

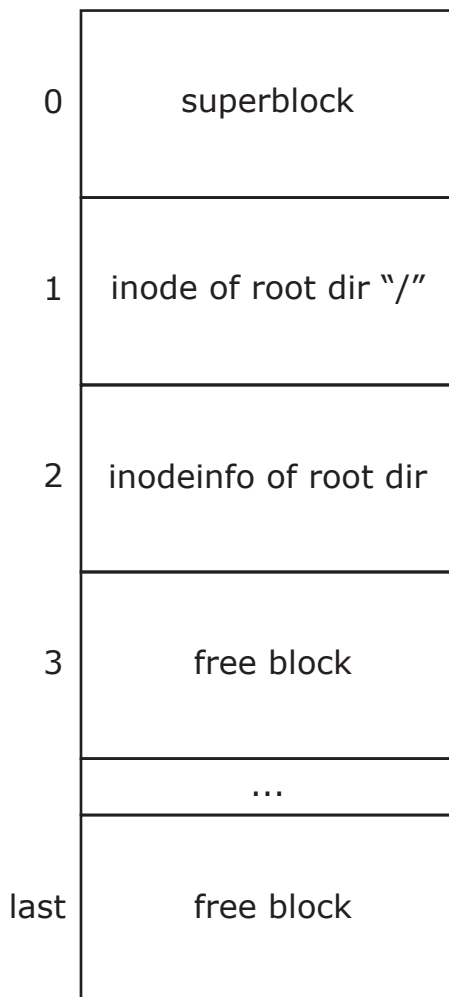
# DCC605 - Sistemas Operacionais

## TP2.B - Sistemas de Arquivos

Yuri Niitsuma  
Cassios Marques

### ORGANIZAÇÃO INICIAL DO SISTEMA DE ARQUIVOS

Após o **fs\_format** o arquivo de imagem contém no bloco 0 o superblock no 1 e 2 contém o **inode** e **inodeinfo** da pasta pasta raiz **/**.



E os blocos seguintes compostos por blocos livres que abordaremos no tópico posterior.

As figuras a seguir contém blocos de 128 bytes para ilustração.

O bloco 0 contendo o superblock

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
00000000	F5	05	C6	DC	00	00	00	00	00	00	02	00	00	00	00	00	00
00000010	80	00	00	00	00	00	00	00	F0	FF	01	00	00	00	00	00	00
00000020	0F	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
00000030	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000040	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000050	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000060	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	02	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
00000090	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	04	00	00	00	00	00	00	00	06	00	00	00	00	00	00	00	00
000000B0	0C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Blocos 1 e 2 contendo os inodes para a pasta raiz.

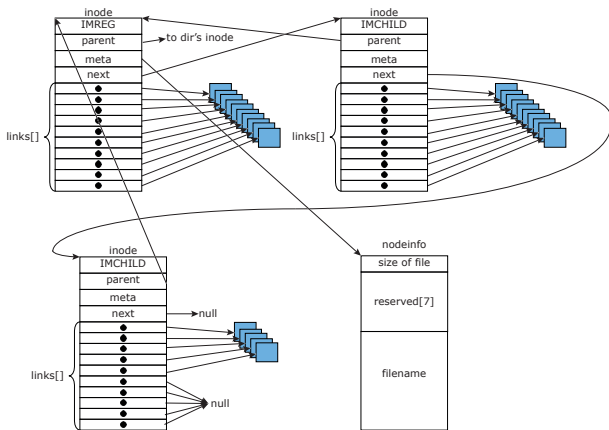
00000070	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000080	02	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
00000090	02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000A0	04	00	00	00	00	00	00	00	06	00	00	00	00	00	00	00	00
000000B0	0C	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000C0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000100	03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000120	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000140	2F	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000150	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000160	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000170	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000180	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000190	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000001B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00

O bloco na figura a seguir contém o inode + inodeinfo + 2 blocos data do arquivo respectivamente.

00000570	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000580	E3	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000590	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005C0	68	6F	73	74	73	00	00	00	00	00	00	00	00	00	00	00	00
000005D0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
000005F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000600	01	00	00	00	00	00	00	00	01	00	00	00	00	00	00	00	00
00000610	0B	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000620	0D	00	00	00	00	00	00	00	0E	00	00	00	00	00	00	00	00
00000630	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000640	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000650	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000660	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000670	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
00000680	31	32	37	2E	30	2E	30	2E	31	09	6C	6F	43	61	6C	68	
00000690	6F	73	74	0A	31	32	37	2E	30	2E	31	2E	31	09	45	6C	
000006A0	65	6D	65	6E	74	61	72	79	4F	53	0A	0A	23	20	54	68	
000006B0	65	20	66	6F	6C	6C	6F	77	69	6E	67	20	6C	69	6E	65	
000006C0	73	20	61	72	65	20	64	65	73	69	72	61	62	6C	65	20	
000006D0	66	6F	72	20	49	50	76	36	20	65	61	70	61	62	6C	65	
000006E0	20	68	6F	73	74	73	0A	3A	3A	31	20	20	20	20	20	69	
000006F0	70	36	2D	6C	6F	63	61	6C	68	6F	73	74	20	69	70	36	
00000700	2D	6C	6F	6F	70	62	61	63	6B	0A	66	65	30	30	3A	3A	
00000710	30	20	69	70	36	2D	6C	6F	63	61	6C	6E	65	74	0A	66	
00000720	66	30	30	3A	3A	30	20	69	70	36	2D	6B	63	61	73	74	
00000730	70	72	65	66	69	75	0A	66	66	30	32	3A	3A	31	20	69	
00000740	70	36	2D	61	6C	6C	6E	6F	64	65	73	0A	66	66	30	32	
00000750	3A	3A	32	20	69	70	36	2D	61	6C	6C	72	6F	75	74	65	
00000760	72	73	0A	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000770	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000780	10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
00000790	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

A organização dos arquivos segue como pedido na especificação:

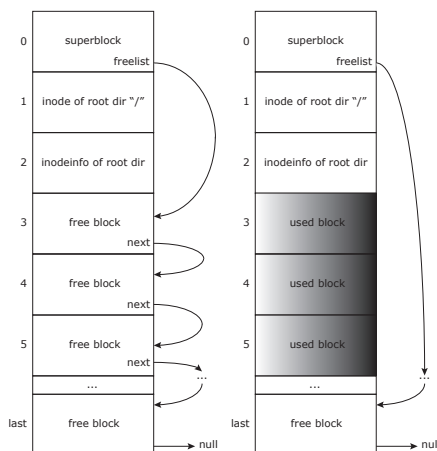


O campo **next** sendo utilizado para apontar para o próximo bloco inode IMCHILD e zero para indicar o último.

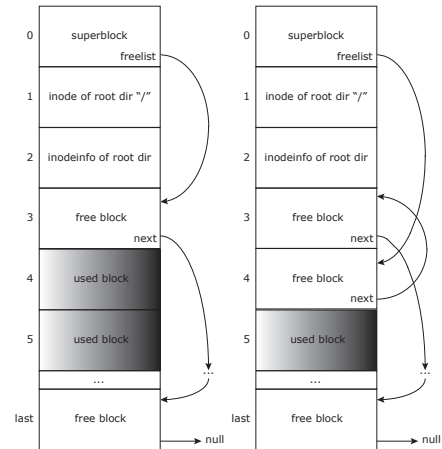
## ORGANIZAÇÃO DAS PÁGINAS DE BLOCOS LIVRES

Os blocos livres são organizados por uma lista encadeada em que cada um é apontado para o próximo bloco que está livre. Quando um bloco é liberado, ele entra no início da fila que é apontado pelo **superblock**.

Vemos um exemplo gráfico em que alguns blocos são retirados e depois liberados.



No exemplo o bloco 3, 4 e 5 são utilizados. Após o bloco 3 é liberado e por final o bloco 4 (que fica apontado como bloco livre inicial pelo **superblock**).



## TESTES

Além dos 6 testes padrão, incluímos um teste que se compõe em 3 partes:

- É criado o sistema de arquivos.
- Cria-se uma estrutura de pastas e insere arquivos pequenos e grandes.
- Extrai todo o conteúdo e é feito a comparação entre os arquivos.

Assim, conseguimos constatar que os arquivos são inseridos e extraídos corretamente.

```
so_tp2_b: make
bin/test 0 24 1024
DEBUG: FORMAT C:
DEBUG: return pointer for sb = 17667104
DEBUG: errno = 9
bin/test 1
DEBUG: Inserting files
DEBUG: file_name: /hosts
DEBUG: file_name: /bin/bash
DEBUG: file_name: /home/cassios/addonsu-arm-signed
DEBUG: file_name: /home/yuri/dhclientconf
bin/test 2
DEBUG: Extracting files "extract_files"
DEBUG: Extracting /hosts
-----
DEBUG: Extracting /bin/bash
-----
DEBUG: Extracting /home/cassios/addonsu-arm-signed
-----
DEBUG: Extracting /home/yuri/dhclientconf
-----
diff files/hosts extract/hosts
diff files/bin/bash extract/bin/bash
diff files/home/cassios/addonsu-arm-signed extract/home/cassios/addonsu-arm-signed
diff files/home/yuri/dhclientconf extract/home/yuri/dhclientconf
yuri@ElementaryOS:~/src/so_tp2_b$
```

O código do teste se encontra em:

[https://github.com/cassios/so\\_tp2\\_b](https://github.com/cassios/so_tp2_b)

Utilizando o comando *make test*.