

Fájlnemek: mi van egy névben?

PCLinuxOS Magazine – 2016. július

Írta: Paul Arnote (parnote)

Mind látjuk azokat. Mind találkozunk velük, valamikor a számítógépes életünk során. Azzal a rettegett, idegesítő fájjal, ami ellenáll a törlésnek, vagy nehezen kezelhető.

Ez különösen akkor válik kellemetlenné, amikor Windows, Mac és Linux felhasználók között osztasz meg fájlokat. Minden platformnak saját szabályai vannak a fájlnevekben megengedhető karakterekre vonatkozóan. Ami egy platformon fájlnevként elfogadható, a másikon tiltott lehet.

Nos, nézzük meg, hogyan bánjunk ezekkel az idegesítő fájlokkal és fájlnevekkel.

Az alapok érintése

Az „alapok” a használt platformtól függően változhatnak. Az alapvető szabályok megértéséhez az ASCII-táblát kell megismerni.

Észre fogod venni, hogy a Windows elnevezési megkötésekből van a legtöbb. Kezdjük azzal, mivel nagy eséllyel lehet szükséged megosztani fájlokat windows-os felhasználókkal.

Először, minden 0 és 31 (decimális érték) közötti ASCII-karakter tiltott. Ezután, az összes következő karakter sem engedélyezett:

- < (kevesebb mint)
- > (több mint)
- : (kettőspont)
- " (dupla idézőjel)
- / (törtjel)
- \ (vissza törtjel)

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	((72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051))	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[[123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135]]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookUpTables.com

A teljes ASCII tábla

csővezeték)
? (kérdőjel)
* (csillag)

És ha ez még nem volna elég, bizonyos kulcsszavakat a Windows magának foglal le és azok használata fájlnevként nem engedélyezett:

CON, PRN, AUX, CLOCK\$, NUL
COM1, COM2, COM3, COM4

(függőleges vonal, vagy

LPT1, LPT2, LPT3, LPT4 (LPT4 csak a DR-DOS bizonyos verzióiban)

LST (csak a 86-DOS-ban és a DOS 1.xx-ben)

KEYBD\$, SCREEN\$ (csak a multitaskos MS-DOS 4.0-ben)

\$IDLE\$ (csak a Concurrent DOS 386-ban, a Multiuser DOS-ban és a DR DOS 5.0-ban és fölötte)

CONFIG\$ (csak az MS-DOS 7.0-8.0-ban)

Fájlnemek: mi van egy névben?

128	Ç	144	É	160	á	176	☼	192	Ł	208	⋈	224	α	240	≡
129	ü	145	æ	161	í	177	☽	193	⌞	209	⌟	225	β	241	≠
130	é	146	Æ	162	ó	178	☿	194	⌠	210	⌡	226	Γ	242	≥
131	â	147	ô	163	ú	179		195	⌢	211	⌣	227	π	243	≤
132	ä	148	ö	164	ñ	180	†	196	–	212	⌥	228	Σ	244	∫
133	à	149	ò	165	Ñ	181	‡	197	‡	213	⌦	229	σ	245	∫
134	å	150	û	166	ª	182	‡	198	‡	214	⌧	230	μ	246	+
135	ç	151	ù	167	º	183	¶	199	‡	215	‡	231	τ	247	±
136	ê	152	ÿ	168	¿	184	¶	200	⌨	216	‡	232	Φ	248	°
137	ë	153	Ö	169	⌈	185	¶	201	〈	217	‡	233	⊖	249	·
138	è	154	Ü	170	⌋	186	¶	202	〉	218	⌈	234	Ω	250	·
139	ì	155	¢	171	½	187	¶	203	⌫	219	■	235	δ	251	√
140	î	156	£	172	¾	188	¶	204	⌬	220	■	236	∞	252	∞
141	ï	157	¥	173	¡	189	¶	205	=	221	■	237	φ	253	²
142	Ä	158	฿	174	«	190	¶	206	≠	222	■	238	ε	254	■
143	Å	159	₪	175	»	191	¶	207	±	223	■	239	∩	255	■

Source: www.LookupTables.com

A kiterjesztett ASCII tábla

És ha még ez sem volna elég, az NTFS fájlrendszer további fájlnevek használatát tiltja meg, mert az NTFS-fájlrendszerben belül használatban vannak.

\$Mft, \$MftMirr, \$LogFile, \$Volume, \$AttrDef, \$Bitmap, \$Boot, \$BadClus, \$Secure, \$Upcase, \$Extend, \$Quota, \$ObjId és \$Reparse

Ez elég lehet ahhoz, hogy feltedd a kérdést: vajon miért olyan népszerű a Windows a köznapi számítástechnikában?

A Linux-rendszereknél csak két karakter tiltott: a NUL (decimális 0 ASCII-kód) és a / (törtjel, 47-es decimális ASCII-kód). Noha az összes többi karakter használata engedélyezett, bizonyos karakterek

használata problémákat okozhat, hosszabb távon. Erről később.

A Mac-rendszereknél csak a : (kettőspont, decimális 58-as ASCII-kód) tilos a fájlnevekben.

Nem állom meg, hogy ne ismétljem: mindez akkor lesz fontos, amikor olyan fájlokat készítesz, amiket „más” rendszerek használóival osztasz meg. Rossz karakter (vagy foglalt kulcsszó, vagy belső fájlnev) használata és a másik rendszer felhasználója nem lesz képes elérni a fájlot.

Biztonságos választások

Ha a fájld keresztplatformos elérhetőségét biztosítani szeretnéd, akkor néhány követendő szabály.

[0-9a-zA-Z_] - Alfánumerikus karakterek és az alsó vonal használata rendben van.

V:*?"<>| és a nulla byte problémás, legalább egy rendszernél és szintén kerülendő.

A szóköz Több fájlrendszerrel argumentum elválasztó, így a szóközt tartalmazó neveket kerüljük, ha lehet.

A kettőspontok (:) parancselválasztók több rendszerrel.

[]^#%&!@:+={}'~ és a [] több héjnál különleges jelentéssel bír és fárasztó kikerülni, így kerülni kell ezeket. Amellett URL-kben borzasztó látványt nyújthatnak.

Kerülendő kezdő karakterek: sok parancssori program a **kötőjelet** [-] különleges argumentum jelzésére használja, ugyanakkor a *nix alapú rendszerek a **pontot** [.] használják kezdő karakterként a rejtett fájlok és könyvtárak jelölésére.

Minden, az ASCII karakterkészletbe nem tartozó elem gondot okozhat a régebbi, vagy sokkal alapszintűbb rendszereknél (pl. néhány beágyazott rendszerrel) és óvatosan használandó.

Így alapvetően csak a [0-9a-zA-Z-.,_] marad, ami biztonsággal és kellemetlenségek nélkül használható (ameddig a fájlnev alfa-numerikusan kezdődik).

Problémás fájlnevek kezelése Linux alatt

Még ha a Linux fájlrendszer a NULL és / kivételével valamennyi karakter használatát engedi, nem jelenti azt, hogy nincsenek olyan karakterek, amik kényelmetlenséget és gondot okozhatnak, amikor olyan fájlnevekkel kell dolgozni, amik azokat a bizonyos karaktereket tartalmazzák. Tetszik, vagy sem a parancssor, és kerülöd-e vagy sem a parancssort, mint a bubópestist, vagy élvezed az erejét, egyszer mind olyan helyzetbe kerülünk,

amikor használjuk. Mint olyannal, érdemes megtanulni, hogyan kezeljük az ilyen fájlokat, különösen, amikor a kényszer visz rá a parancssor használatára (mint amikor az asztali környezet nem tölt be rendesen és a parancssoron keresztül kell javításokat eszközölnöd). Néhány GUI program nem kezeli könnyen ezeket a problémás fájlneveket, mivel néhány GUI program nem más, mint a parancssori változathoz (elé) ragasztott grafikus felület.

Szóközoeket tartalmazó fájlnevek. Tudom, korábban már többször is beszéltünk erről a magazinban, de nem árt ismételní – újból. Úgy tünik a régi windowsos megszokás lassan vész ki. Linux fájlók nevében kerülni kell a szóközt. A parancssorban a szóköz a parancssori opciók elválasztására használatos. Miként azt korábban is mondtam, ez ellen még néhány GUI sincs védve. Néhány GUI program nem más, mint ugyanazon program parancssori verziójának grafikus felülete. Ezen esetekben a szóközoek nem fognak megfelelően működni, különösen, ha a GUI tervezője nem foglalkozott a szóközoeket tartalmazó fájlnevek kezelésével. Ha úgy érzed, hogy egyszerűen KELL valami a fájlnevek „szavai” közé, sokkal jobb akár alsó vonás, akár kötőjel használata szóközoek helyett.

Szerencsére a szóközoeket tartalmazó fájlnevek kezelése a legegyszerűbb dolgok közé tartozik. Egyszerűen dupla idézőjelek közé tedd a fájlnevet. **(Megjegyzendő:** ugyanez használatos Windows alatt is, ha a Windows parancssorát használod!) Alternatívaként „átugorható” a szóköz a szóközoek előtt kirakott visszafelé tört jellel (\) (csak *nix-re igaz ... Windowsnál nem érdemes ez utóbbi megoldással próbálkozni, mivel azt a Windows a könyvtárak és alkönyvtárak jelölésére használja).

Kötőjeles fájlnevek (-). A kötőjel nem jelent problémát, hacsak nem a fájlnev első karaktereként szerepel. Az első karaktert kivéve, a fájlnevben bárhol megjelenve semmi sem kell tenni velük. Használd úgy, ahogy van. A probléma akkor kezdődik, amikor a kötőjel a fájlnev első karaktere.

```
[parnote-toshiba@parnote-toshiba July2016]$ nano -myfile.txt
nano: invalid option -- 'y'
Type 'nano -h' for a list of available options.
```

A parancssorban a kötőjel jelöli a parancssori opció, vagy kapcsolót. Amikor a fájlnev első karaktere kötőjel, akkor a parancssori héjad úgy értelmezi a kötőjel utáni részt, mint egy parancssori opció, vagy kapcsolót. Amennyiben egyik ismert opcióval sem egyezik meg, akkor a parancs sikertelen lesz, l. a lenti ábrát.

Az ilyen probléma kiküszöbölésére két lehetőség van. Első, hogy dupla kötőjelet teszel a fájlnev elé, mint itt: **nano -- -myfile.txt**. Alternatívaként indítható a fájlnev ./ jellel, mutatván, hogy a fájl az aktuális könyvtárban található (feltételezve, hogy éppen a fájl tartalmazó könyvtárban vagy), mint ennél: **nano ./-myfile.txt**.

Fájlnevek kettős kereszttel (#). Minthogy a BASH a # után mindent megjegyzésként kezel, a kettős kereszt jelzés problémát okozhat függetlenül a használata helyétől a fájlnevben. Amennyiben az elején van (#myfile.txt), mindent a # után megjegyzésnek tekint. Ha a kettős kereszt jele valahol a fájlnevben található (my#file.txt), akkor a # (my) utáni részt veszi megjegyzésnek. Egyik esetben sem fogja a fájl megtalálni.

Két megoldás lehetséges. Miként azt a kezdő karakterként használt kötőjel esetén, a fájlnev elé rakható ./ jelezvén, hogy a fájl az aktuális könyvtáron belül található, mint ebben az esetben: **nano ./#myfile.txt**. Alternatívaként, a fájlnev sima idézőjelek közé rakható, mint itt: **nano '#myfile.txt'**.

Fájlnevek pontosvesszővel (;) Ezek a fájlnevek is problémásak lehetnek. A BASH-nél (és vélhetően a többi parancssori héjnal szintén), a pontosvessző több parancs összekapcsolásához használatos. Tehát, amikor a fájlnevben pontosvessző található, mindent az után úgy vesz, mintha új parancs

kezdődött volna. Ez azt is eredményezi, hogy a fájl nem lesz megtalálható, mivel félreértelmezte.

Szerencsére, a módszer ennek a problémának a kezelésére azonos a kettős keresztnél már kitárgyalttal. Azaz, akár ./-t tehetsz a fájlnev elé (nano ./;myfile.txt), akár sima idézőjelbe is teheted (;myfile.txt).

Sima idézőjel használata. Bizonyos „speciális” karaktereket alkalmazva a fájlnevben, csak annyit kell tenned, hogy sima idézőjelbe teszed a fájlnevet. Ezek a \$, !, @, &, (), ", <>, és \ karakterek.

Mellékesen megjegyzendő, ha olyan fájlnevvel kell dolgoznod, amiben szimpla idézőjel van, akkor a fájlnevet dupla idézőjel közé rakd.

Különleges kezelést nem igényel. A többi „speciális” karakter a fájlnevben megjelenhet és nincs szükség különleges eljárásra Linux alatt. Ezek: +, %, *, ^, {}, [], _, =, ?, ., :, ~, és .. Az ilyen karaktereket tartalmazó fájlokat úgy kell kezelni, mint bármilyen más fájl.

Összegzés

Akár követed a fájlnevekre vonatkozó szabályokat, akár nem, bármikor összetalálkozhatasz fájlokkal, amikkel nehéz boldogulni. Valójában feltehetően már találkoztál ilyennel. A szabályok ismerete lehetővé teszi, hogy megfelelően kezeld azokat. Ugyanakkor, a saját fájljaid esetében alkalmazva azokat, elkerülheted a nehézségeket és egyszerűbbé teheted azok dolgát, akikkel esetleg megosztod a fájljaidat.