	Unix V6 file system								
-	were ve gene age.								
	1000 blocks of 512 bytes								
	o not used								
	super-bloch								
	2] - noot directory								
40 blocks	3 for storing file descriptors (i-nodes)								
for i-list	+ directory information								
	41)								
data o	42								
bloch	43 for storing data blocks for files								
	and the free list								
937	999								
	relevant fields of an super block								
	i-size: # blocks in the i-list, here 40								
	n-fræ index into free								
	free [100]:								
<u> </u>	0 1								
	free								
	block numbers of free blocks								
	containing next tree blocks set of free blocks								
	•								
	if=0, no more								
	free blocks								

Suppose there were only 20 data blocks and that the pee list was of length 5:

0	1,416171819
1	2,4,12,13,1415
2	3,4,891011
3	4,4,4567
4	0,40123
5	in in
6	
チ	
19	

11st entry = block contains next set

of 4 free blocks

= 0, then no more of them

free list

= # pee blocks in the list, infree

list of whee brocks

so free[5]

nfree = 4

how to allocate a file f

f Suppose 720 bytes in site since each block is 512 bytes need 2 blocks

copy bo of f into block 19 free[3] = 18

copy b 1 of f into block 18, nfree = 2

now allocate file g of pare 1224 bytes

=> need 3 blocks

copy bo of g into free[2] = block 17, npee = 1

copy b1 of g into free[1] = block 16, hpee = 0

since refree = 0 => block free[1] = 1

contains next pre list

update

Free 2 12 13 14 15

nfree = 4

now allocate b2 of g uto free[4] = bloch 15 nfree = 3 i-nudes (file descriptors) + directories i-numbers begin at 1; storage starts in block 2

1-nodes are 32 bytes long so. $(32 \times 16 = 512)$ 16 of them fit in one block

root diectory in block 2

each i-	node	repr	esent	one	file	-		to correct
		,			3 - 2		plain f	, le
flags		fla	p on	file	8.	φ	007	00
hlinks				to file		0		
uid		user id of owner say user 1						
gid		group id of owner = 1224 bytes						
5'17e0		3 2	1 bits	for A	le siz	ح	- 1	224 bytes
S12e 1	.80							
addr	Ø	1	2	3	4	5	6	7
achne	O	1	hm	e 0/	last	- ac	Class	A
modhne	O	1						cation
				•				

will only deal with small files (\$ X 512b)

3 so addr [i), 0 = i = 7 are

drict address blocks, i.e, give

block # of file contents

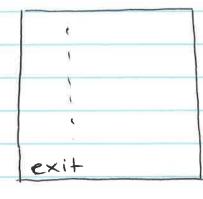
	de a cha
	due ctory entries
	indicated by fear bit of i-node entry
	directory entries are 16 bytes long \$ 32 in one
	i-number file name
	of wit is the file number
	The cross of the second of the
=	by convention 1st 2 entries in each directory
	are and or parent
	parent
	Current
	in block 2 is the noot directory
	1.2, can contain 16 i-nodes
2	flags has 140777 riwin works adar away for a directory is
	contents of the duectory, i.e., if
	addr (0) = 57 then block 57
	Contains derectory entires (up to 32 of them
	Size
	0
	1 0
	2 /
	3 1 ;

- main thread

ubfs dish k

initializes the file system spawns k thread lack thread i, 1 = i = k

file useri, +x+



sequence of commands

- file system is allocated in memory -work on it in memory

14, since 0,1 reserved for o,00)

	AF	ter file	system is ini	ha	lize	dit	will e	ontae		
block	0		unused							
	ł		superblock	superblock						
	2		root director	У	-					
^		e	* 1	1	انحاك	=40	super	block	~	
for i-list		;		-	ž.					
	41		,		nfre	e = 100				
	42		_		free	[100]	-	ialized		
for	, =	i i			* E			fice blo nbers		
data +								c[o] to		
Tree its	999									
	atoma t									
		i_ 1	111 1 2							
			array of 16 i-nodes onto block 2							
	OV	erlay an	array of 16.	(- / (ode	2 01	18 8130	~ _		
	C1	C1	1110777	1						
		1	140777	70	- 1			4		
	$\langle \ \ $		· 0 = superuser	3	2 64.	tcs	this 1'-		26	
index position Ø		3,120,	3 (3 entries)				is for directo		701	
Positive		addr 19	97	1				#		
)		Suppo	se bloc	L 999		
						alloc	ated to	hold		
15 - 11 - 5	{	ž		-	0	1.	•		a	
15 other		1	^		1	invalid for/	• •		block	
entries		16			2	1	1		can	
to 611					8	,			contai,	
the 512		,				,			director	
byte		*			15	U-no	file no	ane	entries (well	
	~			1	13	10. 0.00	0.4		(WEII	

	First, suppose a user 1 creates a directory							
	d1; since the current directory is.							
	initialized as 1, d1 gets created in 1							
	Jo create a directory: 1) Create an entry in the current directory for it							
	=> into block 999 (holdery 1)							
		block	999					
also	0	1 0						
update s	1201	-1 •	ø	assu	ne i-n	unbe	2=-1	
of / 1-1 to be 4		1			means	ľΛ	valid	
2.	thes are 3 2 d1						up from/	
Set)								
	2) Cre a	ate an	i-node	for	the di	rict	ory d1	
	2) Create an 1-node for the directory of 1 into block 2, add an 1-node for d1							
	at position 1							
		block						
	0 {	update 5	12e1 = 4		} i-no	de G	or /	
		flags	= 14077	7	7			
index	05.1	uid	uid = 1			i-node for d1		
		sizel	= 2 (2 entr	ies)				
	A)	adar	998					
		1				Suppo	se block	
14 oth	en (illocated for d	
i-node	()				0	2	a	
entri					-	1	00	
fill				die	ctory St		II.	
		_		dire is em	pty ?		L.	

Note that creating the directory allocated a block to hold entries for the directory (initialized with o, and oo)

Suppose user 1 now did cd to d1, and then executed "cpin f1,+x+ f1v6,+x+"

The col now has block 998 as the current directory. Hence the cpin will create the directory entry for the file here

block 998 (duectory d1)

0				
replate		2		(1) First create a
sizel of d		1	0 0	directory entry
to 3		3	F1V6, +x+	of or the file
			1	2) Create an i-node
	Į	,	£.	for the file.
		blo	CK 2	(suppose f1. +x+
1-node	Ø		for/	has 720 byts)
<i>y.</i>	1		to d1	
		F10	ups = 100777	block 997
inode fo	_	2	d = 1	15+ 5125 Hoan
flv6.txt	2 3		K1=720.	0 P1. +x+) 01
		1.	dar [997 996	block 996
				720-512 = (remaining
				208 bytes Stytes
				flubitet
				internal fragmentation

If move back up to / and create 1 entries in block 999, would need to allocate another block to directory entries and update addr [1] in block 2 (1st i-node in block 2)

For directories size! = number of directory entries in the block

For files, Size1 = the file size in lytes; sixe our max. file size is 4K, the sixe of field is unused.