNagusiaEzarrir01();

    while(1)
    {
        touchRead(&pos_pantaila);
        while(pos_pantaila.px==0 && pos_pantaila.py==0)
            touchRead(&pos_pantaila);

        PaletaNagusiaEzarrir01();
        ErakutsiAtealrekita01(); //fondoak.c fitxategian definituta
        erakutsiAtealrekita01();
    }
}

void initTeklatura01()
{
    int tekla01; //sakatutako tekla gordetzeako aldagai
    int TeklaDetektatu01()
    {
        if(tekla01) //TRUE itzultzen du teklaten bat sakatu dela detektatzen badu
    }

    int SakatutakoTekla010()
    {
        //Sakatutako teklaten balioa izultzen du: A=0;B=1;Select=2;Start=3;Esc=4;Zk=5;
        //Gora=6;Behera=7;R=8;L=9;
    }

    int IrakurriTeklatutuKesta01()
    {
        //Inkesta tekla bat sakatu dela detektatu arte (gora, behera, ezkererra, eskuinera)
        //teklaten balioa izuli
    }

    void TekEtenBaimendu010()
    {
        //Teklatuaren etenak baimendu
        //Lan hau burutzeko lehenengo eten guztia galarrazi behar dira eta bukaeraen baimendu
    }

    void TekEtenGalarazi010()
    {
        //Teklatuaren etenak galarrazi
        //Lan hau burutzeko lehenengo eten guztia galarrazi behar dira eta bukaeraen baimendu
    }

    void konfiguratuTeklatura01()
    {
        //Teklatuaren konfigurazioa bere S/I erregistroak aldatuz
    }

    void TekEten010()
    {
        //Teklatuaren zerbitzu errutina, teklatuaren etenei arreta egiteko
    }
}

```

- SISTEMA ERAGILE KASILKOSH -

- SO hamarkadan Batch sistemak (leheneng SEak)
- SE : programaren exekuzioa kontrolatzeko.
- Asukion lan gabe. SI gaileku mota lepoak
- Eraginkune goa: multiprogramazioa (60h)
- terminals, komunikazioa. Olin protokoloak bat egiten
legten bitartean.
- 60h. → denb. partekutako sist. (UNIx)
- berrizalari → eramangana (73)
- Garezik → erabilizgarilean antzekoak

90. Hamarkadan, Multi atzelak sortu eta interbide grafikoak hobetzi.

Gaur egun, erabilizgarilean antzeko eta administrazio-atzelak erosiak, konputagailu, portabaleko sortu, sisteme autonomoak.

Zerbitzarioa sortzea eta erabiltzaileen eskeritzan du helburua zehatz
betear. Ordorez, bederatzi Bezeroa zerbitzarien eskerotan bilditako atzelak.

91. Linux sortu, UNIx-en oinarritutak, eta software librea, ordoriraz
oz do kopie eta banaketa mugarrak.

GNU lizentziak anitzeko erregulazioak ditu, bildutako bilera, distribuzio
berriak iturburuak modua bernezketa izatea de.

- Urteak pasa ahala gero eta txikirago eta erromangarniago eta eraginkorago.
- Konputagailu pertsonaletarako sistema eragileak murriztu → telefono adimenduetarako eta tabletetarako (Mac, windows)
- Bestela, sistema eragile espezifikok (Android).
- Sarrera / Irteera gailu berrak (kamera, sentsoreak...), interaktivo berrak (uktena-pantaila).
- Gailu hauek sarez konektatu darteazke gure portaidetzorik gabe, nonetik sistema.
- Denbora errealeko sistemak (autoaren oreakoren sistema kontrola), epemuza bat bete behar.
- Erobilera ordkorreko s/i, ~~ba~~ denbora errealeko atazekin ~~etc~~ bateragamitik barna epemuza gainditu dezaketenen (multimedia aplikazioak)
- Sistema eragileen ibilbidea zirkularra → software eta hardwarearen differentzia nabarmena eta ez ziren guztiz bateragamik. Orain mugikorrekin berdina gertatzen da.

Aitzol Elu

Oihan Arroyo

Ander Mancisidor

Batch sisteneke: txartel zuldetude

Ez eraginkorre S/I-ren zain
Monoprogramazio.

Soluzioen

• Multitasko/multiprogramazio → Etxarrelintza

- S/I-ren zain besta atzak gertatu
- Prog. exekuzio tarteletu
- Memoria progr +

- Quantumal (denboratarte txikia)

↓ Erantzun-denbore

↓ txikide izan deder

multiprogramazio sententzia
ezin.

Hori, denbore pertekeltenetako sistema
esaten zain.

(UNix-eraz jabea)

Erabiltsaile arrazko sisteneke

- Segurtasuneko kasoak
- Interfaze grafikoak
- Sozializazio, → Linux
- administracio-atzak errebatu

Sareak

TCP/IP

Bezero-zerbildari esaten
sistene berdutek

Gailu mugikorrak

Sistema ergile espezifikoa

