Resource management

(1) process management

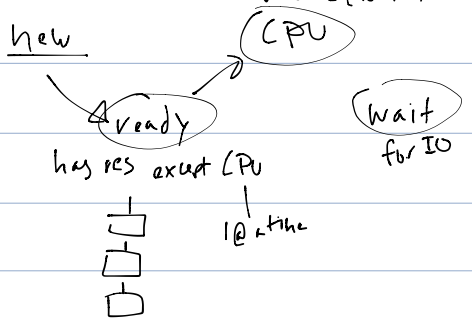
↳ program in execution

↖ program + execution environment { registers ...

move from running → waiting state

waiting for resources to be available

State transition



Synchronization between child & parent

(2) PCB (process control block) PID (process descriptor)

child tells parent 'exit'

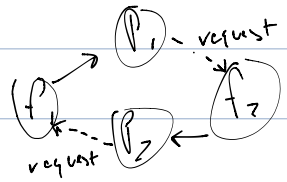
process ID, register contents, program counter, child processes

(3) CPU scheduling

(4) process synchronization

Critical section problem,

(5) Process deadlock handling



deadlock 4 conditions

(6) process communication uses buffer producer CPU ⟷ consumer printer

3 ways: signals (UNIX/Linux)

shared memory value variable in memory

message passing handled by kernel

memory management

(1) memory space allocation/deallocation

(2) trace of free and allocated memory space first fit, best most, next fit

(3) virtual memory management (paging, segmentation, or both)

32-bit

2^{32} bytes = 4GB

2^{64} bytes

File management

mass-storage management

(1) storage allocation / deallocation

(2) disk scheduling

(3) free storage space management

protection & security

Cannot access resources directly

System calls used to access system resources

pipe → otherwise default output

a | b

fork(), execve, wait, exit, open(), read(), write(), pipe(),

trapped to kernel

UNIX/Linux approach

wait: invoked by parent

exit: child

open file: you'll have to exit if doesn't give parameters

file descriptor (pointer pointing to locations of file)

saves time

file pointer = internal pointer used to get next character

CreateProcess(...); ← library function call

Windows

more flexible

API (Application Programming Interface)

Types of OS

(1) batch systems - program treated as a batch (one thing)

JCL (Job Control Language)

everything given to one at a time

F1 → C1 ⇄ F2 → F3

mount FORTRAN Compiler Once

Set up time is the last bottleneck

turnaround time = time when submit until everything (once) is done

Common problem

solutions

multiprogramming - put multiple programs in memory / good for time sharing systems (2)

(CPU switches rapidly between processes)

response time

(3) real time systems - microprocessor

3 types of info text, audio, video processed by

Soft RT systems - if miss 1-2 frames, it's fine

Hard RT systems - must react instantly, can't miss frame

Handheld systems

download and install VirtualBox

www.os-book.com

Linux Virtual Machine

www.virtualbox.org