lecture#1 Chapter#1 Introduction Feb.10.2018 HI Operating Systems Goals: not at a private to a storight (1) overall goal: Execute user programs. (2) primary goal : Conveniency ( and on the computer machine. machine language it's easier for the user to interact with the OS. (3) Secondary goals: efficiency. Eft Computer Resources; (1) CPU . (2) Memory. > OS manages Resaurses. (3) I/O devices. - FIRMANCISTAL I perating System (US); A set of Algorithms that run the computer machine. ) OS manages the computer resources. 105 must manage the computer Resources efficiently Bridge With the Harriston T

Other goals of OS: is insultilization of computer resources. Utilization of CPU: make the CPU as busy as possible printing Utilization of Memory: to benefit or use memory as much as possible. Utilization of I/O devices of \* System performance can be measured with through put. 2011 \* Throughput: number of jobs (programs) that finish execution per 6 Unit of time. 6 mouter Resource 6 Computer 6 MOMONA (S 6 3) [10 devices (6 HARDWARE :-bilt 2011/100010 physical devices = chips, wives 6 run the computer Power supplies. 1 - micro program: A primitive software 1 aprilling and and that communicates with Physical devices which is an Interpretter that Setches 1

Rendinciale Instruction. > Fetch: execute machine - machine Language (Assembly Language) OPERATING SYSTEM: 2 U2 Un Application Packages Software 1 NINSI Compiler DB --- etc 123 Machine Language Microprogram Hardware physical devices a. 107 11/1 12 3 APPLICATION PACKAGES: 4 User Programs. I Operating System Views: OS Goals (1) It's a control program; It controls the execution Soverall goal. of all programs. (2) Extended muchine: Extension of the physical 4 primary goal. Machine. That is it hides all the complexity of system

programming and provide the user with a simple clean machine to deal with

STUDENTS-HUB.com

-

- 1

TIL

D.

- Carl

-2

-

1

Uscanded with a campoanter

luser - 20 Amochine Computer have have here have have The user doesn't have to deal with machine or use cissemply I machine Language. (3) Resource Manager: It manages efficiently " Secondary good. the computer resources. (4) Kernel: part of the OS that's always running and executing instructions Et History & Evolution of OS :-AND AN ANTI AND AN Card reader Printer, Raper, tape input sassa output strong of the property of the provide in the x 1/2 > each line needs a card £ T 11 11 lines the user needs 200 1 1 1 Cards . I Marine 1 \* Hexadecimal was used. CONTRACT MINY

Uscanded with Transcanner

Et Early software :-Cartina 12 Muchine Language Assembly Language (Assemblers) - Linkers: the addition of software to program. - Compilors. \* Performance is poon - a great deal of time is wasted in set up time. No overlap between IIO& CPU execution Low CPU utilization ( due the big difference between IIO speed & CPU speed) example: 27 A first cand reader Can read 1200 cands min 1200/00 = 20 Cardy (Sec. The CPU can process 300 conds/ Sec The state of the second second - Card newson, CPU, Cord needer, CPU 60 Sec 4 Sec 1 60 Sec 4 Sec · percentage of Opu utilization - 4/64~1/16~ 6% All Al A Start Mar M. M. Mary W. W. S. Million of Mary 

lecture#2 Feb.12.2018 Monday AN ALLAN CPU. CPU Instruction 110 19 CPU instruction MAN executes until and C. 349 & 110: reaching Ilo 2 24 Millings letc apponting or neudiag input. wild an the M inster of will fait of 11, W 10 [A] offline Operation: ways (1) Before and the martine 1960 card reader input (CPU) output printer and 18281 Weak memory and the to day (2) Aften by Dicely DE YO, SO/ and tapen and tapen and the rest Aape Card nonster panter -memory offline "preperation" offline preperation 1 20 for the star of a spart of the spart Tape to nemory is much more faster Han Courd reader to memory. Simptone execution. An Barris Marine Marine West

**[B]** Buffering OS 100 CPU 1 9 .... Instruction 12 TIO User Sale in 181 -1 printer = program (2) could reader LUN -100 input (3) A 1112 Buffer HILDE Straption - Aut infue 2 S. L. H. M. M. Buffer \* when CPU reaches IIO it brings data. 11M from buffer not from the cond needer 1.900 Conclusions announce in a page 1 11 the ITO of one Job is overlappeal with the execution of some job Sugar Con William autidis 11200 Spooling K of LCI 100 N. M. Bar alars Asy W Server Top drone - 111 Specil "poot" Aneal . 1940 AP and (T) 1 112 , WI WA input device outait device Memory States of

In specifier 2 kinds of data structures were Introduced (1) Job Queue (Job Pool): Aqueue contains the jobs (programs) that demand execution. (2) Spool Areaus which contains the jobs need printing. · Conclusion: The IIO of one job is overlapped with 1 the execution of another job 6 G EDJ Multiprogramming Batch systems a " Multi-programming" 1 Memory is divided into several 0 Regions (Partitions) And Providen 1 Region 1 Regions Sizos normally different. Each region contains only one job. Region 2 \* The CPU switches to another Job when the first one needs IIO. Region 3 11 Keep in mind: Job > Any job (process / program) is a sequence Epi burst of CPU burst & ITO wait and it must start and end with CPU burst. IIO / etc

Two kinds of jobs: (1) CPU-bound job: Contains few very Poney CPU bursts , most of the time, job needs CPU (2) I/O-bound job: Contains Many very short dis at CPU bursts not > most of the time, the job needs I/O 120 400000000 21 Jul Feb. 14. 2018 lecture #3 Payrollet up a Memory mednesday SO E> Jobs Log Ping >CPU Jourets Muttus E CONTON 1002 19doin Cibb 3 for ing april ietc jetc CPU Switches (changes) to another jub when the first one needs IIO "multi-programming" Jobs CPU 2 dob \* Context Switch TIO pritice waiting \* Context switch IIO etc , etc

Uscanded with Transcanner

\* Context Switch: Saves Register for jobs Reloads Register for job2 LE] Time Shannoy Systems: Same as multi-programming, Memory is divided in to requiers. Several jobs are kept in memory. Each job is assigned a slice of time called quantum Q. Each job is executed for this quantum of time of the CPU switches to another jab when quantum Q is finished. In addition to normal corses; 11 \* IS the job needs IIO. \* IF the Job Finishes execution. 1 \* high priority process 1 multi-programming + Time Shannay Ling along & Depundo, The CPU Switches The CPU switches when when the job needs I/O Quantum Q is finished 5 

Uscanded with Camscanner

STUDENTS-HUB.com

[F] Parallel Systems: Multi-processor systems with more than one CPU in dose communication (1) Tightly coupled Systems; processors share memory, dock, & communication usually takes place in nomory so shall and CPU CPU moun nemory Memori -5 \* There are two types of processing: a\_Symmetric multi-processing & Each CPU has the same identical copy of the OS [Reliable of Simple] 6 A Symmetric multi-processing There's a huge OS that runs this scheme There's one CPU called master CPU which controls other activities of other CPUs CP The relation between the master and other CPUs is called marster / stake velationship waster & Reliable on all cases unless the master CPU is faulty.

(2) Loosly Coupled Systems: Networks dilie ile pelli 19 "Distributed system" I Connect viasenvers. Salar 14 May 199 [G] Real time Systems: takes date using an sinsors. - For systems that need Response real time "immediate" of isloditarial an which and a first the stands of the Manager P. Ley and the second AD rhold Mule L apploing a low sector Alexand 101420 april o 2 start aling (R) an 1. Haritan solla deselución and and restors 18 rodensky figling at alles nor the even descended of

0 Chapter#2 17 Computer System operations 1 Device type 2 PR-2 D PR-1 Disk controller Device Controller printer Controller to be Controller Main Bus Controls memory Memory Controller Activities Memory \* IO can run concurrently with the CPU. \* Each device controller is incharge of a particulant device \* Each device controller has allocal buffer. \* CPU moves data from 1 to menin memory. to from the local buffers. \* I 10 is from the device to local buffer of controller. \* Device Centraller informs CPU that it hers finished its operation by coursing an interrupt.

lecture #4 Feb. 17.2018 Saturdery Interrupt (input is model) CPU) Disk Controller 1 word buffer tine your program Instruction Setch, loool another job IR Instruction Reegister. Each device has a device controller that 1 includes alocal buffer. loods > Boot Sector DUD MIT kerneb the kentels "Boot record." en B. A. Sector Norder Man 2attint EBootstop oci doo 1000 address = procyrom influd and any cresh of Much any most i OIT of and allesting. Derating Systems are Interrupt driven man pro ( CPU is interruption)

V RUE Interrupt: A signal sent to the CPU by: "System Call" P 1 Hardware Software "Trap" and in the off 1 100 examples: 100 - Completion of an I/O. "Hardware Interrupt" 1100 Division by Taro, "Software Interrupt" HAR - Invalid memory access. " hard wave Interrupt". UNIT -Request of an OS service. OS services can be asked: (1) System program -> Formal a: 1 count and she se > Copy A. dat B. dat. 2(2) System Callounde OLE ant rollo It's an assembly language Instruction. Setaking of OIT and Land & Completed Et Interrupt Handling: (1) Interrupt vector (Table) -memory Assume that the Interrupt which comes from the hord disk informine the CPU End aconcen the input is completed has number 45 [45] Interrupt vector Interrupt [45] 0 X25164 CPU CPU 1 Service Routine

(2) By pollarge and of the short It I 10 interrupt structure: MM moreorg setch T10 instruction HD Disk Controller It There are two types of I/O: (a) Synchranaus TIO: after the IIO starts, the Control returns to program only upon I/O completion. > The CPU waits until the ILO is completed. A wait Instruction " CPU is idle" > loop: jup loop (b) Asynchronous TIO: 17 all joint month after the IIO starts, the Control Switches to another program without I lo Completion

THE Et Direct Memory Access (DMA): 3 slow I/O deurces such as keyboard, can senel one character every one will second. 1 mils = 1000 mics. 5 The CPU can process the character in 2 micho seconds through an I 5 S lecture #5 Feb. 19. 2018 3 Monday. 1 Mils L Protection of the state of the char 1000 Wics Manard is long of an > Sends 1 Char every 1000 mics. The CPU needs 2 mics for the service noutine to handle the interruption. (CPU) mud 2 2 1000-2 1000 mirs 1000 mics 2000 mics The CPU is left 1000-2-998 mics for Asynchronous I'lo processing. (nothing (bitsurday) SIT get of promision most But, In high speed ID devices (i.e Hard Disk) The HD can send or recieve chair every 4 mics. Chor char Chour 1 4 mics This leaves the CPU with 4-2=2 mics for Hsynchronous multiprocessing.

\* In this case the OS sends one block of data 2 each time of sends an interrupt to the CPU to process this block. SCRU) ALC: NOT Memory Memory 201 100al buffer one block of date 1 Primary Storage (memory\_RAM) "volatile" Memory is largest storage media accessed G041 directly by the CPU. 1012 1 chi words 0591 41 Memory is array of words, each has address. word size = 2 & bytes 3 more Common 4 bytes and the second The CPU performs the following instructions: (a) Load instruction: Fetch (get) instruction 5 from memory to the IR (Instruction Register) it ((b) Store Instruction: storing a Register in to 5 Memory location. 6 5 -

Et Instruction Cycle: store the result CDU Eyer program 3 perform the \* Instruction Cyde \* operations with 1. D. fetch the given operands. Hars IR clocode 100 Analyze the Instruction - MARKA 1. dei Et Secondary Storage: "Hard Disks, topes, CDs, Flash memories. Lin LIFE Factors that affect the secondary Storage: 11) speed (2) Cost (3) Volatility. " ports/ Ciego" 15 Main The Fastest Storage medium in the system: Register Ceiches RAM Shower HD Tapes \* Cache Memory: chashing: is simply copying data to a faster starage medium to speed up execution of ensure gard performance.

-Examplesi (1) Memory (RAM) is considered as a Cache for HD. (2) Registers are considered as Cache for memory. (3) Instruction Carche Register (ICR) (4) (cycle) FARIE PEOGRAM IR 2 2 ICP; 10 & (2) are concurrent Eparelleli]. from ICR to TR is much faster than 1 RAMTOIR -1 Coche is the fastest carche type because it's build with in CPU. -2 bigger but slower. letc lecture #6 Feb. 21.2018 1 Weelinesday 152 (20.7) 1 Et Hardware Protection: Same The How the OS protects the Computer Resources? 5 - Ilo devices adams plantes à printe Memory Memory Marge And - A CPU. Same Jung 5 0

A A 2 N. 10 \* Dual mode of operation: Ly The OS runs in two modes: D (a) monitor (supervisor) mede: In this case the OS B executes process on behalf of it self. (i.e interrupts) B (b) User mode: The OS executes on behallf of the user E it runs user programs]. 7 7 E Implementation; Is one bit is assigned called "made bit" and formal a conta so Or yomentor mode. ×(A) 1: user mode at attacked and to the off of the Death of the Death States ( Definition: montash wish A Privilleged Instruction is an instruction that's executed only in monitor mode. "OS" PROVIDENT LARGE VINTON AND I COMPANY JOSICHICS & E IIO protection: All I 10 instructions are made privilleged Instructions. It Memory Protection: 1 14 Day of Day of 1 and the To protect the memory Allocation space of user program & the OS it self. Mar Barris

OS 15 1-11 Base Roopister User 1 4282 liver Rogister Base Register Unit Dayister 1000 Usera 5262 Base Register base Poorister limit Rogister Usera 2000 Quuit Pogister \* > leaded to the memory \* where user program -(LA) <address starts from 4262 & has a limit of 1000 \* Logical Address (LA): is the offset of the instruction in your bisolum. 2. LA = the address seen in your program \* Physical Hodress, is the actual address in Memory & PA = Logical Address + Base Register. (i.e) PA=736+4262=4998 The most Important concept; is How the OS Compute the PIA? Rimit Register Bose Register Base + limit *(CPU* PA F Memory fourt

Uscanded with Camscanner

STUDENTS-HUB.com

Et CPU protection: Timer 100 10,000 esme Integer Timer + # of Clock ficks, 15 - With every clock tick the timer is decremented by 1 10 when the timer reaches zero it sends an interrupt to 113 15 the CPU. 16 When the CPU recieves interruption it Interruption 111 executes the interruption service routine is service Raitine HE which is responsible for checking the CPU. 110 Timer can be used in computer time Calculation. ALS. 1 [It Operating System Structure! (1) process Manargement. (2) Mensoy memogement. (3) File System (IIO) Management. -The OS services ain be provided in two methods; Usystem Call; it's an assembly language Instruction. R \* System programs, 

Uscanded with Transcanner

Feb. 24.218 lecture #7 Saturday . LAT Process where it want that that the prove they the K. Process Concepter money which it which > Definition: A process is a program in execution Process = program = job .... und is simple wild will will estimate \* There are different process types in the Systems the man of Barn Strong man (1) Batch Processi generally, Batch process has Low priority. C (2) Time sharing process: pharma paray (1) USERS , Much in Bring proving provingers ( - program developement ?? 1 - data entry. genning (3) System tasks 1200 under 24 Interrupt page ustan 1000

\* Process Contents: The process contains: (1) Code Section (Program Counter "PC") "PC"; is the address of the Instruction executed (a) currently. (2) Dates Section: Memory allocation were Input & Output data is stored (3) Stock Section: Gbbal wighter ; Static. it wy 19D attack 100 18 proglam E Carda Section "PC" (9) ALLAST G Stack Section Data Section Process.

Uscanded with Tamstanmer

\* Process States: (1) NEW; The process is newly Cheated acour Sel Neully Creates) " Pool " Procom 2) READY: The process is Looded in to Menning I ready to run and halle the CPU. The process is in READY queue waiting to have the CRU to runs 3) RUNNING: The process is having the CPU & exacution. (4) WAITING; waiting for some event to happen, typically, waiting for I/O. (5) TERMINATED: The process finishes execution. FXIT ou scheduling NEW TERMINATE READY RUNNING hterrup TID Coupleted ILO needed WAITING Major process state.

12 Feb. 26.208 lecture #2 -Monday Pa It where data about the process is stored? -40 > Answer Process Control Block (PCB); 2.5 It's a kind of a data structure, generally it's Pub 10 PCR a table allo Datapl Process ID 115 (1) process ID. Process state 1110 (2) Process State. program Counter (3) program Counter (PC). (4) Registers. (5) Scheduling Info. -(6) Memory Info (Base of limit Registers) (7) Accounting Info 1 Scheduling of System Queues. Quere scheduling Algerathmy (EIFO) 1 Intermolicite queue T. medium Term scheduling P ev scheduling Perilubertos dol (1009) manp dal -Ready quere "Short-Term" (CPU 1/"Long\_Term Finish execution Exit queue is Finished Merrul I 10 is Completeel Ilo is required I/O queve "HD schedulicy Algorithm" HD equeue 

- Epilonat Term "job" schedulingti-It's selecting a job from the job quelle to be admitted to memory and in term it's added to the READY quelle. This selection "Interrupt "is Invoked " Called ", this intoked (seconds, Minites). That is, the OS has enough D time to decide conefully which jub to fetch. "Long-Tenn" R Et Short-Term Schedulings 2 It's selecting a process "jub" from READY queue to be given to CPU to RUN. ( will i seconds, micro seconds, Name seconds), Therefore it should be very fast. "short-Tenu" 1... > Definition > Degree of wullti-programming is the # of jobs 11 in the memory "READY queue". Therefore, 1 city-Term scheduling controls the degree of wullti-programming. -It Job Scheduling: 10 IF most jobs in memory are CPU band gobs, then the CPU will always be busy "high CPU utilization", but the IK queues are empty. On the other hand, If most jobs are I/O bound jobs, then the I/O queues are always full and CPU almost free "Low CPU utilization" (1) In both cases the system is unbalanced! Long\_Term Scheduling: Selects a mix of CPUZ IN bound jobs So the system will be reasonably balanced 

lecture #9 Feb. 28.2018 Wednesday 4 Et Process Creation; 4 \* parent process creates children processes, which, in turn create other processes, forming a Tree of processes, and internation of a contract 2 thermone tool astrong 20 parente INIT > Resource shaning:- Ohilde (1) parent & children share all & parent Cs resources. (2) children shake subset of children 00 powent's resources. (3) Parent & child share no resources "porent & children Compete for All resources" 9 Execution ;- control ( which ) 9 - Parent-gehildren execute concurrently. 3 Parent waits until chibren terminate. Address Space - " Allocation of process in Memory 3 child duplicate of parent. child has approgram loaded into it. Unix examples! Fork system call creates new process exercice system call used after a fork to replace the process werniary space



fork Computation (fork) (CPU) concurrently parallal Process termination :-\* Process executes last statement & asks the operating System to delete it exit output data from child to parent · Via Fork? - process resources are deallocated by OS. when a process is killed all stits resources are deallocated. \* panent may terminate execution of a childnen processes "Abort" a task assigned to child is no A Vonger required a in to mile has exceeded allocated DUE 1 nesources. - Panent is exiting. # OS doesn't allow duild to continue if its powent terminates. # Cascading Tennination: IF the parent process ends exit all children & Subchildren! are exited.

Uscanded with Tamstanner

STUDENTS-HUB.com

Let cooperating processes :-Concurrent processes are either: (1) Independent process: cannont affect or be affected by the execution of another process. (2) Carperating process: Can affect or be affected by the execution of another process. \* Advantages of process cooperation: & Information Shanning 4 Computation Speed -up. > Modulanty man J Convenience a farth i d chilmin Producer Consumer Problem > Cexample on concurrent process interford \* talking about concurrency requires! Cooperating among processes (i.e Communication among processes) I and id synchronication of processes action To illustrate the idea of cooperating process Consider the producer\_consumer problem; producer process produces information. Consumer process consumes this information. without allowing with wat they are

producer Dala Consumer Produces Consumod Buffer Concurrently \* Examples: CI > print program produces characters consumed by the printer. > Complex praduces cissenably Codes Consumed by the assemblers during > assembler produkes marchine language Code Consumed by the Loader Secture # 10 March 3.2018 MULDER JUNISMUSCIES UP ON THE S Saturday ( \* Data Structure Requiredu ( Const int n; USize of Buffer. type item; Il item = chair, bit, word Var int buffer [n]; inting I index where we add items in the buffer int outs' I index where we take items from the buffer, item next P. I next produced item. item nexts; 11 next consumed item. (1) we will use the Circular Buffer idea in the implementation. 32

1 Parts 10 m=7 EG ] E D [5] 100 C 547 Sheel 11 5 B 531 5 7 OUT = 2 A 52 2 > in -1 57 2 10. D Buffertry - definit -) pertucar Data Consumer nextp BUFFERM netc \* Buffer is Full: (in+1)% N= out Circular Buffer R \* Buffer is Empty: in - alt \* Producer Process: Repeart produce an item in next-P. while (in+1) % N = = outno-operations. II busy waiting buffer [im] = nextp;  $in = (in + 1) \% n_{j}$ Until False

Uscanded with Transcanner

\* Consumer process: Repeat while (in=out) ( no operation; Ilbusy waiting Nexte - buffer Lauf]; out- (out,1)%n3 & Consume the item; · UNFIL False Et Consumer Process + Produker Process > Run Concurrently \* Disaduantage: we only use (-n-1) buffers if we have n buffers. 11 A SHING (A + PH) I Alera a gradient V CONSTRUCTION DE LA in the property of the property of rapping and the second second second and the main in the second of the second sec his huffing I wind Reptort -Stranger and anthread outbours the second provide a provide the state of the DIStinu vout // (fina) privition ols

55 [1] Threads: 3 5 Process (Hearney Weight Process) 2 I code section ! POCESS TD Stack Section! , Ragisters Set Data section Program Gunter Corde Section Cata Section opened files Man Kill Registerest PTD stack settion DO HIPPHIL LECHURY DUIS AND STORING OF Y Thread (Light Weight Process) Contains: Program Counter. Registers set. Stack Section. Manual and and and All peer threads; Shake, Coole section - 11 10 10 days . Data Section. T/O Resources.

Carda Section Data setion morred Files all ID Devised Stack Agista Jack. p \* advantage of threads is: Shannoy Resources. lecture # 11 March 5-208 6 Mondery 6 It There are two kinds of thread support:-GA (1) user lovel: It's totally the responsibility 62 of the user (very Complex & difficult) G 2) at kernel lovel, most OS support this! 6 kind of threads. 6 6 \* Relation ship between user threads & kernel threads; 6 Think about user threads of processes (programs) ( & Think about Kemel threads as CPUs 6

\* we have > multi-threads. > MULIFI- CPUS. 10 F (1) Many - To cone: 2. Many and 1 -Many user threads are welpeal to one kernel thread (B) Kornels thread 10 and the tout luser Horeads. Disadvantage: NO concurrent execution. (2) One - To - One -Each kernel thread is assigned to one user thread. 5 K) { kernel threads. Juser Homeorels n talent Main advantige: This allows concurrent execution. T ! Dis advantage; we need enough kernel threads 7 1 (ス) Many To Many: Many user threads are mapped to an equal or less number of ternel threads. (K)