Distributed Systems Jecture 3	LineariPzability Announcement : THE SCHEDULE (not the notes) ARE AUTHORITATIVE
- Came up in - Today's topic	venties; happens bejone; etc. last class. might maise similan questions motivating lonace propenties but
Describe dist. protocols Modeling, etc. A specipe As building	Specifying What they do? Happens-bebre; Trace traces Properties Common type of Gonrectness propents Common type of Gonrectness propents (Lincorrability)

fault tolerant sy stems CONSISTENCY One system implementing a queue. ALICE Q eng(A) OP ABC ? In what order CAB S do operations get performed? eng(B) Note. What consistency model O eng(c) Oh Is a choice made by the protocol CAM lesignen/distributed system builder !! A connectness requirement - Safety on Liveness When making the choice, need to balance (TRADE OFF BETWEEN) -tase of use La Do programmers need to be extra Cautions when using -> To users understand the behavior Performance ? Tied to each other.

-Fault Holenance)
How to think about an ABSTRACT DATA TYPE
Q1. Why is there a DATA TYPE in this DISTRIBUTED Sys. class?
$Cng(x) = \begin{bmatrix} 0 & pash(1) = \\ pop(1) & pop(1) \\ \vdots & \\ cng(y) = \end{bmatrix} $ Stack
Q20 What is so abstract about this?
NIR SHAVITS ~ 2011. (Older wisdom)infinitely easier and more intuitive for us humans to specify how abstract data structures behave in a sequential setting, where there are no interleavings.
Specify sequential AND Mapping from <u>concurrent</u> to <u>sequential</u> .

...

ALICE)b Peng(A) f eng(B) ∩ P°)h Sequentral Specification - Behavion without onconnergy Po -> Push (5) -> push (6) -> push (6) -> push (7) -> popC))h 1) Exactly one operation at a time. 2 Wait for response before issuing next operation 3) Says what each operation can netwin.

What we are used to: Queues are FIFO, Stadis are ZIFO, Text editors ... Mapping from Concurrent Execution → PUSH(2) ← OK() → PUSH(3) ← OKC7 -> PUSH(7) OR Stack FOK() ->POP() > Pop ())h 1) Operations complete between invocation (inv, and snesponse (snesp, <) PROGRAMMER 2] a total order that a All users an agree on (b) Is consistent with seq. specification ine anizability

Now more pormalls
(a) Notation from paper
object operation process: Invocation object OK(value) process: Return
Remember, a process has at most 1 operation in progress at a time.
(b) < _h Partial Order
$OP_1 < OP_2 \implies Return OP_1$ before invocation OP_2
Operations complete between invocation $(inv, \rightarrow,)$ and subsponse $(resp, <)$
L> op should observe effects morn op
But the total order ($_{\circ}$) we will look for ane SUPERSETS Of \leq_{H}
Observe < , orders all operations issued by

the same process.	• •	• •	0	• •	• •	•
· · · · · · · · · · · · · · · · · · ·	• •	• •	•	• •	• •	•
Why?	• •	• •	0	• •	• •	•
	• •	• •	•	• •	• •	•
	• •	• •	•	• •	• •	0
spush(z) a spush(z) a sok(z) b	• •	• •	•	• •	• •	•
	• •	• •	•	• •	• •	•
Spush(3) d. Sok(3) D.	0 0	• •	•	• •	• •	0
Jok()a Jpop()b Jok()a	• •	• •		• •	• •	•
······································	• •	• •	٠	• •	• •	•
5 pop C/C	• •	•••	•	•••	• •	•
$OPL L_{4}OP3 OP3 $	0 0	• •	•	• •	0 0	•
	• •	• •	٠	• •	• •	•
• • • • • • • • • • • • • • • • • • •	• •	• •	•	• •	• •	•
· · · · · · · · · · · · · · · · · · ·	• •	• •	•	• •	• •	0
$opl <_{4} op2 op3$	• •	• •	•	• •	• •	•
$\cdots \cdots $	• •	• •	0	• •	• •	0
	• •	•••	•	• •	• •	•
	• •	• •	•	• •	• •	•
OP1 OP3	0 0	• •	•	• •	• •	•
Spush(2) a OP2 Spush(3)	• •	• •	•	• •	• •	•
$S push(2) = OP_2 $ $S push(3) b$	• •	• •	•	• •	• •	•
$\int s_{POP}(c) b \qquad $	• •	•••	•	• •	• •	•
				• •	• •	•
	• •	• •	•	• •	• •	•
SOK(2) D SOK() D	• •	• •	•	• •	• •	•
	• •	• •	•	• •	• •	0
· · · · · · · · · · · · · · · · · · ·	• •	• •	•	• •	• •	•
Opl op2 <14 Op3	• •	• •	0	• •	• •	•
· · · · · · · · · · · · · · · · · · ·	• •	• •	•	• •	•••	•
	• •	0 0	0	• •	• •	•
C History H linearizable if and only	4	• •	•	• •	• •	•
(c) History H linearizable 4	• •	0 0	•	• •	• •	
	• •	• •	0	• •	• •	•
	• •	d'	-	· MAT	ie	•

I H that extends H (by ינט קי neturns) s.t. these is total order <s where < complete (H') <= <s De < 5 meets SEQUENTIAL SPEC. complete (H): H with all incomplete / inprogress operations removed OV 6 Pust (2) a 0P2 s push(3)) Spop() b 5 OK(2) D 5 OK() Operations in COMPLETE (H)? Op2 opl s push(2) a Spush(3) a LOK(3) b 15 OK()a J. ok()a. 5 pop C)b Operations in COMPLETE (H)? Llo i and only if

History H linearizable 4 Ø on more I H' that extends H (by netwins) sito these is total order $<_{\rm S}$ where $<_{\rm Complete(H')} \leq <_{\rm S}$ & < meets SEQUENTIAL SPEC < Komplete (H1) opl 5 push(2) a OP2 5 pop() b s push(3) / OK()a 5 0K(2) D 5 0K() LS = 01, L, 01, L, 0B Is It a linearizable history for a stack? OP OP3 5 push(2) a op_ \$ push(3) b Spop() b D OK() b 5 OK(2) D

 $<_{s}$ $H \rightarrow H' \qquad H' \rightarrow \text{Complete}(H')$ OPS OPL b push(2) a opr s proh(3) spop() b 5 OK(2) D K Gle OP2 OPl a spush(3)D D push(2) op3 SPOPC) 6 * OK(2)b JOK() D OK. BUT ARE THERE OTHER OPTIONS? Seq consistency Operations complete between invocation (inv, and suppose (resp. <-> 2)] a total order that (a) All LINED & an OGREE ON

	(b) Is consistent with seq. specification
	: Onder of ops from process P Equivalently <ир [ир: Only invocations A returns for process p]
History H	seq cot if and only if
	H' that extends H (by 0 on more networks) site there is total order $<_{s}$ where $<_{complete(H P)} \equiv <_{s} F_{P}$ $V <_{s}$ meets SEQUENTIAL SPEC
- (omm	non o Strongest guarantee provided by Ctr, Rust, etc.
- Might	appear that operations more in time
· · · · · · · · · · · · · · · · · · ·	push(2) a spush(1)b spop()a 1 1

soking soking sok(2)a - Lincarizable? - Seg cst ? nopenties Non-Blocking If history H is linearizable and H contains incomplete op O then I Mesponse of to O such that H; OR is linearizable OP3 OPL Spush(2) a OPspush(3) SPOPC) b 5 OK(2) 0 Two incomplete operations op 1, op 3

Why	is this use ful?
	- Can delay operation execution without violating consistency
	AllCE AllCE AllCE Det() Det()
	engles bos ret
	CAM engles
local	
	H has operations for two objects x and y
(popC)	a H linearizable \iff HIX linearizable b UL linearizable

rjusui in Fild	•••
What this means?	• •
- The linearization of H (Sori <s) matched<br="">x's seq spec D y's seq spec.</s)>	• • • • • •
- Cannot say anything about a data structure invoke push(2) a with a different seq. spec that	
involce pash(3) B contained to ack-with-len E push involce Jet-ben() c struct stack-with-len E push return OK(1) c Struct stack x pop int len J get-bength 3	
- Why useful?	
Begin Tru C	· · ·
Laden	• • • • • • • •

•

				<u>،</u>	/																				÷											
•	•	•	•	•	•	•	•	•	•	•	•	•	•	-	•	•	•	•	•	•	•	0	• •	•	•		•	•	•	٠	•	•	٠	•	•	• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	•	•	•	•	•	•	•	• •
				•		•	•		•			•			•			•												•			•			
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	• •	•	0	•	•	•	•	•	•	•	•	•	• •
•	•	٠	•	۰	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	٠	•	0	• •	•	٠		•	٠	0	٠	٠	•	٠	•	•	• •
•	•	•	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	٠	•	•	• •	•	•		•	٠	•	٠	٠	•	٠		•	• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	• •	•	•	0	•	•	0	•	•	0	•		•	• •
	•			•			•			•		•	•		•	•		•			•			•	•			•								• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	• •	•	•	•	•	•	•	•	•	•	•	•	•	• •
•	•	•	•	۰	•	•	٠	•	•	•	•	٠	•	•	•	•	•	۰	•	٠	•	•	• •	•	•		•	۰	۰	٠	•	٠	۰	•	٠	• •
•	•	•	•	٠	•	٠	٠	•	٠	•	•	٠	٠	•	•	•	•	•	•	٠	•	•	• •	•	٠		٠	٠	٠	٠	٠	•	٠	٠	•	• •
•	•	•	٠	٠	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•		•	•	•	٠	٠	•	٠	•	•	• •
•	•	•		•	•		•	•	•	•	•	•	•	•	•	•	•	0	•	•	•	•	• •	•	•		•	•	•	•	•	•	•	•	•	• •
•		•	•	•	•	•	•	•	•	•	•			•	•	•	•	•				•		•			•	•							•	• •
		•													•					•																• •
-	-	-		_	_	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-			-	-	_	_	-	-	-	-	-	-	-	
•	-		•	•			•	•	•			•			9										•	•				•			•			• •
0	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	0	•	•	0	•	٠	0	•	•	•	• •
•	•	•		۰	٠	•	•	•	•	•	•	٠	•	•	•	•	•	•	•	٠	•	•	• •	•	۰		•	۰	•	٠	۰		٠	•	٠	• •
•	•	•	٠	٠	•	٠	٠	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	٠		٠	٠	•	٠	٠	•	٠	•	٠	• •
0	•	•	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	0	• •	•	٠	•	•	•	•	•	•	•	•	•	•	• •
•	•	•	•	•	•	•	•	•	•	•	•	•		•	•	•		•		•	•	•		•	•		•	•		•		•	•		•	• •
		•			•			•	•		•		•	•	•			•				•		•				•								
																								-												
	-	4		-		9	-	9	4	Ψ.	-	-	-	-			-	Ψ	-	-		-			•		-	-	9	ų.		9	ų.	4	U .	- •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	• •	0	٠	•	•	٠	•	٠	٠	•	٠	•	٠	• •
0	•	•	•	۰	٠	•	٠	•	•	•	•	•	٠	•	•	•	•	•	٠	٠	٠	0	• •	•	٠	•	٠	٠	۰	٠	٠	•	٠	٠	•	• •
•	•	•	•	٠	•	•	٠	•	٠	•	•	•	•	•	•	•	•	•	٠	•	•	•	• •	•	٠	•	٠	•	•	٠	٠	•	٠	٠	•	• •
•	•	•		•	•		•	•	•	•	•	•	0	•	•	•	•	•	•	•	•	0	• •	•	•		•	•	•	•	•	•	•	•	•	• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•	•		•	•	•	•	•	•	•	•	•	• •
•	•	•	•	•		•	•	•	•	•		•	•		•			•		•	•			•				•	•	•		•	•		•	
																																				• •
•	•	•	۰	۰	•	۰	٠	•	•	•	•	۰	•	•	•	•	•	•	•	•	•	•	• •	•	۰	•	•	۰	٠	٠	٠	٠	۰	٠	٠	• •
0	•	•	•	۰	٠	•	•	٠	•	•	•	•	٠	•	•	•	•	•	•	•	•	•	• •	•	٠	•	٠	٠	•	٠	٠	•	•	٠	٠	• •
0	•	•	٠	٠	•	٠	٠	•	٠	•	•	•	•	•	•	•	•	•	٠	•	•	•	• •	•	•	0	٠	•	•	•	٠	•	•	•	٠	• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	•	• •	•	٠		•	•	•	•		•	٠		•	• •
•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•	•	• •
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•		•			•	•			•		•	•	•	•		•	•	•	•	• •
•																																				• •
																																				• •
0	•	•	•	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	٠	•	0	• •	•	•		•	٠	٠	•	٠	•	۰	٠	•	• •
•	•	•	٠	٠	•	٠	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	• •	•	٠	0	•	•	•	٠	٠	•	٠	•	•	• •
•	•	•		•	•		•	•	•	•	•	•	•	•	•	•	•	•		•	•	•	• •	•	•		•	•		•	•		•	•		• •
	•	•		•			•	•	•	•		•	•	•	•			•		•	•		• •		•		•	•		•			•		•	
•																																				
•	•	۰	۰	۰	٠	•	٠	۰	۰	•	•	٠	٠	•	•	•	•	•	٠	٠	٠	•	• •	•	٠	۰	۰	٠	۰	٠	٠	۰	٠	٠	٠	• •
•	•	•	٠	٠	•	٠	٠	•	٠	•	•	٠	٠	•	•	•	•	•	٠	•	٠	•	• •	•	٠	•	٠	•	•	٠	٠	۰	٠	٠	٠	• •

•	•	•	•	•		•	0	•	•	•	•	•	0	•	•	•	0 0	•	0	-74	7 ⁷	- 2	<u>).</u>	•	•	•	•	•	•	•	•	•		
•	0	0	•	•	•	۰	0	٠		0	0	٠	0	•	0	0	•	٠	•	4	•		•	•	•	•	•	0	•	•	•	D (• •	
•	•	•	٠	•	•	0	0	٠		0	٠	٠	: ((• ,	Ŷ	•	•	٠	٠	• •	٠	٠	•	•	•	•	•	•	•	•	•) (• •	
•	0	0		•	• •	0	0	٠		0	0	٠		<u> </u>	٢	•	• •	٠	٠	• •	٠		•	•	•	•	•	0	•	•	•		• •) 0
•	٠	•	•	•	•	•	•	٠	•	0	٠	۰	•	٠	•	•	•	٠	۰	• •	٠	۰	٠	•	•	٠	•	•	•	•	•) (• •) 0
•	•	•	۰	•	•	•	•	٠	•	0	٠	٠	٠	•	٠	•	•	٠	۰	• •	٠	۰	•	٠	•	•	•	•	•	•	•) (• •) 0
•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	\mathcal{L}		· · · · · · · · · · · · · · · · · · ·	•	$\dot{\mathbf{x}}$	• •		/	./	Ċ.	•	•	•	ำ	Ń	•	•		• •	
•	•	•	•	•		•	•	•	•		•	•	•	. ((\	() ($\boldsymbol{\gamma}$.)		\leq	•	(-Y		2	\mathcal{G}	ر '		•	•	• •		
	•	•	•	•	•	•	•	•		0	•		•	•	•					• •	•		•	•	. /	•	•	•	•		•			
•	•	•	•	•		•	•	•		0	•	•	•	•	•	•	• •	•	•	• •	٠	٠	•	•	•	•	•	•	•	•	•	D (• •) 0
•	•	•	•	•	• •	•	٠	٠	•	0	•	•	•	•	•	•	•	•	• •	4	٠	٠	ا ر	•	1.	•	•	-		•	•	ð (• •) 0
•	0	•		•	• •	0	0	٠		0	•	٠	0	٠	0	•	•	٠		[].	٠		٥٩	, C		10	•	. ۲		•	•) (• •	
•	٠	•	•	•	•	•	•	٠	•	0	٠	٠	٠	•	•	•	•	٠	•										•	•	•		• •) 0
•	•	•	٠	•	• •	٠	۰	٠	0	0	۰	۰	۰	٠	0	•	•	٠	٠	• •	0	• >	• •	.(` (ק	•	•	•	•	•) (• •	•
•	•	0	٠	•	• •	٠	۰	٠		0	۰	٠	٠	۰	•	•	• •	۰	٠	• •	٠	• 1	`• (• 6	/	•	•	•	•	•) (• •) 0
•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•		•	•	•••	•	•	•	•	•	•	•	•	•	•	•		• •	
	•							•	•	•									•		•		,											
•	•	•	•	•		•	•	•		0	•	•	•	•	•	•	(1)			ე .	•	/		./		2	. (4	$\widehat{}$	•	•	p (
•	•	•	•	•	• •	•	•	٠			•		•	•	•	•			2	!.	•	•	•	.(~	•	•		•	•	D (• •) 0
•	•	•	•	•	•	0	•	٠	•	0	•	٠	0	•	0	•	/	•	٠	• •	٠	٠	•	•	•	•	•	•	•	•	•	а (• •) 0
•	•	0	•	•	•	0	•	٠	0	0	٠	٠	•	•	•	•	•	•	٠	• •	٠	۰	•	•	•	•	•	•	•	•	•	Þ (• •	
•	٠	•	٠	•	• •	۰	۰	٠	•	0	٠	٠	•	٠	0	•	(1)	•	2.).	۰		• (2])	ŀ	•	•	•	•		• •) 0
•	•	0	٠	•	• •	٠	٠	٠	0	0	۰	٠	٠	٠	•	•	L I,) •	L	/ .	٠	٠		-	• 1	•	٠	•	•	•	•) (• •	•
•	۰	•	۰	•	• •	۰	۰	٠	•	0	•	٠	۰	•	•	•	• •	•	٠	• •	٠	٠	•	•	•	•	•	•	•	•	•) (• •	, •
•	•	•	•	•	• •	•	•	•	•	•	•	•	•	•	•	•		•	•	•••	•	•	•	•	•	•	•	•	•	•	•		• •	
																	•																	
•	•	•	•	•	• •	0	0	•	•	0	•	٠	0	•	0	•	• •	•	٠	• •	•	٠	•	•	•	•	•	•	•	•	•	D () 0
•	•	•	٠	•	•	۰	•	٠		0	•	٠	•	•	0	•	•	٠	•	• •	٠	٠	•	•	•	•	•	•	•	•	•	Þ (• •	
•	•	0	•	•	•	0	•	٠	0	0	۰	٠	٠	٠	٠	•	•	٠	٠	• •	٠	۰	٠	•	•	•	•	•	•	•	•		• •) 0
•	٠	•	•	•	•	•	۰	٠	•	0	٠	•	٠	•	•	•	•	٠	۰	• •	٠	۰	•	0	•	•	•	•	•	•	•) (• •) 0
																	•																	
																	• •																	
																	•																	
•	•	•	•	•	•	•	•	•		0	•	•	•	•	•	•	•	•	•	• •	•	٠	•	•	•	•	•	•	•	•	•	D () 0
•	•	0	•	•	•	٠	•	٠	0	0	•	٠	•	•	•	•	•	•	•	• •	•		•	•	•	•	•	•	•	•	•	a (• •) 0
•	•	•	•	•	• •	٠	٠	٠	٠	•	•	٠	٠	•	•	•	•	•	٠	• •	٠	•	•	•	•	•	•	•	•	•	•	•	• •	, .
•	•	0	•	•	•	0		٠		0	•	٠	0	•	0	•	•	٠	۰	• •	٠	٠	•	•	•	•	•	•	•	•	•		• •) 0
																	•																	
																	•																	
																	• •																	