NATIONAL UNIVERSITY OF SINGAPORE

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ONLINE EXAMINATION

Matriculation No.:	A0224725H
Module Code:	EE 5903
Number of pages in this PDF file (including this cover page and Declaration Form): i.e. 2+no. of answer pages	7

INSTRUCTIONS TO CANDIDATES

1. Follow the instructions for online examination and invigilation.
2. Write your answers on A4 size paper with black or dark blue ink.
3. Write the question number at the top left corner of each page. Start the answer to each question on a new page. Indicate the part, e.g. "(a)", on the left margin.
4. At the end of the exam:
   a) scan or take photographs of your answers (make sure your writing and/or drawings can be seen clearly);
   b) enter your matriculation number, module code and the total number of pages (including the cover and declaration pages, i.e. 2+number of answer pages) on the cover page;
   c) merge the completed cover page, signed declaration form and your answers into a single PDF file named `<matric_no>-<module code>.pdf` (e.g. `A1234567R-EExxxx.pdf`)
   d) open the PDF file to ensure that it has been generated without error and the contents are correct;
   e) upload your PDF file into the stated LumiNUS exam submission folder within the stipulated deadline. Late submissions will not be accepted.

FOR OFFICE USE ONLY

Question	Mark	Remarks
TOTAL		

Exam Declaration Form

Please read sections A, B and C below. Sign and submit this declaration form together with your answers.

A. Academic, Professional and Personal Integrity

1. *The University is committed to nurturing an environment conducive for the exchange of ideas, advancement of knowledge and intellectual development. Academic honesty and integrity are essential conditions for the pursuit and acquisition of knowledge, and the University expects each student to maintain and uphold the highest standards of integrity and academic honesty at all times.*
2. *The University takes a strict view of cheating in any form, deceptive fabrication, plagiarism and violation of intellectual property and copyright laws. Any student who is found to have engaged in such misconduct will be subject to disciplinary action by the University.*
3. *It is important to note that all students share the responsibility of protecting the academic standards and reputation of the University. This responsibility can extend beyond each student's own conduct, and can include reporting incidents of suspected academic dishonesty through the appropriate channels. Students who have reasonable grounds to suspect academic dishonesty should raise their concerns directly to the relevant Head of Department, Dean of Faculty, Registrar, Vice Provost or Provost.*

B. I have read and understood the rules of the assessments stated below:

a. *Students should attempt the assessments on their own. There should be no discussion or communication, via face to face or communication devices, with any other person during the assessment.*

b. *Students should not reproduce any assessment materials, e.g. by photography, videography, screenshots, copying down of questions, etc. Posting on public forums, e.g. social media and websites, is prohibited.*

C. I understand that by breaching any of the rules above, I would have committed offences under clause 3(I) of the NUS Statute 6, Discipline with Respect to Students, which is punishable with disciplinary action under clause 10 or clause 11 of the said statute.

3) Any student who is alleged to have committed or attempted to commit, or caused or attempted to cause any other person to commit any of the following offences, may be subject to disciplinary proceedings:
(I) plagiarism, giving or receiving unauthorised assistance in academic work, or other forms of academic dishonesty.

I have read and will abide by the NUS Code of Student Conduct (in particular, (A) Academic, Professional and Personal Integrity), B and C when attempting this assessment.

Signature: LUO ZIJIAN　　Date: 5th May, 2021

Matric. No.: A0224725H

Q1

(i) real-time

(ii) $\underset{0}{|}\, J_1 \,\underset{2}{|}\, J_2 \,\underset{5}{|}$ $f_1=3, f_2=5$, $d_1=1, d_2=3$, so $L_1=2$, $L_2=2$ $L_{max}=max(L_i)=2$
In opposite direction, $L_{max}'=4$, it is not optimal minimum.

(iii) the ready queue would converge to first come first serve (FCFS)
In this time quantum, ~~the order of~~ tasks in ready queue are allocated each same time quantum orderly (FCFS)

(iv) The best decision is roll back.
If we detect a fault, we should roll back each step to arrive the safe design, and rearrange these remaining part to satisfy the deadline

(v) (a), (d), (b)

(vi) $R(t)=e^{-1.6\times10^{-3}t}$, $\lambda=\sum_{i=1}^{n}\lambda_i = 4\times10^5 \times 4\times10^{-9} = 1.6\times10^{-3}$, $f(t)=\lambda \cdot e^{-\lambda t}$
$R(t)=\int_t^{\infty}\lambda e^{-\lambda t}dt = e^{-\lambda t} = e^{-1.6\times10^{-3}t}$

(vii) If the token of a task is not the smallest token, this process will be made to wait

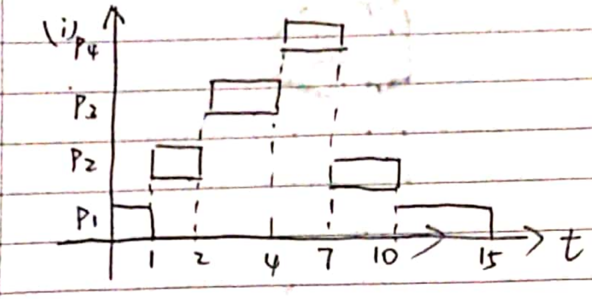
(viii) ~~TRUE~~ FALSE, turn = i

(ix) Availability: $\frac{1}{8767}$. because Availability $=\frac{MTBF}{MTBF+MTTF}=\frac{1}{1+8766}=\frac{1}{8786}$

(x) $P_1(t=5)=0$, $P_2(t=5)=\frac{1}{2}$, $P_3(t=5)=\frac{1}{2}$
Thus the total load $P(t=5)=\frac{1}{2}$

Q2

(i)



t=0, only P_1 is executed

t=1, P_2 preempt, P_1 is left (P_2 arrive at 1

t=2, P_3 preempt, P_2 is left (P_3 arrive at 2)

~~t=3, P_1 preempt, P_3 over (P~~

t=4, P_3 is over, the best next process is P_4

t=7, P_4 is over, next process is P_2

t=10, P_2 is over, next process is P_1

For individual waiting time: $W_1 = 10-1+0-0 = 9$

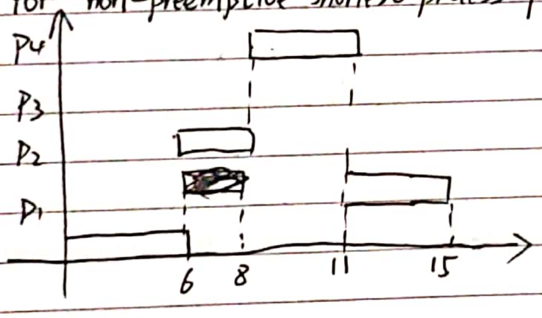
$W_2 = 7-2+1-1 = 5$

$W_3 = 2-2 = 0$

$W_4 = 4-3 = 1$

Therefore, the total waiting time (AWT) $= \frac{9+5+0+1}{4} = 3.75$

Q2 (ii) For non-preemptive shortest process first,



t=0, only P_1 is ready

t=6, P_1 is over, the best one is P_3

t=8, P_3 is over, the best process turns P_4

t=11, P_4 is over, next one is P_2

$W_1 = 0-0=0$

$W_2 = 11-1=10$

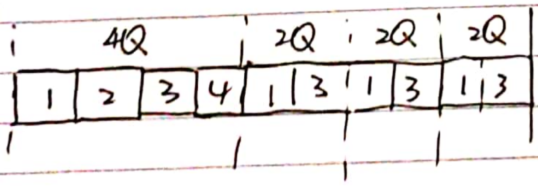
For individual waiting time $W_3 = 6-2=4$

$W_4 = 8-3=5$

$AWT = \frac{0+10+4+5}{4} = 4.75$

I found Preemptive shortest process first perform better than Non-preemptive one, especially it reduce a lot in average waiting time.

Q2 (iii)

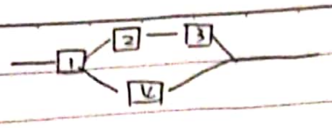


Firstly, 4Q is allocated for four process. P_2, P_4 are over

And then share 2Q for P_1 and P_3 until P_1 is over.

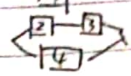
Therefore, the completion time of P_1 is $4+2+2+1 = 9$

Q3.

(i) 

In this part, we call [2]—[3] up part

[4] down part

 right part

$F_1 = 0.08 \quad F_2 = 0.3 \quad F_3 = 0.2 \quad F_4 = 0.1$

$R_1 = 0.92 \quad R_2 = 0.7 \quad R_3 = 0.8 \quad R_4 = 0.9$

$R_{up} = R_2 \cdot R_3 = 0.7 \times 0.8 = 0.56 \qquad F_{up} = 1 - 0.56 = 0.44$

$R_{down} = R_4 = 0.9 \qquad F_{down} = 0.1$

$R_{right} = 1 - (1 - R_{up})(1 - R_{down}) = 1 - (1 - 0.56)(1 - 0.9) = 0.956$

$F_{right} = 0.44 \times 0.1 = 0.044$

$R = R_1 \cdot R_{right} = 0.87952$

$F = 1 - R = 0.12048$

Q3 (ii) suppose first process failure rate is λ_1

second process: $\lambda_2 = \frac{1}{2}\lambda_1$

third process: $\lambda_3 = \frac{1}{4}\lambda_1$

fourth process: $\lambda_4 = \frac{1}{8}\lambda_1$

fifth process: $\lambda_5 = \frac{1}{16}\lambda_1$

$\lambda_1 + \frac{1}{2}\lambda_1 + \frac{1}{4}\lambda_1 + \frac{1}{8}\lambda_1 + \frac{1}{16}\lambda_1 = 8.0 \times 10^{-5}$

$\lambda_1 = 4.13 \times 10^{-5}\, h^{-1} \qquad \lambda_2 = 2.06 \times 10^{-5}\, h^{-1} \qquad \lambda_3 = 1.03 \times 10^{-5}\, h^{-1}$

$\lambda_4 = 5.15 \times 10^{-6}\, h^{-1} \qquad \lambda_5 = 2.58 \times 10^{-6}\, h^{-1}$

Overall failure of this pipeline system: $1 - e^{8.0 \times 10^{-5} t}$

Q3 (iii) (a) Overall reliability $\prod_{k=1}^{n} P_x(k)$

(b) if components are assumed to be independent,

$$1 - \prod_{k=1}^{n}\left[1 - P_x(k)\right]$$

Q5

(i) To solve Producer-Consumer problem,
the basic functions used : wait ( ), signal ( )
wait ( ) : means it detect the status of ~~deadlock~~ lock, if it is no deadlock,
this process can ~~let~~ critical section into.
signal ( ) : means it finish the part of critical section, it can release the ~~dead~~ lock, and then the process finish remaining ~~part~~ section.

The persudo code:

```
do {
        wait (mutex);
                critical section;
        signal (mutex);
        remainder section;
} while (TRUE)
```

Q5 (ii)

```
void producer () {
    while (TRUE) {
        x = produce ();    // produce something (X)
        wait (B) ;    // it detect overflow
        wait (A) ;    // it can access buffer
        append (x) ;    // X into buffer
        signal (A) ;    // producer, release access into buffer
        signal (U) ;    // producer ~~detect~~ send underflow
    }
    return
}
```

```
void consumer () {
    while (TRUE) {
        wait (U) ;    // consumer ~~detect~~ underflow
        wait (A) ;    // consumer get access into buffer
        x = read () ;    // get something (X)
        signal (A) ;    // consumer release access into buffer
        signal (B) ;    // consumer send buffer overflow
        consume (x) ;    // consume X
    }
    return
}
```

I use these variables in my design, the parameters are set here.

A = 1   // control buffer access
U = 0   // control buffer underflow
B = buffer size   // avoid buffer overflow

In producer procedures : ~~if A=1, it m~~
if B < buffer size , means it can produce more
and then if A=1 , means it can get access into this buffer

A0224725H

5

Q5(ii)

In consumer procedures:

if $U>0$, means it can consume one

and then if $A=1$, means it can get access into this buffer.

Q5(iii)

One major drawback: always waiting ~~other~~ each other

a way: activating one object firstly

FALCON

扫描全能王 创建