USN .				

Internal Assessment Test III – AUG 2022

Sub:		MICRO	OCONTROLLE	R AND EMBEDDED SYSTEM Sub Code:	18CS44	Branc	h CSE		
Date:	:	29/08/22	Duration :	90 mins Max Marks: 50 Sem / Sec: IV Sem A/B/		em A/B/C		C	DBE
				Answer any FIVE FULL Questions			ARKS	СО	RBT
1	sys So Qu pro in Th 1. 2. 1. Th the em Th a. b. c. d. e. f. S Th a. b. c. d.	stems. lution: Julity attribute operly any system de e Quality attr Operational C Non-Operatio e operational e hbedded syste e important o Response Throughput Reliability Maintainabili Security Safety	es are the notes and the notes of an Quality Attributed Quality Attributed when it in Quality attributed with the notes of the Quality attributed with the notes of the notes	Attributes outes ibutes represent the relevant quality attributes represent the relevant quality attributes in the operational mode or online mode. outes are onal Attributes quality attributes are y	documented namely		[10] [5] [5]	CO 1	L1
2	T	Ź	•	e fundamental issues in hardware-softwa	· ·	-	[10] [6]	CO 1	L2,L3 Inter nal Asse
	1. th	oftware co-d . Selecting t	lesign. he model - aracteristic	System models are used for capturing a s. A model is a formal system consisting	and describing	5	[4]		ssme nt Test III – AUG

It is hard to make a decision on which model should be followed in a particular system design. Most often designers switch between a varieties of models from the requirements specification to the implementation of the system design.

2. Selecting the Architecture - A model only captures the system characteristics and does not provide information on how the system can be manufactured?

The architecture specifies how a system is going to implement in terms of the number and different types of components and the interconnection among them.

Some of the commonly used architectures in system design are

- **i.** The controller architecture implements the finite state machine model using a state register and two combinational circuits. The state register holds the present state and the combinational circuit's implement the logic for next state and output.
- **ii.** The datapath architecture is best suited for implementing the data flow graph model where the output is generated as a result of a set of predefined computations on the input data.

A datapath represents a channel between the input and output and in datapath architecture the datapath may contain registers, counters, register files, memories and ports along with high speed arithmetic units.

iii. The Finite State Machine Datapath (FSMD) – this architecture combines the controller architecture with datapath architecture. It implements a controller with datapath.

The controller generatesthe control input whereasthe datapath processesthe data. The datapath contains two types of I/O ports, out of which one acts as the control port for receiving/sending the control signals from/to the controller unit and the second I/O port interfaces the datapath with external world for data input and data output.

iv. The Complex Instruction Set Computing (CISC) - architecture uses an instruction set representing complex operations. It is possible for a CISC instruction set to perform a large complex operation with a single instruction.

The use of a single complex instruction in place of multiple simple instructions greatly reduces the program memory access and program memory size requirement.

v. The Very Long Instruction Word (VLIW) - architecture implements multiple functional units (ALUs, multipliers, etc.) in the datapath. The VLIW instruction packages one standard instruction per functional unit of the datapath.

2022

Sub:

MIC ROC

ONT ROLL

ER AND

EMB

EDD FD

SYST

EM Sub

Code .

18CS 44

Bran

ch:

CSE

Date .

29/0 8/22

Dura tion:

90 mins Max Mar

ks: 50 Sem

/ Sec:

IV Sem

A/B/ C

OBE

vi. Parallel processing architecture - implements multiple concurrent Processing Elements (PEs) and each processing element may associate a datapath containing register and local memory.

Single Instruction Multiple Data (SIMD) and Multiple Instruction Multiple Data (MIMD) architectures are examples for parallel processing architecture.

In SIMD architecture, a single instruction is executed in parallel with the help of the Processing Elements. On the other hand, the processing elements of the MIMD architecture execute different instructions at a given point of time.

3. Selecting the language - A programming language captures a Computational Model and maps it into architecture. Any programing language can be used.

A model can be captured using multiple programming languages like C, C++, C#, Java, etc. for software implementations and for hardware implementations languages like VHDL, System C, Verilog, etc.

On the other hand, a single language can be used for capturing a variety of models. Certain languages are good in capturing certain computational model. For example, C++ is a good candidate for capturing an object oriented model.

- **4. Partitioning System Requirements into hardware and software -** From an implementation perspective, it may be possible to implement the system requirements in either hardware or software (firmware). It is a tough decision making task to figure out which one to opt. various hardware software trade-offs are used for making a decision on the hardware software partitioning.
- b) Explain different communication buses used in automotive applications
 Automotive applications make use of serial buses for communication, which
 greatly reduces the amount of wiring required inside a vehicle.

Different types of serial interface buses deployed in automotive embedded applications are

1. Controller Area Network (CAN) - The CAN bus was originally proposed by **Robert Bosch**. It supports medium speed with data rates up to 125 Kbps and high speed with data rates up to 1Mbps data transfer.

CAN is an event-driven protocol interface with support for error handling in data transmission. It is generally employed in safety system like airbag control; power train systems like engine control and Antilock Brake System (ABS); and navigation systems like GPS etc.

2. Local Interconnect Network (LIN) - LIN bus is a single master multiple slave (up to 16 independent slave nodes) communication interface. LIN is a

Ans wer any FIVE FULL Ques tions

MAR KS

CO

RBT

1

Disc uss the oper ation al and nonoper ation al quali ty attri bute s of an **Emb** edde d syste ms.

Solut ion:

Qual ity attri bute low speed, single wire communication interface with support for data rates up s are to 20 Kbps and is used for sensor/actuator interfacing. the nonfunct LIN bus is employed in applications like mirror controls, fan controls, seat ional positioning controls, window. Controls, and position controls where response requi time is not a critical issue. rem ents 3. Media-Oriented System Transport (MOST) Bus - The Media-oriented that system transport (MOST) is targeted for automotive audio/video equipment need interfacing, used primarily in European cars. to be docu men A MOST bus is a multimedia fibre-optic point-to-point network implemented ted in a star, ring' or daisy chained topology over optical fibre cables. prop erly The MOST bus-specifications define the physical layer as well as the application layer, network layer, and media access control. in any MOST bus is an optical fibre cable connected between the Electrical Optical syste Converter (EOC) and Optical Electrical Converter (OEC), which would m translate into the optical cable MOST bus. desig n. The Qual ity attri bute s of any emb edde d syste m are class ified into two, nam ely

> 1. Oper ation al Qual ity Attri bute s

		2.
		Non-
		Oper
		ation
		al
		Qual
		ity
		Attri
		bute
		S
		.
		1.
		Oper
		ation
		al
		Qual
		ity
		Attri
		bute
		S
		The
		oper
		ation
		al
		quali
		ty
		attri
		bute
		S
		repr
		esen
		t the
		relev
		ant
		quali
		ty
		attri
		bute
		S
		relat
		ed to
		the
		emb
		edde
		d
		syste
		m
		whe
		n it
		is in
		the
		oper

	 _
	ation
	al
	mod
	e or
	onlin
	е
	mod
	e.
	The
	impo
	rtant
	quali
	ty
	attri
	bute
	s are
	a.
	Resp
	onse
	31.30
	b.
	Thro
	ughp
	ut
	C.
	Relia
	bility
	d.
	Main
	taina
	bility
	e.
	Secu
	rity
	'
	f.
	Safet
	У
	The
	impo
	rtant
	Non
	Oper
	ation
	al Attri
	bute
	S

	quali ty attri bute s are
	a. Testa bility & Deb ug-a bility
	b. Eval uabil ity
	c. Port abilit y
	d. Time to prot otyp e and mark
	e. Per unit and total cost.
	[10]
	[5] [5]
	CO1
	L1

2 Exp	
	- 1
Exp	- 1
	nl l
air	
the	
fun	
am	
nta	
issu	
sir	
har	
war	
-sot	
war	
esig	Ru
The Court	
foll	
win	ng
issu	
s ar	
son	
e o	
the the	
fun	
nta	
issu	
sir	
har	
war in the second secon	
sof	
war	
esig	gn
Sele	
ting	g
the the	
el- Svet	
Syst	
els	
are	
use	
for some	
сар	Jί

		ι	uring
			and
			desc
			ribin
		£	g the
		!	syste
			m
			char
			acter istics
		'	. A
			mod
			el is
			a
		1	form
			al
			syste
			m
			consi
		5	sting of
			obje
			cts
			and
			com
			posit
			ion
			rules
			.
			It is
			hard
			to
			mak
			e a
			decis
			ion
			on
		'	whic
			h mod
			mod el
			shou
			ld be
			follo
			wed
			in a
			parti
			cular
		!	syste
			m desig
		'	desig n.
			Most

	г	
		ofte
		n
		desig
		ners
		switc
		h
		betw
		een
		a
		varie
		ties of
		mod
		els
		from
		the
		requi
		rem
		ents
		speci
		ficati
		on
		to
		the impl
		eme
		ntati
		on of
		the
		syste
		m
		desig
		n.
		2.
		Selec
		ting the
		Archi
		tectu
		re -
		Α
		mod
		el
		only
		capt
		ures
		the
		syste
		m
		char acter
		istics
		and
		۵۵

	does
	not
	provi
	de
	infor
	mati
	on
	on
	how
	the
	syste
	m
	can
	be
	man
	ufact
	ured
	?
	_,
	The
	archi
	tectu
	re .
	speci
	fies
	how
	a
	syste
	m is
	goin
	g to
	impl
	eme
	nt in
	term
	s of
	the
	num
	ber
	and
	diffe
	rent
	type s of
	com
	pone nts
	and
	the
	inter
	conn
	ectio
	n
	amo

		ng
		the
		m.
		Som
		e of
		the
		com
		monl
		У.
		used
		archi
		tectu
		res
		in
		syste
		m l
		desig
		n are
		i.
		The
		cont
		rolle
		r
		archi
		tectu
		re -
		impl
		eme
		nts
		the
		finite
		state
		mac
		hine
		mod
		el
		usin
		g a
		state
		regis
		ter
		and two
		com
		binat
		ional
		circu
		its.
		The
		state
		regis ter
		ıcı

 ·	
	hold
	s the
	pres
	ent
	state
	and
	the
	com
	binat
	ional
	circu it's
	impl
	eme
	nt
	the
	logic
	for
	next
	state
	and outp
	ut.
	ii.
	The
	data
	path
	archi
	tectu
	re - is
	best
	suite
	d for
	impl
	eme
	nting
	the
	data
	flow
	grap h
	mod
	el
	wher
	e the
	outp
	ut is
	gene
	rate
	d as a
	resul
	1.0301

	t of a
	set
	of
	pred
	efine
	d
	com
	puta
	tions
	on
	the
	inpu
	t
	data.
	A
	data
	path
	repr esen
	ts a
	chan
	nel
	betw
	een
	the
	inpu
	t and
	outp
	ut
	and
	in
	data
	path
	archi
	tectu
	re
	the
	data path
	may
	cont
	ain
	regis
	ters,
	coun
	ters,
	regis
	ter
	files,
	mem
	ories
	and
	 port

	S
	alon
	g
	with
	high
	spee
	d
	arith
	meti
	С
	units
	iii.
	The
	Finit
	e
	State
	Mac
	hine
	Data
	path
	(FSM
	D) –
	this
	archi
	tectu
	re
	com
	bine
	s the
	cont
	rolle
	r
	archi
	tectu
	re
	with
	data path
	archi
	tectu
	re. It
	impl
	eme
	nts a
	cont
	rolle
	r
	with
	data
	path.
	[']
	The

		cont
		rolle
		r
		gene
		rates
		the
		cont
		rol
		inpu
		t
		wher
		east
		he
		data
		path
		proc
		esse sthe
		data.
		The
		data
		path
		cont
		ains
		two
		type
		s of
		1/0
		port
		s,
		out
		of
		whic h
		one
		acts
		as
		the
		cont
		rol
		port
		for
		recei
		ving/
		send
		ing
		the
		cont
		rol
		signa Is
		from
		/to
		the

controller of co		
rolle r unit and the seco nd l/O port inter faces the data path with exter nal worl d for data inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com put ng (CISC) - archi tectu re uses an instr uttio n set repr esen ting com plex ex e		cont
r unit and the seco ned l/O port inter faces the data path with exter nal world for data inpu t and data outp ut. iv. The Com plex instruction set Com put in garage (CISC On Set Com put in garage) in set com put in garage (CISC On Set Com put in garage) in set ceture resures an instruction set representing com plex in set complex in set ceture resures an instruction set in set ceture resures an instruction set in set ceture representing complex in set complex in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set Complex in set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in set in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in garage) in garage (CISC On Set In set in		
and the seco of decomposition of the second		r
the seco and I/O port interfaces the data data data data data data data dat		
seco and I/O port inter faces the data path with exter nal worl d for data inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC) - archi tectu re uses an instr uctio n set repr esen ting		
nd I/O port inter faces the data path with exter nal worl dfor data inpu t and data outp ut. iv. The Com plex Instruction set of the Computing (CISC) - architecture exters and instruction set of the Computing (CISC) - architecture exters and instruction set of the Computing (CISC) - architecture exters and instruction set of the Computing (CISC) - architecture exters and instruction set of the Computing (CISC) - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction set of the CISC - architecture exters and instruction exters and inst		
I/O port inter faces faces the data path with with exter nal land data inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com put ng (CISC) - archi tectu re uses an instr uctio n set repr esen ting com plex plex		
port interfaces the data path with exter nal world for data input tand data output. iv. The Complex Instruction Set Computing (CISC) - architecture results and instruction set output in the computing set output in the computing set output in the computing set output in the country in the c		
inter faces the data path with exter nal worl d for data inpu t and data outp ut. iv. The Com plex instruction set Com puting (CISC) - architecture e uses an instruction set representing com plex per set representing com plex opex oper essenting com plex opex oper essenting com plex opex oper essenting com plex oper		
faces the data path with exter nal worl d for data inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC) - archi tectu re uses an instr uctio n set repr esen ting com plex		
the data path with exter nal worl d for data inpu t and data outp ut. iv. The Com plex Instruction is set Com puting (CISC) - architecture uses an instruction set representing to set representing com plex oper sesenting com plex oper oper sesenting com plex oper plex plex plex plex plex plex plex plex		faces
data path with exter nal worl d for data input 1 and data output. iv. The Com plex Instruction Set Com putting (CISC)) - architecture re uses an instruction set representing com plex oper oper instruction set representing com plex oper oper oper instruction set representing com plex oper oper oper oper oper oper oper oper		
path with exter nal world of for data input and data output, ut. iv. The Complex Instruction Set Computing (CISC) - architecture uses an instruction set exter uction set exter uction set exter uction set exter use uses an instruction set exter uses an instruction set external uses and instruction		
with exter nal worl d for data input and data outp ut. iv. The Com plex Instruction set Com puting (CISC) - architecture uses an instruction set external e		
exter nal worl d for data inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC)- archi tectu re uses an instr uctio n set repr essen ting com plex		with
worl d for data input and data output. iv. The Complex lnstruction Set Computing (CIS) architecture reuses an instruction set representing complex oper		exter
d for data inpu t and data outp ut. iv. The Com plex Instruction n Set Com puting (CISC))- architecture e uses an instruction n set representing com plex opper		
data inpu t and data outp ut. iv. The Com plex Instruction n Set Com puting (CISC) - architecture reuses an instruction n set representing com plex esenting com plex opper		
inpu t and data outp ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC)- archi tectu re uses an instr uctio n set repr esen ting com plex		
t and data outp ut. iv. The Com plex Instruction Set Com puting (CISC) - architecture uses an instruction set representing com plex esenting com plex oper		
data outp ut. iv. The Com plex Instruction Set Com puting (CISC) - architecture uses an instruction set representing com plex oper		Inpu
outp ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC) - archi tectu re uses an instr uctio n set repr esen ting com plex		
ut. iv. The Com plex Instr uctio n Set Com puti ng (CISC)- archi tectu re uses an instr uctio n set		
iv. The Com plex Instr uctio n Set Com puti ng (CISC) - archi tectu re uses an instr uctio n set repr esen ting com plex oper		
The Com plex Instruction Set Computiing (CISC) - architecture uses an instruction set representing complex oper		
Com plex Instruction in Set Com puting (CISC) - architecture uses an instruction in set representing com plex oper		
plex Instr uctio n Set Com puti ng (CISC)- archi tectu re uses an instr uctio n set repr esen ting com plex oper		
Instruction Set Computing (CISC) - architecture uses an instruction set representing complex oper		
uction Set Computing (CISC) - architecture uses an instruction set representing complex oper		
n Set Com puti ng (CISC)- archi tectu re uses an instr uctio n set repr esen ting com plex oper		Instr
Com puti ng (CISC) - archi tectu re uses an instructio n set representing com plex oper		uctio
puti ng (CISC)- archi tectu re uses an instr uctio n set repr esen ting com plex oper		Com
ng (CISC) - archi tectu re uses an instructio n set representing com plex oper		
) - archi tectu re uses an instructio n set representing com plex oper		ng
) - archi tectu re uses an instructio n set representing com plex oper		(cisc
archi tectu re uses an instructio n set represen ting com plex oper) -
re uses an instruction n set representing com plex oper		archi
uses an instr uctio n set repr esen ting com plex oper		
an instruction n set representing complex oper		
instruction n set representing complex oper		
uction set representing complex oper		
n set repr esen ting com plex oper		
repr esen ting com plex oper		
esen ting com plex oper		
ting com plex oper		
com plex oper		
plex oper		
oper		
ation		oper
		ation

 T		
		s. It
		is
		possi
		ble
		for a
		CISC
		instr
		uctio
	1	n set
		to
		perf
		orm a
		large
		com
		plex
		oper
	a	ation
		with
		a
	!	singl
		е
		instr
		uctio
		n.
		The
		use
		of a
		singl
		e
		com
		plex
		instr
		uctio
		n in
		plac e of
		mult
		iple
	9	simp
		le
		instr
		uctio
		ns
	٤	great
		ly
		redu
		ces
		the
		prog ram
		mem
		ory

acce
SS
and
prog
ram
mem
ory
size
requi
rem
ent.
V.
The
Very
Long
Instr uctio
n
Wor
d
(VLI
W) -
archi
tectu
re
impl
eme
nts
mult
iple funct
ional
units
(ALU
`s,
mult
iplier
s,
etc.)
in
the data
path.
The
VLI
W
instr
uctio
n
pack
ages
one
stan

dard instr uctio n per funct ional unit of the data path. vi. Paral lel proc essin g archit tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ent ents (PE3) and and each proc essin g elem ent may asso ciate a data path.		
instructional unit of funct ional unit of fee data path. vi. Paral lel proc. essin g architectu re-impl eme nts mult liple conc urre nt Proc. essin g Elem ents (PES) and each proc. essin g Elem ents (PES) essin g Elem ents (PES) essin g Elem ents (PES		dard
uction new functional unit of the data path. vi. Parallel processin grace architecture implience architecture architecture implience architecture implience architecture		
n per funct ional unit of the data path. vi. Paral lel processin s architecture re-implements multiple conceurre unt processin s Elem ents (PES) and each processin s element ents (PES) and each e		
funct ional unit of the data path. Vi. Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and d each proc essin g elem ent may asso ciate a data path cont ainin g regis		
funct ional unit of the data path. Vi. Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and d each proc essin g elem ent may asso ciate a data path cont ainin g regis		per
ional unit of the data path. vi. Parall lel proc essin garchi tectu re-imple ents multiple conc urre nt Proc essin garchi essis essis es		funct
of the data path. vi. Paral lel lel proc essin g archi tectu re-impl eme nts multiple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ents (PEs) and each proc essin g elem ent may asso ciate a la data path cont linin g en registre.		ional
the data path. vi. Paral lel proc essin grachitecture re-implieme ants mult iple conc urre ant Proc essin grachitecture re-implieme eme ants mult iple conc urre ant proc essin grachitecture re-implieme ents (PES) and each proc essin grachitecture resident and each proc essin grachitecture resident grachitecture residen		
data path. vi. Paral lel proc essin si archi tectu re-impl eme nts multi iple conc urre archive essin si si archive eme nts multi iple conc urre archive essin si si essimate enti enti enti enti enti enti enti en		
path. vi. Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PES) and each proc essin g elem ent (PTS) and each proc essin g rejis		
vi. Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem eat at adata path		data
Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent (PES) and teach proc essin g elem ent may asso ciate a data path cont ainin g regis		path.
Paral lel proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent (PES) and teach proc essin g elem ent may asso ciate a data path cont ainin g regis		
ele proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PES) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
proc essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
essin g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent tmay asso ciate a data path cont ainin g regis		
g archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
archi tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PES) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
tectu re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent (PEs) and proc essin g elem ent cont ainin g rregis		archi
re- impl eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
impleme eme nts mult iple concurre nt Processin g Elem ents (PEs) and each processin g elem ent may asso ciate a data path cont ainin ainin g regis		
eme nts mult iple conc urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
nts mult iple concurre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g g regis		
iple conc urre nt Proc essin g Elem ents (PES) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
conc urre nt Proc essin g Elem ents (PES) and each proc essin g elem ent may asso ciate a data path cont		
urre nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont cont ainin g regis		
nt Proc essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
Processin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
essin g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
g Elem ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		essin
ents (PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		g
(PEs) and each proc essin g elem ent may asso ciate a data path cont ainin g regis		Eiem
and each proc essin g elem ent may asso ciate a data path cont ainin g regis		
each proc essin g elem ent may asso ciate a data path cont ainin g regis		
processin gelem ent may asso ciate a data path cont ainin g regis		
essin g elem ent may asso ciate a data path cont ainin g regis		
g elem ent may asso ciate a data path cont ainin g regis		
elem ent may asso ciate a data path cont ainin g regis		
ent may asso ciate a data path cont ainin g regis		elem
may asso ciate a data path cont ainin g regis		
asso ciate a data path cont ainin g regis		
ciate a data path cont ainin g regis		asso
data path cont ainin g regis		ciate
path cont ainin g regis		
cont ainin g regis		
ainin g regis		
g regis		cont
regis		
regis ter		g
l ter		regis
		ter

	and
	local
	mem
	ory.
	Singl
	e
	Instr
	uctio
	n
	Mult
	iple
	Data
	(SIM
	D)
	and
	Mult
	iple
	Instr uctio
	n
	Mult
	iple
	Data
	(MI
	MD)
	archi
	tectu
	res
	are
	exa
	mple
	s for
	paral
	lel
	proc
	essin
	g archi
	tectu
	re.
	In
	SIM
	D
	archi
	tectu
	re, a
	singl
	e
	instr
	uctio
	n is exec
	CVEC

	· · · · · ·	 1
		uted
		in
		paral
		lel
		with
		the
		help
		ofth
		e
		Proc
		essin
		g
		Elem
		ents.
		On
		the
		othe
		r
		hand
		, the
		proc
		essin
		g
		elem
		ents
		of
		the
		MIM
		D
		archi
		tectu
		re
		exec
		ute
		diffe
		rent
		instr
		uctio
		ns at
		a
		give
		n
		poin
		t of
		time.
		3.
		Selec
		ting
		the
		lang
		uage
		- A
		prog

	ram
	ming
	lang
	uage
	capt
	ures
	a
	Com
	puta
	tiona
	Mod el
	and
	map
	s it
	into
	archi
	tectu
	re.
	Any
	prog
	rami
	ng lang
	uage
	can
	be
	used
	.
	A
	mod el
	can
	be
	capt
	ured
	usin
	g
	mult
	iple
	prog
	ram ming
	lang
	uage
	s like
	C,
	C++,
	C#,
	Java,
	etc.
	for

 _		
		soft
		ware
		impl
		eme
		ntati
		ons
		and
		for
		hard
		ware
		impl eme
		ntati
		ons
		lang
		uage
		s like
		VHD
		L,
		Syste
		m C,
		Veril og,
		etc.
		C.C.
		On
		the
		othe
		r
		hand
		, a
		singl
		e lang
		lang uage
		can
		be
		used
		for
		capt
		uring
		a _.
		varie
		ty of
		mod els.
		Cert
		ain
		lang
		uage
		s are
		good
		in
		capt

		uring
		certa
		in
		com
		puta
		tiona
		mod
		el.
		For
		exa
		mple
		, C++
		is a
		good
		cand
		idate
		for
		capt
		uring
		an
		obje
		ct
		orie
		nted
		mod
		el.
		4.
		Parti
		tioni
		ng
		Syste
		m
		Requ
		irem
		ents
		into
		hard
		ware
		and
		soft
		ware
		-
		Fro
		m an
		impl
		eme
		ntati
		on
		pers
		pecti
		ve, it
		may

		be
		possi
		ble
		to
		impl
		eme
		nt
		the
		syste
		m
		requi rem
		ents
		in
		eithe
		r
		hard
		ware
		or
		soft
		ware
		(firm
		ware). It
		is a
		toug
		h
		decis
		ion
		maki
		ng
		task
		to
		figur e out
		whic
		h
		one
		to
		opt.
		vario
		us
		hard
		ware
		soft
		ware trad
		e-off
		s are
		used
		for
		maki
		ng a
		decis

		ion
		on
		the
		hard
		ware
		soft
		ware
		parti
		tioni
		ng.
		Evol
		Expl
		ain diffe
		rent
		com
		muni
		catio
		n
		buse
		S
		used
		in
		auto
		moti
		ve
		appli
		catio
		ns
		Auto
		moti
		ve
		appli catio
		ns
		mak
		e
		use
		of
		seria
		I
		buse
		s for
		com
		muni
		catio
		n,
		whic
		h
		great
		ly
		redu
		ces

 T	
	the
	amo
	unt
	of
	wirin
	g
	requi
	red
	insid
	e a
	vehi
	cle.
	Diffe
	rent
	type
	s
	ofser
	ial
	inter
	face
	buse
	S
	depl
	oyed
	in
	auto
	moti
	ve
	emb
	edde
	d d
	appli
	catio
	ns
	are
	1.
	Cont rolle
	rolle
	Area
	Net
	work
	(CAN
)-
	The
	CAN
	bus
	was
	origi
	nally
	prop
	osed

		1	
			by
			Robe
			rt
			Bosc
			h. It
			supp
			orts
			medi
			um
			spee d
			with
			data
			rates
			up
			to
			125
			Kbps
			and
			high spee
			d
			with
			data
			rates
			up
			to
			1Mb
			ps
			data
			trans fer.
			101.
			CAN
			is an
			even
			t-dri
			ven
			prot
			ocol inter
			inter face
			with
			supp
			ort
			for
			error
			hand
			ling
			in
			data
			trans missi
			on.

	It is
	gene
	rally
	empl
	oyed
	in
	safet
	У
	syste
	m
	like
	airba
	g cont
	rol;
	pow
	er
	train
	syste
	ms
	like
	engi
	ne
	cont rol
	and
	Antil
	ock
	Brak
	e
	Syste
	m
	(ABS
);
	and
	navi
	gatio n
	syste
	ms
	like
	GPS
	etc.
	2.
	Local
	Inter
	conn
	ect
	Net work
	(LIN)
	- LIN
	bus
	.,,,,,

		is a
		singl
		e
		mast
		er
		mult
		iple
		slave
		(up
		to 16
		inde
		pend
		ent
		slave node
		s)
		com
		muni
		catio
		n
		inter
		face.
		LIN
		is a
		low
		spee
		d,
		singl
		e
		wire
		com muni
		catio
		n
		inter
		face
		with
		supp
		ort
		for
		data
		rates
		up
		to 20
		Kbps
		and
		is used
		for
		sens
		or/a
		ctuat
		or
		inter

facin B. LIN LIN LIN LIN LIN LIN LIN LI		1
LIN bus is empl oyed in appli catio ns like mirr rir ror cont rols, fan cont rols, seat posit ionin g cont rols, wind ow. Cont rols, wind ow. Cont rols, wind ow. Cont rols, wind ow. Cont rols wind ow. Co		facin
bus is empl oyed in appli catio ns like mirr or cont rols, fan cont rols, seat posit ionin g cont rols, wind ow Cont rols, wind ow Cont rols, sold ion ion cont rols sold ion		g.
bus is empl oyed in appli catio ns like mirr or cont rols, fan cont rols, seat posit ionin g cont rols, wind ow Cont rols, wind ow Cont rols, sold ion ion cont rols sold ion		l IIN l
is employed in application in applic		
employed in applications in ap		
oyed in appli cations so like mirr or controls, fan controls, fan controls, seat positionin g controls, wind ow. Controls, wind ow. Controls, and position controls, wind ow. Controls, and position controls where e e response time is not a critic al issue .		
in application is like mirr or cont rols, fan cont rols, seat posit ioionin in g cont rols, wind ow. Cont rols, and posit ioin for rols, and posit ioin cont rols where e e response time is not a critic al issue se time is not a critic al issue se time in the role of the role		oyed
appli cations like mirr or cont or cont rols, fan cont rols, seat position in g cont rols, wind ow. Cont rols, and position cont rols, wind ow. Cont rols, and position cont rols, and a position cont rols where e response time is not a critic all is sue control rols where e response time all is not a critic all is not a criti		
cations like mirr or cont rols, fan cont rols, seat positionin g cont rols, wind ow. Cont rols, wind ow. Cont rols, and position in cont rols, and position in cont rols, and position in cont rols, and and position in cont rols where e response time is not a critic all is sue controls.		
like mirr or cont rols, fan cont rols, fan cont rols, seat posit ionin g g cont rols, wind ow. Cont rols, and posit ion cont rols, and posit ion cont rols where e resp onse time is not a critic all issue		catio
mirr or cont rols, fan cont rols, seat posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols and cont cont rols and cont cont cont cont cont cont cont cont		ns
or controls, fan controls, seat posit ionin g controls, wind ow. Controls, and posit ion controls, and posit ion controls where e resp onse time is not a critical issue		
cont rols, fan cont rols, seat posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols, and posit ion cont rols where e resp onse time is not a critic al issue .		
rols, fan cont rols, seat posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols where e e resp onse time is not a critic al issue		
fan controls, seat posit ionin g controls, wind ow. Controls, wind ow. Controls, and posit ion controls where e resp onse time is not a critical issue		
cont rols, seat posit ionin g g cont rols, wind ow. Cont rols, and posit ion cont rols where e response time is not a critic al issue		fan
rols, seat posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue		
seat posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue		
posit ionin g cont rols, wind ow. Cont rols, and posit ion cont rols wher e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		
ionin g cont rols, wind ow. Cont rols, and posit ion cont rols wher e e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		
g cont rols, wind ow. Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue .		ionin
cont rols, wind ow. Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue .		
wind ow. Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue		cont
ow. Cont rols, and posit ion cont rols wher e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		rols,
Cont rols, and posit ion cont rols where e resp onse time is not a critic al issue		
rols, and posit ion cont rols where e resp onse time is not a critic al issue		
and posit ion cont rols where e resp onse time is not a critic al issue		
position cont rols where e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		rols,
ion cont rols where e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		and
cont rols wher e resp onse time is not a critic al issue		
rols wher e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		
where e response time is not a critic al issue . 3. Media-Ori ente d Syste		rols
e resp onse time is not a critic al issue . 3. Medi a-Ori ente d Syste		
response time is not a critic al issue . 3. Media-Ori ente d Syste		
onse time is not a critic al issue		
time is not a critic al issue		onse
not a critic al issue		
critic al issue		
al issue . 3. Medi a-Ori ente d Syste		not a
3. Medi a-Ori ente d Syste		
3. Medi a-Ori ente d Syste		al
3. Medi a-Ori ente d Syste		
Medi a-Ori ente d Syste		'
Medi a-Ori ente d Syste		3.
a-Ori ente d Syste		
ente d Syste		
d Syste		
Syste		d
		Syste

	Tran
	spor
	t
	(MO
	ST)
	Bus -
	The
	Medi
	a-ori
	ente d
	syste
	m
	trans
	port
	(MO
	ST) is
	targe
	ted
	for
	auto
	moti
	ve audi
	o/vi
	deo
	equi
	pme
	nt
	inter
	facin
	g,
	used
	prim
	arily in
	Euro
	pean
	cars.
	A
	MOS
	T
	bus
	is a
	mult
	imed ia
	fibre
	-opti
	С
	poin
	t-to-
	poin
 I	

	Г	
		t
		netw
		ork
		impl
		eme
		nted
		in a
		star,
		ring'
		or
		daisy
		chai
		ned
		topo logy
		over
		optic
		al
		fibre
		cabl
		es.
		The
		MOS
		T
		bus-
		speci ficati
		ons
		defin
		e the
		physi
		cal
		layer
		as
		well
		as
		the
		appli catio
		n
		layer
		, ,
		netw
		ork
		layer
		, and
		medi
		a
		acce
		SS
		cont rol.
		101.

the state of the s	MOS T bus s an optic al ibre
t is operation of the contract	T bus s an ptic al ibre
	bus s an ptic al ibre
is of the contract of the cont	s an ptic al ibre
o fi	ptic al ibre
fi c	al ibre
fi c	ibre
c	
	cabl
	e
	onn
	ecte
	d
	etw
	een
	the
	lect
	ical
	Opti
	cal
	onv
	rter EOC
	and
	Opti
	cal
	lect
	ical
	onv
	rter
	OEC
),
	vhic
	h
	voul
	d
	rans
	ate
	nto
	the
	ptic
	al cabl
	е
	иOS
	7103 T
	us.
	10]
	[6]

(4) CO1 L2,L3 3 With FSM mod el, discuss she design n and oper ation of auto mati c tea/c offee vend ing mac hine. Solution: The FSM representation of for the above e requirem ent tailon for the above e requirem ent is sho wn		Γ	
L2,L3 3 With PSM mode el, discuss ss the design n and operation of auto matrix c tea/c offee vend ing mac hine. Solut ion: The PSM mode e e e equiverance in the above e e requirem nent is sho			[4]
L2,L3 3 With PSM mode el, discuss ss the design n and operation of auto matrix c tea/c offee vend ing mac hine. Solut ion: The PSM mode e e e equiverance in the above e e requirem nent is sho			
L2,L3 3 With PSM mode el, discuss ss the design n and operation of auto matrix c tea/c offee vend ing mac hine. Solut ion: The PSM mode e e e equiverance in the above e e requirem nent is sho			
With FSM mod el, discu ss the design n n and oper ation of auto mati c tatio fine. Solut ion: The FSM mad hine. Solut ion:			CO1
With FSM mod el, discu ss the design n n and oper ation of auto mati c tatio fine. Solut ion: The FSM mad hine. Solut ion:			. .
With FSM mod el, discu s s the design n and oper ation of auto mati c t tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation for the abov e requirem ent is sho			L2,L3
With FSM mod el, discu s s the design n and oper ation of auto mati c t tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation for the abov e requirem ent is sho			,
FSM mod el, discu ss the design n and oper ation of auto oper ation of auto mati c tea/c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation of nor the abov e requi rem ent is sho			3
FSM mod el, discu ss the design n and oper ation of auto oper ation of auto mati c tea/c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation of nor the abov e requi rem ent is sho			
FSM mod el, discu ss the design n and oper ation of auto oper ation of auto mati c tea/c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation of nor the abov e requi rem ent is sho			\\\/ith
mod el, discu ss the design n and oper ation of auto mati c t tea/c offee vend ing mac hine. Solut ion: The FSM representation n for the above e requirem rem ent is sho			
el, discu ss the desig n and oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation for the abov e requi			
discu ss the desig n and oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			
S S the design n and oper ation of auto matic c tea/c offee vend ing mac hine. Solut ion: The FSM representation for the above e requirement is shown			discu
desig n and oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requirem ent is sho			SS
n and oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation for the above e requirem ent is sho			the
and oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM representation for the above e requirement is sho			
oper ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			
ation of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tation n for the above e requi rem ent is sho			
of auto mati c tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			oper
auto mati c c tea/c offee vend ing mac hine. Solut ion: The FSM repr essen tatio n for the abov e requi rem ent is sho			of
mati c teach offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the above e requi rem ent is sho			
c tea/c offee vend ing mac hine. Solut ion: The FSM repr essen tatio n for the abov e requi rem ent is sho			
tea/c offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			
offee vend ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			tea/c
ing mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			offee
mac hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			
hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			ing
hine. Solut ion: The FSM repr esen tatio n for the abov e requi rem ent is sho			
Solut ion: The FSM repr esen tatio n for the abov e requirem ent is sho			
The FSM repr esen tatio n for the abov e requi rem ent is sho			nine.
The FSM repr esen tatio n for the abov e requi rem ent is sho			Solut
The FSM repr esen tatio n for the abov e requi rem ent is sho			
The FSM repr esen tatio n for the abov e requi rem ent is sho			
The FSM repr esen tatio n for the abov e requi rem ent is sho			[OBJ]
FSM repr esen tatio n for the abov e requi rem ent is sho			
representation for the above erequirement is sho			
esen tatio n for the abov e requirem ent is sho			
tatio n for the abov e requirement is sho			
n for the abov e requirement is sho			
the abov e requirement is sho			n for
abov e requi rem ent is sho			
e requirem ent is sho			
requirem ent is sho			
rem ent is sho			
ent is sho			
is sho			ent
			is
wn			
			wn

	in
	the
	belo
	w
	figur
	l igui
	e. It
	cont
	ains
	four
	state
	S
	nam
	ely
	i.
	Wait
	for
	coin
	ii.
	Wait
	for
	User
	Inpu
	t
	iii.
	Disp
	ense
	Tea
	iv.
	Disp
	ense
	Coff
	ee.
	The
	even
	t
	ʻlnse
	rt
	Coin'
	(5
	rupe
	e
	coin
	inser
	tion)
	trans
	ition
	s the
	state
	 ·

	to
	'Wai
	t for
	User
	Inpu
	t'.
	The
	syste
	m staye
	stays in
	this
	state
	until
	a
	user
	inpu
	t is
	recei
	ved
	from the
	butt
	ons
	'Can
	cel',
	'Tea'
	or
	'Coff
	ee'.
	lf lf
	the
	even
	t
	trigg
	ered
	in
	'Wai
	t
	State
	' is 'Can
	cel'
	butt
	on
	pres
	s,
	the
	coin
	is
	push
	ed
	out

	and
	the
	state
	"tran
	sitio
	ns to
	'Wai
	t for
	Coin'
	If
	the
	even
	t .
	recei
	ved in
	the
	'Wai
	t
	State
	' is
	eithe
	r"
	Tea'
	butt
	on
	pres
	s, or
	'Coff
	ee'
	butt on
	pres
	s,
	the
	state
	chan
	ges
	to
	ʻDisp
	ense
	Tea'
	and
	'Disp
	ense Coff
	ee'
	resp
	ectiv
	ely.
	'
	Once

the coffice/te a vent ing i over the respective e state s	fe ce nd is er, e
coffice/te e/te a vent ing i over the resp ective e state s	fe ce nd is er, e
e/te a vend ing i over the resp ective e state s	te nd is er, e
a vend ing i over the resp ective e state s	nd is er, e
vending ing i over the resp ective e state s	is er, e
ing i over the resp ective e state	is er, e
over the resp ective e state	er, e
the resp ective e state	e
resp ectiv e state	
ective e state	יט ו
e state	<u>.</u> l
state s	
tran	15
ition	
baci	
to the	
the	
\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	
t foi	
Coin	
state	- 1
few	
mod	ui
ficat	
ons	
like	
add	
ng a	
time	
out	
for	
the	
l t	
	ופ
	- 1
(Cur	
entl	'y
the	
vva t	
State	
'is	ິ
	, ni
te; i	
be	
re-c	
esig	
esign est to the control of the cont	,'' to

		а
		time
		out
		base
		d
		'Wai
		t
		State
		′. If
		no
		user
		inpu
		t is
		recei
		ved
		withi
		n the
		time
		out
		peri
		od, the
		coin
		is
		retur
		ned
		back
		and
		the
		state
		auto
		mati
		cally
		trans
		ition
		s to
		'Wai
		t for
		Coin'
		on the
		time
		out
		even
		t)
		and
		capt
		uring
		othe
		r
		even
		ts
		like,
		'Wat

	er
	not
	avail
	able'
	, (Tag./
	'Tea/
	Coff
	ee
	Mix
	not
	avail
	able'
	and
	chan
	ging
	the
	state
	to an
	'Erro
	r
	State
	' can
	be
	adde
	d to
	enha
	nce
	this
	desig
	n.
	[10]
	[10]
	[4]
	["]
	[6]
	ا ا
	CO1,
	1 (01,
	CO2
	,,,,
	L2,L3
	,
	4
	- "
	Defi
	ne
	Proc
	ess.
	Expl
	ain
	in
	detai

	 1
	I the
	struc
	ture,
	mem
	ory
	orga
	nizat
	ion
	and
	state
	trans
	ition
	of
	the
	proc
	ess.
	Solut
	ion:
	Proc
	ess
	A
	'Proc
	ess'
	is a
	prog
	ram,
	or
	part
	of it,
	in
	exec
	utio
	n.
	Proc
	ess
	also
	kno
	wn
	as an
	insta
	nce
	of a
	prog
	ram
	in
	exec
	utio
	n.
	'''
	Mult
<u> </u>	

	iple
	insta
	nces
	of
	the
	sam
	e
	prog
	ram
	can
	exec
	ute
	simu
	Itane
	ousl
	y.
	'
	A
	proc
	ess
	requi
	res
	vario
	us
	syste
	m
	reso
	urce
	s like
	CPU
	for
	exec
	uting
	the
	proc
	ess;
	mem
	ory
	for
	stori
	ng
	the
	code
	corr
	espo
	ndin
	g to
	the
	proc
	ess
	and
	asso
	ciate
	1 0.000

	d
	varia
	bles,
	1/0
	devi
	ces
	for
	infor
	mati
	on
	exch
	ange
	, etc.
	The
	Struc
	ture
	of a
	Proc
	ess
	The
	conc
	ept
	of
	'Proc
	ess'
	lead
	s to
	conc urre
	nt
	exec
	utio
	n
	(pse
	udo
	paral
	lelis
	m)
	of
	tasks
	and
	ther
	eby
	the
	effici
	ent
	utilis
	ation
	of
	 the

CPU and othe r system reso urce s. Conc. urre the exec utio n is achi eved thro ugh the sharing of CPU amo ng the sharing of CPU amo esse s. A proc. esse s. A proc. esse s. A proc. ess of regis ters, proc.		
and othe r syste m reso urce urce s. Concurre int exec utio n is achi eved through the sharing of CPU among the processe s. A processe s. A processe s. A processe s. A processe s. and hold sa a set of regis ters,		CPU
che r syste m reso urce s. Conc urre nt exec utio n is achi eved thro ugh the shari ng of CPU amo ng the proc esse s. A proc ess (imit ate) as a proc ess r in prop ertie s and hold s s set of regis ters,		
r syste mo reso urce s. Conc urre int exection is achi eved through the sharing of CPU amo ing the process of t		
syste m reso urce s. Concurre nt execution is achi eved through the sharing of CPU among the processe s. A processe s. A processe s. A processe s. and hold sa a processe s. and hold sa set of regis ters,		
m reso urce s. Conc urre nt exec ution is achi eved through the sharing of CPU among the processes. A processes. A A processes. S. (imit ate) as a processo rin propertie s and hold sa set of regis ters,		
reso urce s. Conc urre nt exec utio n is achi eved thro ugh the shari ng of CPU CPU amo ng the proc ess s. A proc ess si mimi cs (imit ate) as a proc ess or in prop ertie s and hold s a s a set of regis ters,		
urce s. Concourre nt exec utio n is achi eved thro ugh the shari ng of CPU amo ng the proc esse s. s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s s and hold s a s s set of regis ters,		
S. Concurrent thexecution is achieved through the sharing of CPU among the processes. A processes. A processes. (imit ate) as a processe essor in properties and hold of a processe essor in properties and hold sa set of regis ters,		
urre nt exec utio n is achi eved thro ugh the shari ng of CPU amo ng the proc esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
urre nt exec utio n is achi eved thro ugh the shari ng of CPU amo ng the proc esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
nt exection is achinal		Conc
executions is achieved through the sharing of CPU among the processes. A processes. A processes. A processes. C (imit ate) as a processes as and hold sa as a set of regis ters,		
utio n is achi eved thro ugh the shari ng of CPU amo ng the proc esse s. A proc ess s (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
n is achi eved thro ugh the sharing of CPU amo ng the processe s. A processe smimi cs (imit ate) as a processo r in propertie s and hold s a set of regis ters,		
achi eved thro ugh the sharing of CPU among the processe s. A processe s. A processe mimics (imit ate) as a processo r in properties s and hold s a set of regis ters,		
eved thro ugh the sharing of CPU among the processes. A process miming cs (imit ate) as a process rin properties s and hold s a set of regis ters,		
through the sharing of CPU among the processes. A processes s. A processes miming cs (imit ate) as a processor in properties s and hold s a set of registers,		
ugh the sharing of CPU amoong the processes. A process s. A process mimin cs (imit ate) as a process or in properties s and hold s and hold s a set of regis ters,		
the sharing of CPU amo ng the processe s. A processe mimices (imit ate) as a processor in properties s and hold s a set of regis		
sharing of CPU amo ng the processe s. A process mimi cs (imit ate) as a processor in properties s and hold s a set of regis ters,		l ugii
ng of CPU amo ng the processe s. A process mimi cs (imit ate) as a processor in properties s and hold s a set of regis ters,		
CPU amo ng the proc esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		ng of
amo ng the proc esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		CPU
ng the proc esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
the processe s. A proceess mimi cs (imit ate) as a proceess rin propertie s and hold s a set of regis ters,		
processe s. A process mimi cs (imit ate) as a processo r in propertie s and hold s a set of regis ters,		the
esse s. A proc ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
A process miming cs (imit ate) as a processor in properties s and hold s a set of regis ters,		
process mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		S.
process mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
ess mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
mimi cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
cs (imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
(imit ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
ate) as a proc esso r in prop ertie s and hold s a set of regis ters,		
as a proc esso r in prop ertie s and hold s a set of regis ters,		
proc esso r in prop ertie s and hold s a set of regis ters,		
esso r in prop ertie s and hold s a set of regis ters,		
r in prop ertie s and hold s a set of regis ters,		
propertie s and hold s a set of regis ters,		
ertie s and hold s a set of regis ters,		
s and hold s a set of regis ters,		
hold s a set of regis ters,		S
s a set of regis ters,		
set of regis ters,		
of regis ters,		
regis ters,		
ters,		
proc		ters,
Į proc		nroc
		proc

	· · · · · · · · · · · · · · · · · · ·	
		ess
		statu
		s, a
		Prog
		ram
		Coun
		ter
		(PC)
		to
		poin
		t to
		the
		next
		exec
		utabl
		e
		instr
		uctio
		n of
		the
		proc
		ess,
		a
		stack
		for
		holdi
		ng
		the
		local
		varia
		bles
		asso
		ciate d
		with
		the
		proc
		ess
		and
		the
		code
		corr
		espo
		ndin
		g to
		the
		proc
		ess.
		This
		can
		be
•		

	Т	
		visua
		lised
		in
		belo
		w
		figur
		е
		A
		proc
		ess
		whic
		h
		inhe
		rits
		all-th
		е
		prop
		ertie
		s of
		the
		CPU
		can
		be
		consi
		dere
		d as
		a
		virtu
		al
		proc
		esso
		r,
		awai
		ting
		its
		turn
		to
		have
		its
		prop
		ertie
		S
		switc
		hed
		into
		the
		physi
		cal
		proc
		esso
		r.
I		

when the process get its turn its register and the process counter register main per terman per ter
n th pro ess get its turn its regi ter and the pro ran cou ter regi ter man per ter ter man per ter ter man per ter to the the phy cal regi ter of the terms of the ter
pro ess get its turn its regi ter and the pro ran cou ter man per
ess get its turn its regi ter and the pro ran cou ter regi ter ma pec
get its turn its regiter and the pro ran cou ter regiter ma pector to the phy cal regiter ma cou ter regiter ma pector to the phy cal regiter term to the phy cal regiter term the phy cal regiter t
its turr its register. and the prospect of the man per term and per term and the prospect of the man per term and per ter
turn its register. and the program counter register man ped to the phy cal register. to the phy cal register.
its regiters and the program could ter regiter man ped ter man ped
regiter: and the pro ran cou ter regi ter ma pec to the phy cal regi ter of
tern and the program counter register man ped to the phy cal register of the phy cal register.
and the pro ran cou ter regi ter ma pec
the pro ran cou ter regi ter ma pec to the phy cal regi ter of the phy cal region to the phy cal region ter of the phy cal region term of the phy cal reg
pro ran cou ter regi ter ma pec
ter registers of the
ter of the
ter reginter reginter ter reginter ter reginter ter reginter regin
termal per to the phy cal registers of the
to the phy cal register.
to the phy cal register.
to the phy cal register.
the phy cal registers of the phy cal registers.
the phy cal registers of the phy cal registers.
phy cal reging terms of the
cal region of the
regi ters of the
ten of the
the
the latest the latest term of th
CPL
A pro
ess
whi
h h
inh
rite
all-t
e
pro
erti
s o the
CPU
car
be
con
der
d a
a
virt
al al

 T	
	proc
	esso
	r,
	awai
	ting
	its
	turn
	to
	have
	its
	prop ertie
	S
	switc
	hed
	into
	the
	physi
	cal
	proc
	esso
	r.
	Whe
	n the
	proc ess
	gets
	its
	turn,
	its
	regis
	ters
	and
	the
	prog
	ram
	coun
	ter
	regis
	ter
	map ped
	peu
	to
	the
	physi
	cal
	regis
	ters
	of
	the

CPU. rom a mem ory pers pecti ve, the mem ory occu pied by the proc ess is segr egat ed into thre e regio ns, nam ely, Stac k mem ory, Data mem ory, Data mem ory ory and Code mem ory s s sho wn in abov e figur e. The 'Stac k'			
a a mem ory pers pective, the the mem ory occu pied by the process is segregated into three e regions, namely, State k mem ory, Data mem ory and code mem ory and code mem ory and code mem ory and the first personal code mem or a first pe			CPU.
mem ory pers pective. ve, the mem ory occupied by the process is segar egated into three erggions, anamely, Stac & mem ory, Data mem ory, Data mem ory and Code mem ory and Code mem ory and code mem mem ory s s shown in above e figure. The control of the contr			rom
ory pers pecti ve, the mem ory occu pied by the proc ess is segr egat ed into thre e regio ns, nam elv, elv, Stac k mem ory, Data mem ory, Data mem ory s s sho wn in abov e figur e. The			a
pers pective, the mem ory occu pied by the proc ess is segr egat ed into thre e regio ns, nam ely, Stac k mem ory, Data mem ory, Data mem ory s s sho wn in abov e figur e. The			mem
pective, ve, the mem ory occu piede by the process is segregated into three e regions, namely, Stack k mem ory, Data mem ory, Data mem ory and Code mem ergs show which is show which in above e figure.			
ve, the mem ory occu pied by the process is segar egar ed into three e regions, nam ely, Stack mem ory, Data mem ory, Data mem ory and Code mem ory a s s sho wn in abov e figur e.			pers
the mem ory occu pied by the proc ess is segr egat ed into thre e regio ns, nam elv, Stac k mem ory, Data mem ory, Data mem ory and Code de mem orya s s sho wn i abov e figur e. The			
mem ory occu pied by the proc ess is segrification of the end of t			ve,
ory occupied by the process is segregat eat into three e region ns, nam ely, Stac k mem ory, Data mem ory and Code mem ory and and and code mem ory and code			
occu pied by the proc ess is segr egat ed into thre e regio ns, nam ely, Stac k mem ory, Data mem ory, Data mem ory s s sho wh in abov e figur e.			
pied by the proc ess is segr egat ed into thre e regio ns, nam elv, Stac k mem ory, Data mem ory and Code mem ory s s sho wn in abov e figur e.			
by the proc ess is segregat ed into three e regions, namely, Stack mem ory, Data mem ory and Code mem ory s show in above e figure.			
process is segregated into three eregions, namely, Stack memmory, Datamem ory, Datamem ory, and Codemem orys shown in above efigure.			by
ess is segregat ed into three e regions, namely, Stack k mem ory, Data mem ory, and Code mem orya s shoo wn in above e figure.			
is segrated into three e regions, namely, State k memory, Data memory and Code memory s shown in above e figure.			
segr egat ed into thre e regio ns, nam elv, Stac k mem orv, Data mem ory and Code mem orya s sho wn in abov e figur e.			
egat ed into thre e regio ns, nam elv, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e. The			
ed into thre e regio ns, nam ely, Stac k mem ory, Data mem ory and Code mem orys s sho wn in abov e figur e. The Stac			egat
thre e regio ns, nam ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e. The			ed
e regio ns, nam ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e.			
e regio ns, nam ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e.			thre
ns, nam ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e.			е
nam ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e. The			regio
ely, Stac k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e. The			
Stac k mem ory, Data mem ory and Code mem orya s s sho wn in abov e figur e. The			
k mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e.			ely,
mem ory, Data mem ory and Code mem orya s sho wn in abov e figur e.			
ory, Data mem ory and Code mem orya s sho wn in abov e figur e. The 'Stac			
Data mem ory and Code mem orya s sho wn in abov e figur e. The 'Stac			
ory and Code mem orya s sho wn in abov e figur e. The 'Stac			Data
and Code mem orya s sho wn in abov e figur e. The 'Stac			
Code mem orya s s sho wn in abov e figur e. The 'Stac			
mem orya s sho wn in abov e figur e. The 'Stac			
orya s sho wn in abov e figur e. The 'Stac			
s sho wn in abov e figur e. The 'Stac			
sho wn in abov e figur e. The 'Stac			
wn in abov e figur e. The 'Stac			
abov e figur e. The 'Stac			wn
e figur e. The 'Stac			in
e figur e. The 'Stac			abov
e. The 'Stac			
The 'Stac			
'Stac			e.
'Stac k'			
			'Stac
		<u> </u>	k′

		mem
		ory
		hold
		s all
		tem
		pora
		ry
		data
		such
		as varia
		bles
		local
		to
		the
		proc
		ess.
		<u> </u>
		Data
		mem ory
		hold
		s all
		glob
		al
		data
		for
		the
		proc
		ess.
		Proc
		ess
		State
		s
		and
		State
		Tran
		sitio
		n
		The
		creat
		ion
		of a
		proc
		ess
		to its
		term
		inati on is
		not a
		singl
		e
	<u> </u>	

		step
		oper
		ation
		. The
		proc
		ess
		trave
		rses
		thro
		ugh
		а
		serie
		s of
		state
		S
		duri
		ng
		its
		trans
		ition from
		the
		newl
		у
		creat
		ed
		state
		to
		the
		term
		inate
		d
		state
		The
		cycle
		thro
		ugh
		whic
		h a
		proc
		ess
		chan
		ges
		its
		state
		from
		"ne
		wly creat
		ed'

		to
	, , , , , , , , , , , , , , , , , , ,	exec
		utio
		n
		com
	р	lete
		d'is
		kno
		wn
		as Proc
		ess
	l ì	Life
		Cycl
		e'.
		The
		ario
		us
		tate s
		hro
		ugh
	l w	vhic
		ha
		oroc
		ess
		rave
		ses duri
		ng a
	P	roc
		ess
	L	Life
		Cycl
		е
		ndic
		ites
		the
		curr
		ent tatu
		tatu s of
		the
		roc
	6	ess
	l w	vith
		esp
		ect
		to
		ime
	٥	and

 -		
		also
		provi
		des
		infor
		mati
		on on
		what
		it is
		allo
		wed
		to
		do
		next.
		- Belo
		w
		figur
		e
		repr
		esen
		ts
		the vario
		us
		state
		s
		asso
		ciate
		d
		with a
		proc
		ess.
		he
		state
		at
		whic h a
		proc
		ess
		is
		bein
		g .
		creat
		ed is refer
		red
		as
		'Crea
		ted
<u> </u>	<u> </u>	

	State '.
	The
	Oper
	ating Syste
	m
	reco
	gnise
	s a
	proc
	ess in
	the
	'Crea
	ted
	State
	' but no
	reso
	urce
	s are
	alloc
	ated
	to
	the
	proc
	ess.
	The
	state
	wher
	e a
	proc
	ess
	is ince
	pted
	into
	the
	mem
	ory
	and awai
	ting
	the
	proc
	esso
	r
	time
I	

		for
		exec
		utio
		n, is
		kno
		wn
		as
		'Rea
		dy
		State
		'. At
		this
		stag
		e,
		the
		proc ess
		is
		plac
		ed in
		the
		'Rea
		dy
		list'
		queu
		e .
		main
		taine
		d by the
		OS.
		03.
		The
		state
		wher
		e the
		sour
		ce
		code
		instr
		uctio
		ns
		corr
		espo ndin
		g to
		the
		proc
		ess
		is
		bein
		g

	exec
	uted
	is
	calle
	d
	'Run
	ning State
	'. At
	whic
	h the
	proc
	ess
	exec
	utio
	n
	happ
	ens.
	l l
	Bloc
	ked
	State
	/Wai
	State
	refer
	s to
	a state
	wher
	e a
	runn
	ing
	proc
	ess is
	tem
	pora rily
	'"'
	susp
	ende
	d
	from
	exec
	utio
	n
	and
	does
	not
	have
	imm
	ediat
	e
<u> </u>	

	<u></u>		
			acce
			ss to
			reso
			urce
			s.
			The
			bloc
			ked
			state
			migh
			t be
			invo
			ked
			by
			vario
			us
			cond
			ition
			s like:
			the
			proc
			ess
			ente
			rs a
			wait
			state
			for
			an
			even
			t to
			occu
			r
			(e.g. Wait
			ing
			for
			user
			inpu
			ts
			such
			as
			keyb
			oard
			inpu
			t)
			_
			or
			waiti
			ng for
			getti
i		l	5000

ng acce ss to a a shar ed reso urce A state where the process complete sits exection in its known with a same state to a state to a process from one state to another is known as sition of a process from another is known as sition one state to another is known with a same state to a same		 	
acce ss to a sharshar ed reso urce A state wher e the proc ess com plete s its exec utio utio n is kno wn as "." The trans ition of a state "," The trans ition of a state to anot her her is kno un as state to anot the her is kno un as state to anot the her is kno un as state to anot the her is kno un as state to anot the her is kno un as 'Stat e trans ition'		r	ng
ss to a shar ed reso urce A state where the process complete sits execution is known as Complete d state. '.' The transition of a process from one state to another the transition of a process from the transition of the transition of transition			
a shar ed reso urce A state where e the process is sits execution in is known as 'Complete d' State d' ',' The transition or a process from one state to another is known as 'State to a			
shar ed reso urce A state where the process complete sits its execution is known as "Complete de destate". The transition of a process from one state to another the transition of a process from the transition one state to another the transition and the transition one state to another transition and the transition and the transition and the transition and transition and transition and transition are transition.			
ed reso urce A state where the process complete sits execution in is known as Complete d State d State d State to another sits the transition of a process from one state to another is known as a state to a state to another is known as a state to a stat			
reso urce A state where e the proc. ess complete sits exec. ution is known as state to to anot here is known on the state to to anot here is known on the state to to anot here is known on the state to to anot here is known on the state to to anot here is known on the state to to anot here is known on the state to to anot here is known on the state to the state the state to the state the stat			
A state where e the process complete sits execution is known as as 'Complete distant the transmitten one state to another is known as 'Complete distant the transmitten one state to another is known as 'State e transmitten'			
state where e the process complete sits execution in its sits exec		ur	rce
state where e the process complete sits execution in its sits exec			
where e the process complete sits execution is known as 'Complete d' State		/	A
e the process complete sits execution in is known as 'Complete d' State '.' The trans ition of a process from one state to anot her is known as 'State to anot her is 'State to anot			
process com plete sits execution in is known as 'Com plete d' d' State s'. The trans ition of a process from one state to anot her is known as 'State to anot her is known as 'State to to anot ther is known as 'State to to anot there is known as 'State to to anot there is known as 'State to to anot the trans ition'			
ess com plete s its execution in is known as 'Com plete d state transition of a process from one state to anot here is known as 'State to anot here is 'Stat			
com plete s its sits exec utio n is kno wn as "Com plete d State ' The trans ition of a proc ess from one state to anot her is kno wn as "Stat e e trans ition'			
plete s its exec utio n is kno own as 'Com plete d State '. The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
s its exect uttion is known as 'Complete d' State '.' The transition of a process from one state to anot her is known as 'State e transition'			
execution is known as 'Complete d' State d' State '.' The transition of a process from one state to anot her is known as 'State to anot her is known as 'State e transition'		ple	ete
utio n is kno wn as 'Com plete d State '. The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
n is kno wn as 'Com plete d d State '.' The trans ition of a process from one state to anot her is kno wn as 'State e trans ition'			
kno wn as 'Com plete d State '.' The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
wn as 'Com plete d State '. The trans ition of a a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
as 'Com plete d State '. The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
'Com plete d State '.' The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e e trans ition'			
plete d State '.' The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
d State '.' The trans ition of a proc ess from one state to anot her is kno wn as 'Stat' e trans ition'			
The trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
The trans ition of a proc ess from one state to anot her is kno wn as 'State e trans ition'		Sta	ate
trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			'.
trans ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
ition of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
of a proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
proc ess from one state to anot her is kno wn as 'Stat e trans ition'			
ess from one state to anot her is kno wn as 'State e trans ition'			
from one state to anot her is kno wn as 'State e trans ition'			
one state to anot her is kno wn as 'State e trans ition'			
state to anot her is kno wn as 'Stat e trans ition'			
to anot her is kno wn as 'Stat e trans ition'			
anot her is kno wn as 'Stat e trans ition'			
her is kno wn as 'Stat e trans ition'			
is kno wn as 'Stat e trans ition'			
wn as 'Stat e trans ition'			
as 'Stat e trans ition'		kı	no
'Stat e trans ition'			
e trans ition'			
trans ition'			
ition'			
			.
Whe		_	,,
whe whe na			
		''	<u> </u>

	proc
	ess
	chan
	ges
	its
	state
	from
	Read
	y to
	runn
	ing
	or
	from
	runn
	ing
	to
	bloc
	ked
	or
	term
	inate
	d or
	from
	bloc
	ked
	to
	runn
	ing,
	the
	CPU
	alloc
	ation
	for
	the
	proc
	ess
	may
	also
	chan
	ge.
	[10]
	[10]
	[4]
	[4]
	[6]
	[0]

 T	
	CO2
	L4
	5
	With
	a neat
	diagr
	am,
	expl ain
	oper
	ating
	syste
	m archi
	tectu
	re.
	Solut
	ion:
	The
	Oper
	ating
	Syste m
	(OS)
	acts
	as a bridg
	e
	betw
	een the
	user
	appli
	catio ns/
	tasks
	and
	the
	unde rlyin
	g
	syste
	m reso
	urce
	S
	thro ugh
	ا مق

sy sy	set of
sy sy	
	OI I
	yste
	m
	unct
	onal
	ties
a	and
se	ervi
c	es.
	»
	Γhe
	rim
	ary
	unct
	ons f an
	per
	ting
	yste
	ms
	re –
	»
	Лak
e	the
	yste
	m
	onv
	nie
	t to
	use
	»
)rga
	nize
	and
	nan
	age
	the
	yste
	m
	eso
	ırce
	S
	ffici
	ntly
	and
	ctly
	ctly
	•

 T	<u> </u>	
		The
		follo wing
		Figur e
		gives
		an
		insig ht
		into
		the
		basic
		com pone nts
		of
		an oper ating syste m and their inter faces with rest of the worl d.
		The Kern el:
		» The kern el is the core

		 1
		of
		the
		oper
		ating
		syste
		m. It
		is
		resp
		onsi
		ble
		for
		man
		agin
		g
		the
		syste
		m
		reso urce
		s
		and
		the
		com
		muni
		catio
		n
		amo
		ng the
		the
		hard
		ware
		and
		othe
		r syste
		m
		servi
		ces.
		»
		Kern
		el
		acts
		as
		the
		abstr
		actio
		lavor
		layer betw
		een
		syste
		m
		reso
<u> </u>		

urce s and user applications. ** ** ** ** ** ** ** ** ** ** ** ** *		
s and user appil catio ons. Wern el cont ains a set of syste mibra ries and servi ces. For a gene rall purpose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess m		urce
applications. Where elecont ains a set of system ilibraties and services. Where elecont ains a set of system ilibraties and services. Where elecont ains different ains different elecont ains ains different elecont ains ains ains ains ains ains ains ains		
user applications. ** ** ** ** ** ** ** ** ** ** ** ** *		
applications. """ """ """ """ """ """ """		
cations. "" "" "" "" "" "" "" "" ""		
cations. "" "" "" "" "" "" "" "" ""		appli
ns. " Kern el el cont ains a set of syste m libra ries and servi ces. " For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		catio
Kern el cont ains a set of syste m libra ries and servi ces. » For a gene ral purp Ose OS, the kern el el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
Kern el cont ains a set of syste m libra ries siand servi ces. Servi ces. For a gene ral purp ose O.S., the kern el el cont aains diffe rent servi ces like mem ory manage men t, proc ess man age men t, proc ess man age men t, proc ess man age men t, t, the total process the total		
el cont ains a set of syste milbra ries and servi ces. » For a gene ral purp ose oS, the kern el cont ains diffe rent servi ces like mem ory man age men t, t, proc ess man age men t, t,		»
cont ains a set of syste milibra ries and servi ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		Kern
ains a set of syste m ilibra ries and servi ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces ilike mem ory man age men t, proc ess man age men		el
a set of syste of syste memory ces. "For a gene ral purp ose OS, the kern el cont ains different servi ces like memory man age men t, proc ess man age men t, t, t, proc ess man age men t, t, t, proc ess man age men t,		
of syste m libra ries and servi ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
syste m libra ries and servi ces. » For a gene ral ral purp ose OS, the kerm el cont ains diffe rent servi Ces like mem ory man age men t, proc ess man age men t, t,		
m libra ries and servi ces. Sor a gene ral purpo ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, proc ess man age men t, t, proc ess man t, t, t, the total content to the tota		
libra ries and servi ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, proc ess man age men t,		
ries and servi ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
and servi ces. » For a gene ral purp ose OS, the kerm el cont ains diffe rent servi ces like mem ory man age men t, proces ess man age men t, t, the transfer ess ess ess ess ess ess ess ess ess e		
servi ces. "For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, process man age men t, t, en contact c		
ces. » For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, proc		
» For a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, t,		(63.
a gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, t,		» For
gene ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
ral purp ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
purp ose OS, the kern el cont ains different servi ces like mem ory man age men t, proc ess man age men t, proc ess man age men t,		
ose OS, the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t, proc ess man age men t, proc ess man age men t,		
the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
the kern el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		OS,
el cont ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
cont ains different servi ces like mem ory man age men t,		
ains diffe rent servi ces like mem ory man age men t, proc ess man age men t,		
different servi ces like mem ory man age men t, proc ess man age men t,		
rent servi ces like mem ory man age men t, proc ess man age men t,		
servi ces like mem ory man age men t, proc ess man age men t,		
ces like mem ory man age men t, proc ess man age men t,		
like mem ory man age men t, proc ess man age men t,		
mem ory man age men t, proc ess man age men t, t,		
ory man age men t, proc ess man age men t, t,		
man age men t, proc ess man age men age men t, t, the man age men t, the man the men t, the man the man t, the man th		
age men t, proc ess man age men age men t,		
men t, proc ess man age men t,		
t, proc ess man age men t,		
proc ess man age men t,		
man age men t,		proc
age men t,		
men t,		
t,		
time		
•		time

		man
		age
		men
		t, file
		syste
		m
		man
		age
		men
		t, I/O
		syste
		m
		man
		age
		men
		t.
		Proc
		ess
		Man
		age
		men
		t:
		deal
		S
		with
		man
		agin
		g the
		proc
		ess/
		tasks
		.
		»
		Proc
		ess
		man
		age
		men
		t
		inclu
		des
		-
		»
		setti
		ng
		up a
		mem
		ory
		for

	the
	proc
	ess
	»
	loadi
	ng
	proc
	ess
	code
	into
	mem
	ory
	»
	alloc
	ating
	syste
	m
	reso
	urce
	S
	»
	sche
	dulin
	g
	and
	man
	agin
	g the
	exec
	utio
	n of
	the
	proc
	ess
	»
	"
	setti
	ng
	up .
	and
	man
	agin
	g
	Proc
	ess
	Cont
	rol

Bloc
k
(PCB
)
′
»
"
intor
inter
proc
ess
com
muni
catio
n .
and
sync
hron
izati
on
»
proc
ess
term
inati
on/
delet
ion,
etc.
D .:
Prim
ary
Me
mor
У
Man
age
men
t:
refer
s to
a
volat ile
mem
ory (DA
(RA
M),
wher
е
nro-
proc
esse

	s are
, ,	load
	ed
	and
	varia
	bles
	and
	shar
	ed
	data
	are
	store d.
	u.
	»
	"
	The
	Me
	mor
	У
	Man
	age
	men
	t
	Unit
	(MM
	U) of the
	kern
	el is
	resp
	onsi
	ble
	for –
	»
	Keep
	ing a
	track
	of whic
	whic h
	part
	of
	the
	mem
	ory
	area
	is
	curr
	ently
	used
	by

	· · · · · · · · · · · · · · · · · · ·	
		whic
		h
		proc
		ess
		»
		Alloc
		ating
		and
		De-a
		lloca
		ting
		mem
		ory
		spac
		e on
		a
		need basis
		•
		File
		Syste
		m
		Man
		age
		men
		t:
		File
		is a
		colle
		ction
		of
		relat
		ed
		infor
		mati
		on. A file
		coul
		d be
		a
		prog
		ram
		(sour
		ce
		code
		or
		exec
		utabl
		e),
		text
		files,

		imag
		e
		files,
		word
		docu
		men
		ts,
		audi
		0/
		vide o
		files,
		etc.
		» A
		file
		syste
		m
		man
		age
		men
		t servi
		ce of
		kern
		el is
		resp
		onsi
		ble
		for –
		»
		Th
		The
		creat ion,
		delet
		ion
		and
		alter
		ation
		of
		files
		»
		Crea
		tion,
		delet
		ion,
		and
		alter
		ation
		of
	<u> </u>	

		т т	
			direc
			torie
			S
			»
			Savi
			ng of
			files
			in
			the
			seco
			ndar
			У
			stora
			ge
			mem
			ory
			»
			Provi
			ding
			auto
			mati
			С
			alloc
			ation
			of
			file
			spac
			e .
			base
			d on
			the
			amo
			unt
			of
			free
			runn
			ing
			spac
			e
			avail
			able
			»
			"
			Drovii
			Provi
			ding
			flexi
			ble
			nami
			ng
	1	<u> </u>	

 <u></u>	
	conv
	ersio
	n for
	the
	files.
	1/0
	Syste
	m
	(Devi
	ce)
	Man
	age
	men
	t:
	Kern
	el is
	resp
	onsi
	ble for
	routi
	ng
	the
	1/0
	requ
	ests
	comi
	ng
	from
	diffe
	rent
	user
	appli
	catio
	ns to the
	appr
	opria
	te
	1/0
	devi
	ces
	of
	the
	syste
	m.
	, lm
	» In
	a well
	struc
	ture
	d

		os,
		direc
		t
		acce
		ss to
		1/0
		devi
		ces
		is
		not
		allo
		wed;
		acce
		ss to
		the
		m is
		esta
		blish
		thro
		ugh
		Appli
		catio
		n
		Prog
		ram
		ming
		Inter
		face
		(API)
		»
		The
		kern
		el
		main
		tains
		list
		of all
		the
		1/0
		devi
		ces
		of
		the
		syste
		m.
		»
		"
		"Dev
		ice
	<u> </u>	-

			vlan
		''	/Iaii
		l a	ager " of
		"	`of
		t	the
		l k	kern
			el is
			esp
			onsi
			ble
		1	for
			and
			ling
		'	1116 All
			all
			I/O
		re	elat
			ed
			per
			tion
			S.
			»
		Т	The
			Devi
			ce
			M an
		a	ager
			is
		r	esp
			onsi
			ble
		10	or –
			»
		L	.oad
			ing
			and
			ınlo
			adin
		g	g of
		l d	devi
			ce
			lrive
			rs
			»
		l _F	xch
			angi
		"	nigi
			ng
			nfor
		n	nati
			on
 <u> </u>			

<u></u>	г г		_
		and	ıl
		the	
		syste	
		m	·
		spec	, l
		fic	"
		cont	
		rol	
		signa	<u> </u>
		ls to	
		and	
		from	
		the	
		devi	
		ce.	
		Prot	
		ectio	د
		n	
		Syste	
		ms:	
		Mod	
		ern	
		opei	r
		ating	g
		syste	
		ms	
		are	
		desi	g
		ned	
		in	
		such	۱
		way	,
		to	
		supp	э
		ort	
		mult	
		iple	
		user	
		S	
		with	
		diffe	دِ
		rent	t
		leve	!
		s of	
		acce	دِ
		SS	
		per	
		miss	
		ons.	
		»	-
		The	,
		prot	

		1
		ectio
		n
		deal
		S
		with
		impl
		eme
		nting
		the
		secu
		rity polic
		ies
		to
		restr
		ict
		the
		acce
		ss of
		syste
		m
		reso
		urce
		s and
		parti
		cular
		user
		by
		diffe
		rent
		appli
		catio
		n or
		proc
		esse
		s and
		diffe
		rent
		user.
		[10]
		[ran
		[4]
		[6]
		[6]
		CO2
		L1,L3
Ц		

 T	 	
		6
		Illust rate
		the
		conc
		ept
		of
		"dea
		dloc
		k"
		with a
		neat
		diagr
		am.
		Men
		tion
		the diffe
		rent
		cond
		ition
		S
		favor
		a dead
		lock
		situa
		tion.
		Solut
		ion:
		Dea
		dloc
		k is
		the
		cond
		ition in
		whic
		h a
		proc
		ess
		is
		waiti
		ng for a
		reso
		urce
		held
		by

	anot
	her
	proc
	ess
	whic
	h is
	waiti
	ng
	for a
	reso
	urce held
	by
	the
	first
	proc
	ess;
	henc
	e,
	none
	of
	the
	proc
	esse
	s are able
	to
	mak
	e
	any
	prog
	ress
	in
	their
	exec
	utio
	n.
	Proc
	ess A
	hold
	s a
	reso
	urce
	"x"
	and
	it
	want
	s a
	reso
	urce
	"y" held
	by
	l na

		Proc
		ess
		В.
		Proc
		ess B
		is
		curr
		ently
		holdi
		ng reso
		urce
		"y"
		"y and
		it
		want
		s the
		reso
		urce
		"x"
		whic
		h is
		curr
		ently
		held
		by
		Proc
		ess
		A.
		Both
		hold
		the
		resp
		ectiv
		е
		reso
		urce
		S
		and
		they
		com
		pete
		each othe
		r to
		get
		the
		reso
		urce
		held
		by
		the
		resp
		ectiv

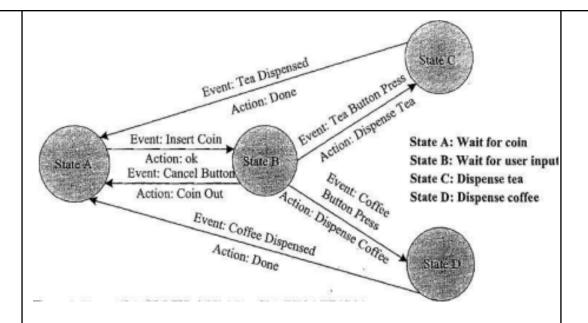
	e
	proc
	esse
	S.
	Cond
	ition s
	Favo
	ring
	Dea
	dloc
	k:
	Mut ual
	Exclu
	sion:
	The
	crite
	ria
	that
	only one
	proc
	ess
	can
	hold
	a
	reso
	urce
	at a time.
	Mea
	ning
	proc
	esse
	S
	shou ld
	acce
	SS
	shar
	ed
	reso
	urce
	s with
	mut
	ual
	exclu
	sion.
•	

	<u> </u>	
		Hold
		&
		Wait
		: The
		cond
		ition
		in
		whic
		ha
		proc
		ess hold
		s a
		shar
		ed
		reso
		urce
		by
		acqu
		iring
		the
		lock
		cont
		rollin
		g the shar
		ed
		acce
		SS
		and
		waiti
		ng
		for
		addit
		ional
		reso
		urce s
		held
		by
		othe
		r
		proc
		esse
		S.
		No
		Reso
		urce
		Pree mpti
		on:
		The
		crite

	T	
		ria
		that
		Oper
		ating
		Syste
		m
		cann
		ot
		take
		back
		a
		reso
		urce
		from
		a
		proc
		ess whic
		h is
		curr
		ently
		holdi
		ng it
		and
		the
		reso
		urce
		can
		only
		be
		relea
		sed
		volu ntari
		ly by
		the
		proc
		ess
		holdi
		ng it.
		Circu
		lar
		Wait
		: A
		proc
		ess is
		waiti
		ng
		for a
		reso
		urce
		whic
L		

		h is
		curr
		ently
		held
		by
		anot
		her
		proc
		ess
		whic
		h in
		turn is
		waiti
		ng
		for a
		reso
		urce
		held
		by
		the
		first
		proc
		ess.
		In
		gene ral
		ther
		e
		exist
		s a
		set
		of
		waiti
		ng
		proc
		ess
		P0, P1
		Pn
		with
		P0 is
		waiti
		ng
		for a
		reso
		urce
		held
		by D1
		P1 and
		P1 is
		waiti

				, ,
				ng for a reso urce held by PO, ,Pn is
				waiti ng for a reso urce held by PO and PO is waiti ng for a reso urce held by Pn and
				[3]
				[3]
				[4]
				CO2
				L4
3	With FSM model, discuss the design and operation of automatic tea/coffee vending machine. Solution:	[10] [4] [6]	C O 1, C O 2	L2,L3



The FSM representation for the above requirement is shown in

the below figure. It contains four states namely

- i. Wait for coin
- ii. Wait for User Input
- iii. Dispense Tea
- iv. Dispense Coffee.

The event 'Insert Coin' (5 rupee coin insertion), transitions the state to 'Wait for User Input'. The system stays in this state until a user input is received from the buttons 'Cancel', 'Tea' or 'Coffee'.

If the event triggered in 'Wait State' is 'Cancel' button press, the coin is pushed out and the state "transitions to 'Wait for Coin'.

If the event received in the 'Wait State' is either "Tea' button press, or 'Coffee' button press, the state changes to 'Dispense Tea' and 'Dispense Coffee' respectively.

Once the coffee/tea vending is over, the respective states transition back to the 'Wait for Coin' state.

A few **modifications** like adding a timeout for the 'Wait State' (Currently the 'Wait State' is infinite; it can be re-designed to a timeout based 'Wait State'. If no user input is received within the timeout period, the coin is returned back and the state automatically transitions to 'Wait for Coin' on the timeout event) and capturing other events like, 'Water not available', 'Tea/Coffee Mix not available' and changing the state to an 'Error State' can be added to enhance this design.

Define Dungage Fundain in detail the atmost one arrangement of the state of	[10]	CO	L4
Define Process. Explain in detail the structure, memory organization and state		2	
transition of the process.			
Solution:			
Process	[4]		
A 'Process' is a program, or part of it, in execution.	[6]		
Process also known as an instance of a program in execution.			
Multiple instances of the same program can execute simultaneously.			
A process requires various system resources like CPU for executing the process;			
memory for storing the code corresponding to the process and associated variables,			
I/O devices for information exchange, etc.			
The Structure of a Process			
The concept of 'Process' leads to concurrent execution (pseudo parallelism) of tasks			
and thereby the efficient utilisation of the CPU and other system resources.			
Concurrent execution is achieved through the sharing of CPU among the processes.			
A process mimics (imitate) as a processor in properties and holds a set of registers,			
process status, a Program Counter (PC) to point to the next executable instruction of			
the process, a stack for holding the local variables associated with the process and the			
code corresponding to the process.			
This can be visualised in below figure			
A process which inherits all-the properties of the CPU can be considered as a virtual			
processor, awaiting its turn to have its properties switched into the physical			
processor.			
When the process gets its turn, its registers and the program counter register mapped			
to the physical registers of the CPU.			
A process which inherits all-the properties of the CPU can be considered as a virtual			
processor, awaiting its turn to have its properties switched into the physical			
processor. When the process gots its turn, its registers and the program counter register manned			
When the process gets its turn, its registers and the program counter register mapped to the physical registers of the CPU.			
rom a memory perspective, the memory occupied by the process is segregated into			
three regions, namely, Stack memory, Data memory and Code memoryas shown in			
above figure.			
The 'Stack' memory holds all temporary data such as variables local to the process.			
Data memory holds all global data for the process.			
Process States and State Transition			
The creation of a process to its termination is not a single step operation. The process			
traverses through a series of states during its transition from the newly created state to			
the terminated state.			
The cycle through which a process changes its state from "newly created' to			
'execution completed' is known as 'Process Life Cycle'.			
The various states through which a process traverses during a Process Life Cycle			
indicates the current status of the process with respect to time and also provides			
information on what it is allowed to do next.			
- Below figure represents the various states associated with a process.			
he state at which a process is being created is referred as 'Created State'. The			
Operating System recognises a process in the 'Created State' but no resources are			
allocated to the process.			
The state, where a process is incepted into the memory and awaiting the processor			
time for execution, is known as 'Ready State'. At this stage, the process is placed in			
the 'Ready list' queue maintained by the OS.			
The state where the source code instructions corresponding to the process is being			
executed is called 'Running State'. At which the process execution happens.			
Blocked State/Wait State refers to a state where a running process is temporarily			
suspended from execution and does not have immediate access to resources.			
The blocked state might be invoked by various conditions like: the process enters a			

wait state for an event to occur (e.g. Waiting for user inputs such as keyboard input) or waiting for getting access to a shared resource A state where the process completes its execution is known as 'Completed State'. The transition of a process from one state to another is known as 'State transition'. When a process changes its state from Ready to running or from running to blocked or terminated or from blocked to running, the CPU allocation for the process may also change.			
With a neat diagram, explain operating system architecture.	[10]	со	L1,L3
Solution:		2	
The <i>Operating System (OS)</i> acts as a bridge between the user applications/ tasks and the underlying system resources through a set of system functionalities and services.	[4] [6]		
» The primary functions of an operating systems are – » Make the system convenient to use			
» Organize and manage the system resources efficiently and correctly. The following Figure gives an insight into the basic components of an operating system and their interfaces with rest of the world.			
 The Kernel: The kernel is the core of the operating system. It is responsible for managing the system resources and the communication among the hardware and other system services. 			
» Kernel acts as the abstraction layer between system resources and user applications.			
 » Kernel contains a set of system libraries and services. » For a general purpose OS, the kernel contains different services like memory management, process management, time management, file system management, 			
I/O system management. Process Management: deals with managing the process/ tasks.			
 Process management includes – setting up a memory for the process 			
» setting up a memory for the process» loading process code into memory			
» allocating system resources			
» scheduling and managing the execution of the process			
 senteduring and managing the execution of the process setting up and managing Process Control Block (PCB) 			
» inter process communication and synchronization			
» process termination/ deletion, etc.			
Primary Memory Management: refers to a volatile memory (RAM), where			
processes are loaded and variables and shared data are stored.			
» The Memory Management Unit (MMU) of the kernel is responsible for –			
» Keeping a track of which part of the memory area is currently used by which process			
 Allocating and De-allocating memory space on a need basis. 			
File System Management: File is a collection of related information. A file could be a			
program (source code or executable), text files, image files, word documents,			
audio/ video files, etc.			
» A file system management service of kernel is responsible for –			
» The creation, deletion and alteration of files			
» Creation, deletion, and alteration of directories			
» Saving of files in the secondary storage memory			
» Providing automatic allocation of file space based on the amount of			
free running space available			
» Providing flexible naming conversion for the files.			
I/O System (Device) Management: Kernel is responsible for routing the I/O requests			
coming from different user applications to the appropriate I/O devices of the			
system.			

Illustrate the concept of "deadlock" with a neat diagram. Mention the different conditions favor a deadlock situation. Solution:			
Illustrate the concept of "deadlock" with a neat diagram. Mention the different conditions favor a deadlock situation. Solution: Deadlock is the condition in which a process is waiting for a resource held by another process which is waiting for a resource held by the first process; hence, none of the processes are able to make any progress in their execution. Process A holds a resource "x" and it wants a resource "y" held by Process B. Process B is currently holding resource "y" and it wants the resource "x" which is currently held by Process A. Both hold the respective resources and they compete each other to get the resource held by the respective processes. Conditions Favoring Deadlock: Mutual Exclusion: The criteria that only one process can hold a resource at a time. Meaning processes should access shared resources with mutual			
 Hold & Wait: The condition in which a process holds a shared resource by acquiring the lock controlling the shared access and waiting for additional resources held by other processes. No Resource Preemption: The criteria that Operating System cannot take back a resource from a process which is currently holding it and the resource can only be released voluntarily by the process holding it. Circular Wait: A process is waiting for a resource which is currently held by another process which in turn is waiting for a resource held by the first process. In general there exists a set of waiting process P0, P1 Pn with P0 is waiting for a resource held by P1 and P1 is waiting for a resource held by P0,,Pn is waiting for a resource held by P0 and P0 is waiting for a resource held by Pn and 	[10] C	CO 2	L4