

NÉV

grep, egrep, fgrep – mintához illeszkedő sorokat nyomtat

ÁTTEKINTÉS

grep [**-[[AB]**]*num*] [**-[CEFGVBchilnsvw]**] [**-e**] *mintá* | **-f***file*] [*fájlok...*]

LEÍRÁS

Grep a megnevezett bemeneti *fájlok* -ban a megadott *mintá* -hoz illeszkedő sorokat keres. Amennyiben nincs bemenő fájlnev megadva, vagy az a **-** jel, a szabványos bemenetet olvassa. Alapértelmezés szerint **grep** a mintához illeszkedő sorokat kinyomtatja.

Három fő változata van a **grep** -nek, melyeket a következő opciókkal szabályozhatunk:

- G** A *mintá* -t alap szabályos kifejezésként (basic regular expression) kezeli. Részletesebben lásd lentebb. Ez az alapértelmezett.
- E** A *mintá* -t bővített szabályos kifejezésként (extended regular expression) kezeli. Részletesebben lásd lentebb.
- F** A *mintá* -t fix stringekből álló listának veszi, melyek újsor-jelekkel vannak elválasztva, és amelyekből bármelyikhez való illeszkedést keressük.

Létezik továbbá két programváltozat: **egrep** és **fgrep** . **Egrep** hasonló, de nem azonos **grep** —**E**, -vel és kompatibilis a történelmi Unix **egrep** -pel. **Fgrep** azonos **grep** —**F** -fel.

A **grep** minden változata megérti az alábbi opciókat:

- num* Az illeszkedést mutató sorok környezetükkel együtt lesznek kinyomtatva, azaz ezen sorok előtt és utáni *num* számú sor is nyomtatásra kerül. Eközben a **grep** program sosem fog egy sort többször kinyomtatni, még akkor sem, ha ezek a környezetek átfedik egymást.
- A** *num*
Az illeszkedést mutató sorok utáni *num* sor is nyomtatásra kerül.
- B** *num*
Az illeszkedést mutató sorok előtti *num* sor is nyomtatásra kerül.
- C** Azonos **-2** -vel.
- V** Kinyomtatja a **grep** verziószámát a szabványos hibakimenetre. Ezt a verziószámot célszerű csatolni minden hibajelentéshez. (Lásd lent.)
- b** A kimeneti sorok előtt kinyomtatja a bemeneti fájlban belüli offset értéket. (Azaz azt, hogy a sor hányadik bájjal kezdődik a fájl kezdete után.)
- c** Elhagyja a szokásos kimenetet, ehelyett az illeszkedést mutató sorok számát írja ki minden fájl esetére. A **-v** opcióval (lásd lent), a nem illeszkedő sorokat számolja.
- e** *mintá*
A *mintá* -t használja mintának. Hasznos, ha a *mintá* — jellel kezdődik.
- f** *fájl* A *mintá* -t a *fájl* -ból veszi.
- h** Több fájlban való kereséskor elhagyja a kimeneten a fájlnevek sor eleji kiírását.
- i** Nem különbözteti meg a kis- és nagybetűket sem a *mintá* -ban, sem a bemeneti fájlban.
- L** Elhagyja a szokásos kimenetet, és csak azon *fájlok* neveit adja meg, amelyekből *nem* származna kimenet.
- l** Elhagyja a szokásos kimenetet, és csak azon *fájlok* neveit adja meg, amelyekből származna kimenet.
- n** A kimenet sorai elején megadja a sor bemeneti fájlban belüli számát.
- q** Csendes (quiet) üzemmód: elhagyja a normális kimenetet. (Ilyenkor a **grep** -nek csak a visszatérési értéke ad információt.)
- s** Elhagyja a nemlétező vagy olvashatatlan fájlokról szóló hibajelentéseket.
- v** Megfordítja az illeszkedés értelmét: a mintához nem illeszkedő sorokat választja ki.
- w** Csak azokat a sorokat választja ki, amelyekben az illeszkedés teljes szavakból származik. Azaz az illeszkedést mutató szövegrész előtt és után nem állhat betű, szám vagy aláhúzás-jel.
- x** Csak teljes sorokkal való illeszkedést vizsgál.

SZABÁLYOS KIFEJEZÉSEK (REGULAR EXPRESSIONS)

A szabályos kifejezés egy *mintá*, amely a stringeknek egy halmazát írja le. A szabályos kifejezések az aritmetikai kifejezésekhez hasonlóan konstruálhatók, azaz különböző operátorokkal kisebb

kifejezésekből építhetők fel.

Grep kétféle változatát érti a szabályos kifejezések nyelvtanának (syntax): az “alap” (basic) és a “bővített” (extended) változatot. A GNU **grep** -ben nincs különbség a kétféle nyelvtan által elérhető funkciókban. Más megvalósításokban az alap szabályos kifejezések kevésbé hatékonyak. A következő leírás a bővített szabályos kifejezésekre vonatkozik; az alap reguláris kifejezéseknek ettől való különbségeit ezután összegezzük.

Az elemi építőkövek olyan szabályos kifejezések, melyek egyetlen karakterhez illeszkednek. A legtöbb karakter (pl. a betűk és a számok) szabályos kifejezések, melyek önmagukhoz illeszkednek. A speciális jelentéssel rendelkező metakarakterek (pl. a ., *, [karakterek) elé backslash-t kell írni, hogy speciális jelentésük helyett a karakterhez való illeszkedést vizsgálhassuk. (Pl. a mintában levő \[fog illeszkedik a szövegbeli nyitó szögletes zárójellel.)

A [és] jelek közé írt karakterlista illeszkedik a listában szereplő bármely karakterhez. Amennyiben a lista a ^ jellel kezdődik, az illeszkedés a listában *nem* szereplő karakterekkel áll fenn. Például a **[0123456789]** szabályos kifejezés bármely számjegyhez illeszkedik.

ASCII karakterek tartománya az első és utolsó karakterek megadásával adható meg, ha ezeket '-' jel választja el. (Pl. [a-f] ugyanaz, mint [abcdef].)

Végül a karakterek néhány speciális osztálya előre definiált névvel rendelkezik. (Ezen nevek jelentése angolul magától értetődő, itt kifejtjük őket.)

[:alnum:]

betű vagy szám

[:alpha:]

betű

[:cntrl:]

vezérlőkarakter

[:digit:]

számjegy

[:graph:]

grafikus karakter

[:lower:]

kisbetű

[:print:]

nyomtatható karakter

[:punct:]

elválasztó, központosító jel (.,;?!)

[:space:]

szóköz

[:upper:]

nagybetű

[:xdigit:]

hexadecimális szám

Például **[:alnum:]** jelentése: **[0-9A-Za-z]**, azzal a kitételrel, hogy az utóbbi forma az ASCII kódolástól függ, míg az előbbi nem, így átvihető. (Megjegyzés: ezen osztálynevekben szereplő zárójelek a szimbolikus nevek részei, és a zárójeles listákat határoló zárójelek mellett ezeket is meg kell adni.)

A legtöbb metakarakter elveszti speciális jelentését egy listán belül. Egy] jelet a minta első elemeként szerepeltetve csatolhatunk a listához. Hasonlóan a ^ jel az első kivételével bármely helyre kerülve a '^' jelet fogja jelenteni. Végül a '-' jelet a lista utolsó elemeként kell írni, ha nem metakarakterként akarjuk értelmezni.

A pont (.) minden karakterhez illeszkedik. A \w szimbólum az **[:alnum:]** szinonímája, míg \W ugyanaz, mint **[^[:alnum:]]**.

A ^ és a \$ jelek metakarakterek, melyek rendre a sorkezdő és -záró üres srtingekhez illeszkednek.

Hasonlóan a `<` és `>` szimbólumok a szavak elejénél illetve végénél található üres stringhez illeszkednek. A `\b` a szóhatáron levő üres stringhez illeszkedik, míg `\B` a *nem* szóhatáron levő üres stringhez illeszkedik.

Az egy karakterhez illeszkedő szabályos kifejezéseket a következő ismétlési operátorok egyike követheti:

- ? Az előző tag opcionális, és legfeljebb egyszer illeszkedik.
- * Az előző tag nulla vagy több alkalommal illeszkedik.
- + Az előző tag egy vagy több alkalommal illeszkedik.
- {*n*} Az előző tag pontosan *n* alkalommal illeszkedik.
- {*n*,} Az előző tag *n* vagy több alkalommal illeszkedik.
- {*m*} Az előző tag legfeljebb *m* alkalommal illeszkedik.
- {*n*,*m*} Az előző tag legalább *n* de legfeljebb *m* alkalommal illeszkedik.

Két szabályos kifejezés összefűzhető; a kapott szabályos kifejezés minden olyan stringhez illeszkedik, amely a két szabályos kifejezéshez illeszkedő két részstringből állítható össze a sorrend megtartása mellett.

Két szabályos kifejezés összekapcsolható a `|` infix operátorral; a kapott szabályos kifejezés minden stringhez illeszkedik, amelyikhez valamelyik rész kifejezés illeszkedik.

Az ismétlés nagyobb precedenciájú, mint az összefűzés, ami viszont a választóoperátornál (azaz `|`-nál) nagyobb precedenciájú. Egy részkifejezés zárójelbe tehető, hogy felülbíráljuk a precedenciát.

A `\n` visszahivatkozást (backreference) jelöl, ahol *n* egy számjegy. Ez illeszkedik ahhoz a stringhez, amely a szabályos kifejezés ezt megelőző *n*-edik zárójeles alkifejezéshez illeszkedett.

Az alap szabályos kifejezésekben a `?`, `+`, `{`, `|`, `(`, és `)` metakarakterek elvesztik speciális jelentésüket; helyettük a backslash-es változatukat kell használni: `\?`, `\+`, `\{`, `\|`, `\(`, és `\)`.

Az **egrep** programban a `{` metakarakter elveszti speciális jelentését, helyette a `\{` használható.

DIAGNOSZTIKA

Általában a kilépési állapot (exit status) 0, ha volt illeszkedés és 1, ha nem volt egyetlen illeszkedés sem. A `-v` opció megfordítja ennek értelmét.

A 2 kilépési állapot hibát jelez: nyelvtani hiba volt a mintában vagy elérhetetlen a bemeneti fájl, esetleg más rendszerhiba történt.

HIBÁK

A hibajelentések e-mail-en a **bug-gnu-utils@prep.ai.mit.edu** címre küldendőek. Győződjön meg róla, hogy a "grep" szó szerepel valahol a "Subject:" mezőben.

Nagy ismétlési számok a `{m,n}` ismétlési operátorban nagy memóriefelhasználást eredményezhetnek. Továbbá néhány más bonyolult szabályos kifejezés exponenciálisan növekvő időt és tárhelyet követelhet, és azt eredményezheti, hogy a **grep** kifut a memóriából.

A visszahivatkozások nagyon lassúak és exponenciálisan növekvő futásidőt igényelnek.

MAGYAR FORDÍTÁS

Horváth András <horvatha@rs1.szif.hu>