47.5/52 + 5 ⇒ 52/52

NAME Evan Bonsignori

Student-ID 907 190445 (old)
901021190

CSCI
Mid-Term Exam
Time: 75 minutes

1- Under what circumstances a process will be eliminated? (5 pts.)

1. It has successfully completed/Finished Execution
2. Parent of child kills child. Reasons could be:
   (i) Child overstepped boundaries in usage of resource
   (ii) Task child is working on has already been completed or the child is no longer needed.

+5

3. Cascading Termination/Cancellation; ~~deleting~~ terminating a parent process causes children to also terminate/cancel.

2- Define OS, and compiler (2 pts. each)

1. OS: Collection of data and programs in charge of managing computer resources & supporting virtual characteristics of computer.

+2

2. Compiler: Compiles entirety (all lines) of a source program into a ~~native machine's machine code/language.~~

??

???

+2 +0

3- What are the components of Java virtual machine and tasks they perform? (5 pts.)

(1) Class loader: Loads source program bytecode to JVM on the native machine.

+1

+ 5

(2) Class verifier: Verifies that the bytecode will not harm the native machine. what type of harms?

+2

(3) JVM Interpreter: Can act as 2 cases:

(i) Acts as an interpreter; Compiles one bytecode at a time into machine code and feeds each one by one to the native machine to handle.

(ii) Acts as a compiler: compiles entire program (all bytecodes) into one program in machine code, and feeds the entire thing to native machine to handle.

+14

4- Itemize the advantage(s) and disadvantage(s) of:

a) Queueing model approach for evaluating CPU scheduling algorithms (3 pts.)

b) One-to-one multi-threading model (3 pts.)

c) Simple, layered, and virtual machine operating system structures (8 pts.)

(a) Advantages: (i) Uses dynamic data (This is a better model of reality than static)

Disadvantages: (i) Uses complex and often difficult to interpret mathematical models that aren't always viable to implement.

+3 (ii) Makes assumptions in order to create a closed form model. For example, when modelling the distribution of CPU bursts to derive the formula $n = \lambda * w$ (length of ready queue = Avg CPU burst * waiting) we assume that the rate that jobs entering the queue is the same as jobs leaving the queue, which leads to a closed form (but imperfect) equation.

(b) Advantages: Achieves Great Parallelism +3

Disadvantages: High overhead costs (user threads mapped direct to each kernel thread)

(c) (1) Simple:

Advantages: Simplicity / Easy to implement because no planning need.

Disadvantages: Difficult to expand w/o errors because it's one big program where all

+1.5 functionality is coupled. This makes it inflexible, error prone, and difficult to follow at the code level for debugging. The MS-DOS system for example, eats into the command interpreter RAM when a new process begins.

(2) Layered:

Advantages: (i) Modularization (Layers are modularized to functionality in each layer)

(ii) Error Containment: (If an error happens when using layer k, it is handled in layer(k))

(iii) Easier to write OS (than simple)

(iv) Easier to debug OS (bc Error Containment)

+4 (V) Easier to Expand (bc we can use functionality of previous layers, and layers are logically & clearly separated)

Disadvantages: (i) Can't access layers above - this makes deciding what functionality should go into each specific layer more difficult.

(ii) Inefficiency - A layer must access previous layers to do anything

(3) Virtual Machine

Advantages: (i) Protection (Perfect if no communication /sharing between Users

(ii) Great research vehicle for OS bc it provides an OS environment to test (i.e. CPU sched Algo's).

Disadvantages: (i) Lack of sharing /communications between Users

Solutions to this disadvantage:

+2.5

(1) Treat every user as node on network & nodes can send msg's to each other

(2) Use a shared mini-disk to store communications on.

5- A set of jobs along with their arrival and CPU burst times are given below.

Job	CPU Burs time	Arrival time
J1	7	0
J2	4	0
J3	1	2
J4	3	2
J5	2	4
J6	3	4
J7	1	5

+10

Calculate the average waiting time for the following algorithms using the above set of jobs:

a. Preemptive SJF (5 pts.)
b. Round Robin (time slice = 3) (3 pts.)
c. Preemptive Round Robin (time slice = 5) (5 pts.)

Show **all** of your work **neatly**; otherwise, you receive score of zero.

(a)

Job	CPU Remain	
J1	~~7~~, ~~0~~	0
J2	~~4~~, ~~3~~, ~~1~~, 0	0
J3	~~1~~, 0	2
J4	~~3~~, 0	2
J5	~~2~~, 0	4
J6	~~3~~, 0	4
J7	~~1~~, 0	5

(a) Job Queue (From Right to Left):

Please follow the format used in class

← 11 | J4 | 8 | J5 | 6 | J7 | 5 | J2 | 4 | J2 | 3 | J3 | 2 | J2 | 0

cont 21 | J1 | 14 | J6 ← cont

?

~~we didn't discuss this case for SJF in Preemptive RR, the next job J6 would get priority, however depending on if J4 may get it if for loop select it first. For F.S.T.F~~

Average Waiting Time Per Job

J1: 21 - 0 - 7 = 14
J2: 5 - 0 - 4 = 1
J3: 3 - 2 - 1 = 0
J4: 11 - 2 - 3 = 6
J5: 8 - 4 - 2 = 2
J6: 14 - 4 - 3 = 7
J7: 6 - 5 - 1 = 0

+5

=> Overall Average Wait time = (14 + 1 + 6 + 2 + 7 + 0 + 0) / 7

(a) = $\boxed{30/7 \text{ time units}}$

(b)

J1	~~7~~, ~~4~~, ~~1~~, 0	0
J2	~~4~~, ~~1~~, 0	0
J3	~~1~~, 0	2
J4	~~3~~, 0	2
J5	~~2~~, 0	4
J6	~~3~~, 0	4
J7	~~1~~, 0	5

Job Queue (R 2 L):

← 15 | J6 | 12 | J5 | 10 | J4 | 7 | J3 | 6 | J2 | 3 | J1 | 0

21 | J1 | 20 | J2 | 19 | J1 | 16 | J7 ←

Average Waiting Time / Job:

J1: 21 - 0 - 7 = 14 } 30
J2: 20 - 0 - 4 = 16
J3: 7 - 2 - 1 = 4
J4: 10 - 2 - 3 = 5
J5: 12 - 4 - 2 = 6
J6: 15 - 4 - 3 = 8
J7: 16 - 5 - 1 = 10

+3

=> Average Waiting Time for RR (b) = (14 + 16 + 4 + 5 + 6 + 8 + 10) / 7 = $\boxed{63/7}$

(c) FRR

J1	~~7~~, 2	0
J2	~~4~~, ~~2~~, ~~1~~, 0	0
J3	~~1~~, 0	2
J4	~~3~~, 0	2
J5	~~2~~, ~~1~~, 0	4
J6	~~3~~, 0	4
J7	~~1~~, 0	5

TS = 5

Job Queue:

5 | J2 | cpu burst for J2 = 1

← | J2 | 6 | J7 | ~~5~~ 6 | ~~J5~~ J7 | 4 | J2 | 3 | J3 | 2 | J2 | 0 ← Start

TS=6 TS=3 TS=4 TS=5 TS=5 TS=6 TS=3 TS=3

21 | J1 | 19 | J1 | 14 | J4 | 11 | J6 | 8 | J5 | 7 ←

Avg Waiting Time:

J1: 21 − 0 − 7 = 14
J2: 7 − 0 − 4 = 3
J3: 3 − 2 − 1 = 0
J4: 14 − 2 − 3 = 9
J5: 8 − 4 − 2 = 2
J6: 11 − 4 − 3 = 4
J7: 6 − 5 − 1 = 0

⇒ Average Waiting Time =

$$\frac{(14 + 3 + 0 + 9 + 2 + 4 + 0)}{7} = \boxed{\frac{32}{7}}$$

~~+3~~ + 2   26 → 32

+11.5

6-Compare and itemize the difference(s) between: (Just point out the differences between any given to concepts. If you only provide definitions of the concepts and no clearly pointed out the differences you receive score of zero regardless of your answer.)

LT = long term sheduler
ST = short term sheduler

a) Long term scheduler and short-term scheduler. (4 pts.)
b) Thread and process (4 pts.)
c) Hard and soft real time Operating systems (2 pts.)
d) Relocatable and absolute modules. (3 pts.)

(a) (i) LT shedules programs on hard disk to be fed to RAM, ST sheduler processes on RAM in ready queue to be fed to CPU +1
+1 (ii) frequency of making a ranking decision is greater for ST than LT
+1 (iii) Time taken to make a ranking decision is shorter for ST than LT

(b) ✓(i) Thread shares data, code, and resource sections
✓(ii) Context switching for a thread is faster
✓(iii) A thread is smaller
✓(iv) Protection isn't needed amongst peer threads since they work together to hurt one task.
(V) I/O is faster for threads because files don't need to be closed & reopened between peer threads working on same table
The Same
✓(VI) Threads allow a sequential program to run in parallel.
+3.5

(c) (i) Hard R.T. guarentees that processes execute/finish in real time, there is no such guarentee for soft R.T. OS's. +2
???

What is a fixed module?
(d) (i) Fixed modules have fixed position/addresses in the memory space, when relocatable modules can be move/relocated to a different address space in memory.
+3