Conceptes Avançats de Sistemes Operatius

De

Genís Bosch

CASO 2018-2019

SISTEMA OPERATIU

Programa intermediari usuari màquina. Entorn d'execució convenient i eficient per executar programes. Gestió màquina segura. Protecció als usuaris.

MACH INTERFACES

- Task: procés classic d'Unix. Dividit en dos: contenidor de recursos és el task. Entitat passiva, no executada. task_create
- Thread: procés classic d'unix. Dividit en dos: entorn d'execució és el thread. Part activa. Cada task suporta més d'un thread. Tots comparteixen recursos task. Mateix espai d'adreces però amb sp i pc diferents. thread_create
- Port: canal de comunicacions per a comunicar dos threads. Un recurs, propietat d'una task. Thread té acces al pertanyer a una task. get_privileged_ports
- Host: Obtenció informació basica: processadors, memoria disponible, versió kernel, data i hora.
- Processor: informació processadors. Slot, master. Processor set. Tasks associades.
- Device: Open read close write map
- VM: demanar i alliberar memòria. copiar regions. mapejar memory objects.

COMENTARIS

Al utilitzar mmap, no és necessari copiar l'informació llegida per el read de kernel a user. El read, en canvi, fa una crida i necessita el copy to user, però permet treballar a nivell de byte. en canvi, mmap necessita un read de com a mínim una pàgina.

En cas de llegir molta memoria, malloc utilitzarà mmap, mentre que al demanar menys memòria, s'utilitzarà sbrk. Sbrk avança el punter amb l'espai de memòria reservat, mmap mapeja una memòria virtual a física en qualsevol posició, o la que el kernel decidexi.

Write sense sobreescriure un fitxer, necessita llegir per a escriure el que s'ha solicitat més el que resta per a aconseguir un bloc. Si es sobreescriu, no cal utilitzar read.

sbrk(increment): modifica la mida del segment de dades canviant el program break. només al final de la data segment del procés.

mmap (addr, size, prot, flags, fd, offset): mapeja fitxers a memòria tot creant un nou mapeig a la memòria virtual del procés. Qualsevol part del procés.

COMPARATIVA CLONE/FORK

Clones linux: funció clone(), crea un procés nou. Depenent dels parametres. pot o no compartir parts del seu context amb el pare.

Mach clones: task_create + thread_create, tots comparteixen el context d'execució. Llavors fixar estat thread_set state.(coneixement arquitectura on executa) Finalment thread_resume.

Pthread : creació transparent a l'arquitectura, s'utilitza pthread_create().

SINCRONITZACIÓ

S'utilitzen:

- Instruccions atòmiques.
- Spin locks:

  __sync_lock_test_and_set(&lock,1) crida atòmica a while(lock==1); lock =1;

  Per a evitar sobrecarrega d'instruccions.
- Padding (espai entre variables per evitar colisions a cache)
- Mutex: pthreads i variables de condició-> exclusió mutua en pthread.
- Futex: linux mutex, no sempre accedeix a sistema si no hi ha contenció.
- Grand central Dispatch: Interficie basada en cues de traball que substitueix la interfície de pthreads. El programa principal crea una sèrie de cues i encua funcions per executar. (workers). L'aplicació crida al planificador GCD.

COMTADORS HARDWARE I UTILITATS

Cycles, cache misses, loads stores, tlb hits. Llibreria PAPI per extreure informació.

GCC/BINUTILS:

- ld linker
- as assembler
- dlltool lib management
- gproof profiler

- nm/objdump/ readlef object viewers
- size/strings simple viewers
- ELF -> codi + dades + simbols i adreces + reubicacions + infomració debug

AVALUACIÓ RENDIMENT

- Temps d'execució.
- Speedup (sequencial/ paral·lel).
- Bandwith (dades transmeses/temps).
- Latency (cost iniciar operacions).
- Estadistica tractada amb mitjana i desviació estandard. Eines com top, htop, ps, time, /usr/bin/time, vmstat, iostat, gettimeofday (temps desde 1-1-1970 "epoch").
- Varis clocks: realtime (epoch), monotonic (afectat per daemon ntp), monotonic_raw, process_cputime_id (temperatures consumit CPU proces), thread_cputime_id (temps cpu gastat per fil).

VIRTUALITZACIÓ

- Entorn virtual sencer a SO i aplicacions, diferent a la maquina fisica.
- Procés en el sistema host.
- Ofereix protecció (maquina aillada), compartició de recursos (processadors, memoria, disc, xarxa), i facilitat per engegar o parar i fer proves.
- Spha de permetre a aplicació executar instruccions en mode sistema: virtual user mode + virtual kernel mode en mode fisic.
- System consolidation: ajuntar els serveis oferts per diverses màquines fisiques en una de sola (física), utilitzant una màquina virtual per a cada una de les originals.
- Intel VT-x: monitor VMM, ID proces virtual, taula de pagines extesa.
- AMD-V: processor guest mode, control data structure.
- Linux containers: compartir un sol kernel entre jerarquies de processos.

SISTEMES DE FITXERS

GESTIÓ DE QUOTES

Habilitat de limitar la quantitat de dades que un usuari (o grup d'usuaris) té en un sistema de fitxers(partició). És independent del sistema de fitxers. Requereix que el sistema de fitxers la suporti i que el kernel la suporti. La partició ha de ser muntada amb opcions usrquota. Activa amb quotaon i quotaoff, edita edquota. elimina amb quota –v. Grace period és el temps durant el qual l'usuari pot arribar al limit hard. un cop passat, només pot arribar al limit soft.

JOURNALING

Diverses operacions al disc en cada operació al fitxer. (esborrar fitxer requereix entrar al directori, marcar inode lliure, marcar blocs de dades com a lliures…). Si el sistema s'apaga al mig d'aquestes operacions, recuperarho és costos. Journaling garanteix consistencia del sistema de fitxers i pot guardar-se en un disc diferent. Escriptures asíncrones al journal (registre de totes les escriptures/lectures). Inclou una estructura de dades de suport a la recuperació del sistema de fitxers. Canvis atòmics al journal. S'escriu per avançat. Dos tipus, fisic (grava una copia de cada bloc), logic (grava els canvis a les metadades) (pot tenir dades corruptes).

FITXERS EN XARXA

- NFS: transparent als usuaris, implementat amb remote proc calls i centralitzat en un servidor.
- AFS: distribuit en diferents servidors
- OBEX: protocol d'intercanvi de dades amb dispositius mòbils. Fitxers de tot tipus i entrades de calendari, tarjetes de visita... Per a bluetooth, infrared, usb, sèrie...
- FUSE: interficie lligada amb obex. Estructura dels servidors. Permet treballar remotament a un filesystem a partir de fuse al sistema operatiu i obex com a aplicació del servidor, que es conecta mitjançant bluetooth, wifi...

MECANISME DE BOOT

BIOS/DOS, inici del disc conté MBR: bootstrap code area, seguit d'un 0, physical drive time, bootstrap code area, signature, 5a5a, partition table, 55aa. Partition table conté bootloaders, ja que no té prou espai per tots els kernels.

NOUS SISTEMES

- UEFI: open firmware i unified extensible firmware interfrace. Pot usar el MBR per protegir una taula de particions que de l'altra manera seria fàcil esborrar. Conté directoris de boot, amb bootloader.efi i fitxers de configuració. Permet també que cada bootloader tingui un directori per ell sol
- GPT: nova taula de particions, tipus GUID (Global Unified Identifiers) Aquesta conté un protective MBR per detectar GPT en sistemes antics, n'hi ha una copia a final del disc.

GESTIÓ DE DISPOSITIUS

Estructura del kernel: conté dispositius de caràcter, de block, de usb i de xarxa.

DISPOSITIUS DE CARÀCTER

Permeten l'accés a la informació caràcter a caràcter. Punts com imatge de la memoria principal de l'ordinador (amb adreces físiques), ports, memòria del kernel, fuse(per a comunicar la llibreria de FUSE amb el driver del kernel, el descriptor del fitxer obtingut s'usa per relacionar punt de muntatge i aplicació que s'ocupa del sistema de fitxers en espai usuari). Utilitzats també pels terminals vcs i vcsa. Es preparen seleccionant major i minor del dispositiu, i programant el controlador en un mòdul extern. register_chrdev i unregister_chrdev. Copy to user i copy from user.

DISPOSITIUS DE BLOC

Tals com ram, SAATA, SCSI, IDE, permeten l'accés a l'informació a nivell de block. Dona suport a discos. Un mòdul pot rebre arguments en ser carregat (insmod modul argument=N). Els arguments es defineixen dins el modul. Suport gestió de memòria interna al kernel en linux/slob_def.h i linux/slub_def.h o linux/slab.h. Funcions register_blkdev i insmod per a ser carregat.

DISPOSITIUS USB

Connectats en un bus, cadascun atén a les peticions que se li dirigeixen. (vendor:product com a ID).

- Libusb és la llibreria d'usuari multi-SO que permet llistar, accedir i veure caracterisitques dels dispositius USB.
- USB endpoints-> buffer per a la transmissió de dades. registre o regió mapejada en memòria. enpoint 0 és de control i sempre existeix. Fins a 30.
- Operacions USB -> suporten connexió, desconnexió i suspend/resume. probe enregistra l'interficie. conegudes per file_operations.

DISPOSITIUS DE XARXA

Ethernet, detectable, inicialitzable, i suspend/resume.

TEMPS REAL

Puntualitat, arribar al temps especificat sempre. Determinisme: coneixer el temps d'execució de cada funció o tasca. Deadline: temps en que la tasca ha d'haver acabat.

- Falla quan el resultat està disponible massa tard.
- Soft real time – perd utilitat si els resultats arriben tard. – plans de vol de companyies aèries.
- Firm real time – descartar resultats que arriben tard. – video rendering.
- Hard real time – una fallada de deadline és una fallada total del sistema. – sistema de control d'un avió.

CARACTARÍSTIQUES

- Deadline: Temps màxim en el qual la tasca s'ha d'haver executat, per atal que el sistema pugui continuar funcionant.
- Deadline miss: és la pèrdua del deadline en una tasca, les consequències poden ser greus.
- Tasques:
  - Periòdiques -> es repeteixen indefinidament seguint un període d'activació, habitualment responen a un event extern.
  - Aperiòdiques -> comencen i acaben sense repetir-se.

- Jitter -> variació en el temps d'execució d'un procés deguda a la seva interacció amb altres processos o interrupcions.
- Mentre que en un OS es permet el multitasking per a donar fairness, en un RTOS s'utilitzen les prioritats per a tractar els processos de forma estricta. Tampoc es permet la sobrecàrrega del sistema, és a dir, més tasques de les que es poden executar.

SCHEDULER RTOS

Tenen en compte la prioritat dels processos i també el seu estat. Els canvis d'estat són produits per events externs o per una altre tasca. Els processos ready amb més prioritat corren. En cas de compartir prioritat, es tria el que fa més temps que no s'ha activat.Els processos que esperaven un event són activats quan passa l'event en l'ordre fixat per la seva prioritat.

POLÍTIQUES

- Rate-monotonic scheduling: Per tasques periòdiques, amb prioritat estàtica depenent de la freqüència de la tasca. Sempre s'executa la que té la freqüència més alta.
- Earlies Deadline First: Per a tasques periòdiques, però amb prioritat dinàmica, que canvia en funció del deadline de la tasca. Sempre s'executa la tasca amb el deadline més pròxim.

HERÈNCIA DE PRIORITATS

Per a evitar una inversió de prioritats, on el flux més prioritari espera per entrar en una regió crítica que té un flux menys prioritari. En aquest cas, es transfereix la prioritat del flux més prioritari al flux menys prioritari per tal de que surti de la regió tant aviat com pugui.

SOSTRE DE PRIORITAT

Recursos amb prioritats, amb un nivell superior al de la tasca més prioritaria que l'usa. En usar-lo, totes les tasques s'executen en aquest nivell de prioritat.

RT-PREEMPT

Incorpora idees de RT a Linux, proporcionant carecterístiques de hard RT a Linux (permet regions crítiques amb preempció, implementa herència de prioritats de kernel i converteix gestors d'interrupcions en fluxos, que poden canviar de prioritat).

XENOMAI I RT LINUX

Xenomai

Suport per a temps real en Linux. Es tracta d'un partxe al kernel, amb utilitats d'usuari. Afegeix un conjunt de característiques de temps real a la configuració del kernel, i permet executar serveis en temps real al costat d'aplicacions que no requereixen temps real. Implementa un ordre en la distribució d'events: Adeos, basat en dominis de procecció, amb prioritat estàtica. Llavors, els events es distribueixen prime al domini més prioritari. Aquests dominis son eficients, amb ràfagues curtes d'execució i poden inhibir events o interrupcions. Els Xenomai

threads, que poden córrer en mode kernel o usuari, són l'entorn d'execució standard de Xenomai. En resum, els event i interrupcions són tractats en primera instància per el ring 0, més prioritari, que és xenomai, i en cas de ser events de baixa prioritat, s'els deixa per a ser tractats per Linux de forma secundaria, al ring 1. Xenomai 3 és un co-kernel, és a dir, aplica preemption sobre l'scheduler de linux amb els processos RT.

RT Linux

Fa correr linux com a preemptive proces, no com a co-kernel. Així doncs, es tracta d'un RTOS en el que hi ha una tasca en background que és Linux. Utilitza POSIX threads.

POSIX Threads

Permeten scheduling i tasques, però no es poden assignar prioritats.