

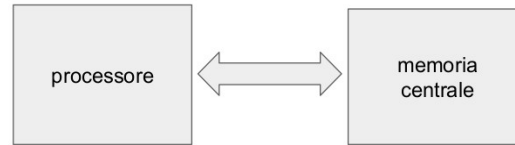
Laboratorio di Programmazione  
Corso di Laurea in Informatica  
Gr. 3 (N-Z)  
Università degli Studi di Napoli Federico II

A.A. 2022/23  
A. Apicella

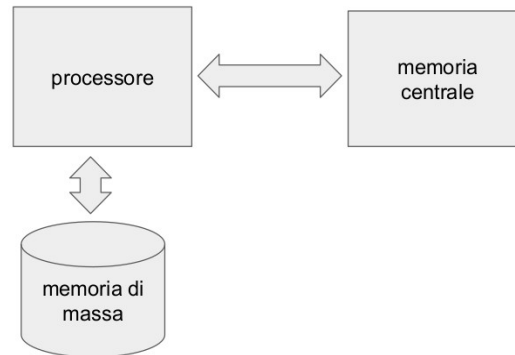
**I/O su file**

- Può capitare che un programma abbia necessità di leggere dati non da stdin (i.e. la tastiera), ma da file esterni. E.g., quando caricate il “salvataggio” di una partita ad un videogioco
- Quando un programma termina, dato che tutti i suoi dati sono in memoria centrale, questi vengono persi. Può essere necessario memorizzare dei dati su memoria di massa. E.g., quando salvate i dati di una partita ad un videogioco

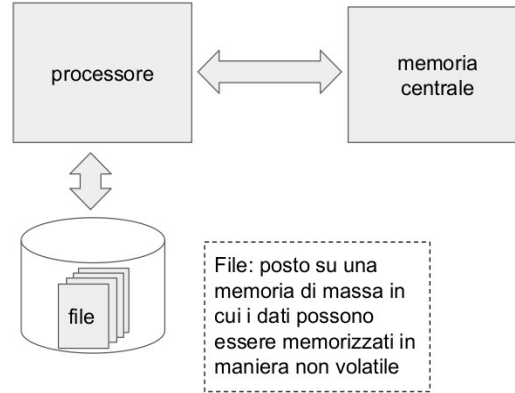
## I/O su file



## I/O su file



## I/O su file

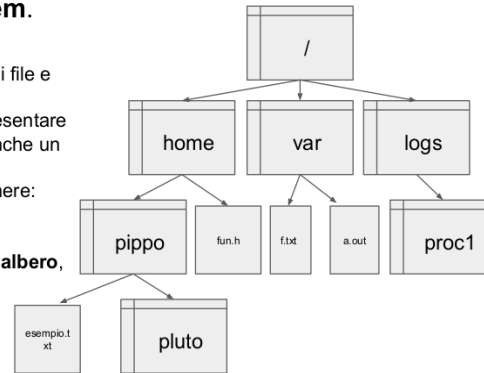


## File

L'insieme delle informazioni memorizzate sulla memoria di massa è organizzato secondo una struttura che detta **filesystem**.

- I sistemi Linux e Windows utilizzano i concetti comuni di file e directory (anche dette "cartelle" in Windows).
- Un file è un insieme di informazioni che possono rappresentare dei dati (un testo, un'immagine, una canzone, etc.) o anche un programma (ad esempio un file eseguibile)
- Una directory è una sorta di contenitore che può contenere:
  - file
  - altre directory

si viene quindi a delineare una struttura **gerarchica ad albero**, detto *directory tree* o *albero delle directory*.



perceived (pain), on pain's composite and

- la sequenza delle *directory* (cartelle, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

```
/home/pippo/esempio.txt
```

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (cartelle, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

/home/pippo/esempio.txt  
estensione

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (cartelle, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

`/home/pippo/esempio.txt`  
nome

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (cartelle, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

`/home/pippo/esempio.txt`  
separatore

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (*cartelle*, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

`/home/pippo/esempio.txt`  
prima directory da  
attraversare

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (*cartelle*, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

`/home/pippo/esempio.txt`  
seconda directory  
da attraversare

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path è composto da:

- la sequenza delle *directory* (*cartelle*, in Windows) da “attraversare” per trovare il file
  - ogni directory è separata da un carattere separatore (“/” nei sistemi Linux, “\” nei sistemi windows)
- il nome del file
- l'estensione estensione
  - preceduta da un “.”
  - generalmente (ma non più obbligatoriamente) di 3 caratteri
  - nei sistemi Windows, determina il tipo di file. Nei sistemi Linux la sua presenza non è obbligatoria, in quanto il tipo è ricavato in altro modo

Esempio:

/home/pippo/esempio.txt  
path

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path può essere:

- **assoluto**: comprende l'intera sequenza di directory da attraversare
  - su sistemi Linux, inizia sempre con “/”

Esempio:

/home/pippo/esempio.txt  
path assoluto

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path può essere:

- **assoluto**: comprende l'intera sequenza di directory da attraversare
  - su sistemi Linux, inizia sempre con "/"
- **relativo**: comprende le directory da attraversare *da un dato punto in poi*
  - su sistemi Linux, inizia con "."

Esempio:

`./pippo/esempio.txt`  
path relativo a partire dalla directory  
/home

## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path può essere:

- **assoluto**: comprende l'intera sequenza di directory da attraversare
  - su sistemi Linux, inizia sempre con "/"
- **relativo**: comprende le directory da attraversare *da un dato punto in poi*
  - su sistemi Linux, inizia con "."

Esempio:

`./esempio.txt`  
path relativo a partire dalla directory  
/home/pippo



## File

I sistemi operativi con file system ad albero identificano un dato file attraverso un percorso (*path*). Un path può essere:

- **assoluto**: comprende l'intera sequenza di directory da attraversare
  - su sistemi Linux, inizia sempre con "/"
- **relativo**: comprende le directory da attraversare *da un dato punto in poi*
  - su sistemi Linux, inizia con "." /" (in certi casi può essere omesso)
  - E' possibile utilizzare un path relativo a partire dalla directory in cui ci si trova
  - Per conoscere la directory in cui ci si trova, Linux mette a disposizione il comando `pwd` (*Print Working Directory*)
  - Si ricorda che, in Linux, è possibile muoversi attraverso le directory utilizzando il comando `cd` (*Change Directory*)
  - può contenere "." /" per indicare che è necessario "tornare indietro" di una directory.

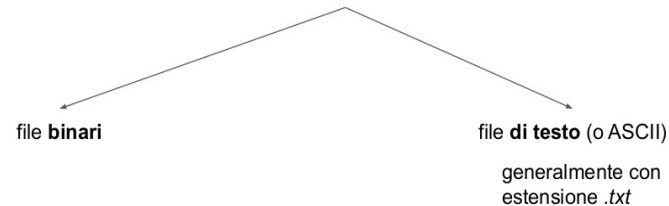
Esempio: ci si trova nella directory `/home/pippo`, e si vuole accedere al file `f.h` contenuto nella directory `/home/` è possibile accedervi tramite:

- path assoluto: `/home/f.h`
- path relativo: `../f.h` oppure `../f.h` oppure `../../home/f.h` oppure `../pippo../f.h`

## I/O su file

### Tipi di file

Il C è in grado di gestire 2 tipi di file:



## I/O su file

### File binari

il contenuto del file rispecchia fedelmente il contenuto della memoria centrale

```
int a = 369;
```

### memoria centrale

codifica binaria di 369 su 4 byte

00000000 00000000 00000001 01110001

### File binario

00000000 00000000 00000001 01110001

## I/O su file

### File di testo

il contenuto del file è la codifica ASCII di quanto contenuto in memoria centrale

```
int a = 369;
```

### memoria centrale

codifica binaria di 369 su 4 byte

00000000 00000000 00000001 01110001

### tabella ASCII

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	00		1	01		2	02		3	03	
4	04		5	05		6	06		7	07	
8	08		9	09		10	0A		11	0B	
12	0C		13	0D		14	0E		15	0F	
16	10		17	11		18	12		19	13	
20	14		21	15		22	16		23	17	
24	18		25	19		26	1A		27	1B	
28	1C		29	1D		30	1E		31	1F	
32	20	Space	33	21	!	34	22	"	35	23	#
36	24	\$	37	25	%	38	26	&	39	27	'
40	28	(	41	29	)	42	2A	*	43	2B	+
44	2C	,	45	2D	-	46	2E	.	47	2F	/
48	30	0	49	31	1	50	32	2	51	33	3
52	34	4	53	35	5	54	36	6	55	37	7
56	38	8	57	39	9	58	3A	:	59	3B	;
60	3C	<	61	3D	=	62	3E	>	63	3F	?
64	40	@	65	41	A	66	42	B	67	43	C
68	44	D	69	45	E	70	46	F	71	47	
72	48		73	49		74	4A		75	4B	
76	4C		77	4D		78	4E		79	4F	
80	50		81	51		82	52		83	53	
84	54		85	55		86	56		87	57	
88	58		89	59		90	5A		91	5B	
92	5C		93	5D		94	5E		95	5F	
96	60		97	61		98	62		99	63	
100	64		101	65		102	66		103	67	
104	68		105	69		106	6A		107	6B	
108	6C		109	6D		110	6E		111	6F	
112	70		113	71		114	72		115	73	
116	74		117	75		118	76		119	77	
120	78		121	79		122	7A		123	7B	
124	7C		125	7D		126	7E		127	7F	
128	80		129	81		130	82		131	83	
132	84		133	85		134	86		135	87	
136	88		137	89		138	8A		139	8B	
140	8C		141	8D		142	8E		143	8F	
144	90		145	91		146	92		147	93	
148	94		149	95		150	96		151	97	
152	98		153	99		154	9A		155	9B	
156	9C		157	9D		158	9E		159	9F	
160	A0		161	A1		162	A2		163	A3	
164	A4		165	A5		166	A6		167	A7	
168	A8		169	A9		170	AA		171	AB	
172	AC		173	AD		174	AE		175	AF	
176	B0		177	B1		178	B2		179	B3	
180	B4		181	B5		182	B6		183	B7	
184	B8		185	B9		186	BA		187	BB	
188	BC		189	BD		190	BE		191	BF	
192	C0		193	C1		194	C2		195	C3	
196	C4		197	C5		198	C6		199	C7	
200	C8		201	C9		202	CA		203	CB	
204	CC		205	CD		206	CE		207	CF	
208	D0		209	D1		210	D2		211	D3	
212	D4		213	D5		214	D6		215	D7	
216	D8		217	D9		218	DA		219	DB	
220	DC		221	DD		222	DE		223	DF	
224	E0		225	E1		226	E2		227	E3	
228	E4		229	E5		230	E6		231	E7	
232	E8		233	E9		234	EA		235	EB	
236	EC		237	ED		238	EE		239	EF	
240	F0		241	F1		242	F2		243	F3	
244	F4		245	F5		246	F6		247	F7	
248	F8		249	F9		250	FA		251	FB	
252	FC		253	FD		254	FE		255	FF	

codifica ASCII  
del carattere 3

codifica ASCII  
del carattere 6

### File di testo

00110011 00110110 00111001

codifica ASCII  
del carattere 9

## I/O su file

File di testo

memoria centrale

```
int a = 369;
```

il contenuto del file è la codifica ASCII di quanto  
contenuto in memoria centrale

codifica binaria di 369 su 4 byte

00000000 00000000 00000001 01110001

tabella ASCII

DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR	DEC	HEX	CHAR
0	00	NUL	9	09	TAB	18	12	SPACE	27	1B	ESC
1	01	SOH	10	0A	LF	19	13	CR	28	1C	FS
2	02	STX	11	0B	VT	20	14	SO	29	1D	GS
3	03	ETX	12	0C	FF	21	15	SH	30	1E	RS
4	04	END	13	0D	RI	22	16	SI	31	1F	US
5	05	START	14	0E	RD	23	17	DL	32	20	SP
6	06	ACK	15	0F	SD	24	18	CU	33	21	SP
7	07	NAK	16	10	LD	25	19	NU	34	22	SP
8	08	SYN	17	11	GD	26	1A	UH	35	23	SP
9	09	ETB	18	12	PD	27	1B	LS	36	24	SP
10	0A	CAN	19	13	DD	28	1C	SE	37	25	SP
11	0B	EM	20	14	UD	29	1D	FE	38	26	SP
12	0C	END	21	15	HD	30	1E	FF	39	27	SP
13	0D	CR	22	16	ND	31	1F	SS	40	28	SP
14	0E	LF	23	17	RD	32	20	TT	41	29	SP
15	0F	VT	24	18	SD	33	21	UU	42	2A	SP
16	10	RI	25	19	LD	34	22	VV	43	2B	SP
17	11	GD	26	1A	UD	35	23	WW	44	2C	SP
18	12	PD	27	1B	HD	36	24	XX	45	2D	SP
19	13	DD	28	1C	ND	37	25	YY	46	2E	SP
20	14	UD	29	1D	RD	38	26	ZZ	47	2F	SP
21	15	HD	30	1E	SD	39	27	[	48	30	SP
22	16	ND	31	1F	LD	40	28	\	49	31	SP
23	17	RD	32	20	UD	41	29	^	50	32	SP
24	18	SD	33	21	DD	42	2A	_	51	33	SP
25	19	LD	34	22	UD	43	2B	`	52	34	SP
26	1A	UD	35	23	HD	44	2C	aa	53	35	SP
27	1B	HD	36	24	ND	45	2D	ab	54	36	SP
28	1C	ND	37	25	RD	46	2E	ac	55	37	SP
29	1D	RD	38	26	SD	47	2F	ad	56	38	SP
30	1E	SD	39	27	LD	48	30	ae	57	39	SP
31	1F	LD	40	28	UD	49	31	af	58	3A	SP
32	20	SP	41	29	DD	50	32	ag	59	3B	SP
33	21	SP	42	2A	UD	51	33	ah	60	3C	SP
34	22	SP	43	2B	HD	52	34	ai	61	3D	SP
35	23	SP	44	2C	ND	53	35	aj	62	3E	SP
36	24	SP	45	2D	RD	54	36	ak	63	3F	SP
37	25	SP	46	2E	SD	55	37	al	64	40	SP
38	26	SP	47	2F	LD	56	38	am	65	41	SP
39	27	SP	48	30	UD	57	39	an	66	42	SP
40	28	SP	49	31	DD	58	3A	ao	67	43	SP
41	29	SP	50	32	UD	59	3B	ap	68	44	SP
42	2A	SP	51	33	HD	60	3C	aq	69	45	SP
43	2B	SP	52	34	ND	61	3D	ar	70	46	SP
44	2C	SP	53	35	RD	62	3E	as	71	47	SP
45	2D	SP	54	36	SD	63	3F	at	72	48	SP
46	2E	SP	55	37	LD	64	40	au	73	49	SP
47	2F	SP	56	38	UD	65	41	av	74	4A	SP
48	30	SP	57	39	DD	66	42	aw	75	4B	SP
49	31	SP	58	3A	UD	67	43	ax	76	4C	SP
50	32	SP	59	3B	HD	68	44	ay	77	4D	SP
51	33	SP	60	3C	ND	69	45	az	78	4E	SP
52	34	SP	61	3D	RD	70	46	[	79	4F	SP
53	35	SP	62	3E	SD	71	47	\	80	50	SP
54	36	SP	63	3F	LD	72	48	^	81	51	SP
55	37	SP	64	40	UD	73	49	_	82	52	SP
56	38	SP	65	41	DD	74	4A	aa	83	53	SP
57	39	SP	66	42	UD	75	4B	ab	84	54	SP
58	3A	SP	67	43	HD	76	4C	ac	85	55	SP
59	3B	SP	68	44	ND	77	4D	ad	86	56	SP
60	3C	SP	69	45	RD	78	4E	ae	87	57	SP
61	3D	SP	70	46	SD	79	4F	af	88	58	SP
62	3E	SP	71	47	LD	80	50	ag	89	59	SP
63	3F	SP	72	48	UD	81	51	ah	90	5A	SP
64	40	SP	73	49	DD	82	52	ai	91	5B	SP
65	41	SP	74	4A	UD	83	53	aj	92	5C	SP
66	42	SP	75	4B	HD	84	54	ak	93	5D	SP
67	43	SP	76	4C	ND	85	55	al	94	5E	SP
68	44	SP	77	4D	RD	86	56	am	95	5F	SP
69	45	SP	78	4E	SD	87	57	an	96	60	SP
70	46	SP	79	4F	LD	88	58	ao	97	61	SP
71	47	SP	80	50	UD	89	59	ap	98	62	SP
72	48	SP	81	51	DD	90	5A	aq	99	63	SP
73	49	SP	82	52	UD	91	5B	ar	100	64	SP
74	4A	SP	83	53	HD	92	5C	as	101	65	SP
75	4B	SP	84	54	ND	93	5D	at	102	66	SP
76	4C	SP	85	55	RD	94	5E	au	103	67	SP
77	4D	SP	86	56	SD	95	5F	av	104	68	SP
78	4E	SP	87	57	LD	96	60	aw	105	69	SP
79	4F	SP	88	58	UD	97	61	ax	106	6A	SP
80	50	SP	89	59	DD	98	62	ay	107	6B	SP
81	51	SP	90	5A	UD	99	63	az	108	6C	SP
82	52	SP	91	5B	HD	100	64	[	109	6D	SP
83	53	SP	92	5C	ND	101	65	\	110	6E	SP
84	54	SP	93	5D	RD	102	66	^	111	6F	SP
85	55	SP	94	5E	SD	103	67	_	112	70	SP
86	56	SP	95	5F	LD	104	68	aa	113	71	SP
87	57	SP	96	60	UD	105	69	ab	114	72	SP
88	58	SP	97	61	DD	106	6A	ac	115	73	SP
89	59	SP	98	62	UD	107	6B	ad	116	74	SP
90	5A	SP	99	63	HD	108	6C	ae	117	75	SP
91	5B	SP	100	64	ND	109	6D	af	118	76	SP
92	5C	SP	101	65	RD	110	6E	ag	119	77	SP
93	5D	SP	102	66	SD	111	6F	ah	120	78	SP
94	5E	SP	103	67	LD	112	70	ai	121	79	SP
95	5F	SP	104	68	UD	113	71	aj	122	7A	SP
96	60	SP	105	69	DD	114	72	ak	123	7B	SP
97	61	SP	106	6A	UD	115	73	al	124	7C	SP
98	62	SP	107	6B	HD	116	74	am	125	7D	SP
99	63	SP	108	6C	ND	117	75	an	126	7E	SP
100	64	SP	109	6D	RD	118	76	ao	127	7F	SP

codifica ASCII  
del carattere 3

File di testo

00110011 00110110 00111001

codifica ASCII  
del carattere 9

codifica ASCII  
del carattere 6

visualizzabile  
con un  
semplice editor  
di testo