

Legyen saját IRC-botod!

Egy kis Perl-tudás, kevés hálózati ismeret és egy csipetnyi RFC-olvasgatás a biztos receptje egy egyéni IRC-bot készítésének.

Az IRC (Internet Relay Chat) egy olyan TCP/IP-alapú protokoll, amely hálózaton keresztül valós idejű párbeszédet folytatását teszi lehetővé. Ha két ügyfél csevegni szeretne egymással, mindketten csatlakoznak egy-egy kiszolgálóhoz (nem feltétlenül ugyanahhoz). Legalább az egyiküknek tudnia kell a másik azonosítóját (becenévét, azaz nickjét), hogy a beszélgetés megkezdődhessen. Az első ügyfél üzenete elküldéséhez azt a kiszolgálót használja, amelyikhez kapcsolódott. Amikor a kiszolgáló megkapja a kézbesítendő üzenetet, felállít egy útvonalat, amelyen kiszolgálóról kiszolgálóra terjed az üzenet, amíg csak el nem éri azt, amelyikhez a másik ügyfél kapcsolódott. Így ő megkapja az üzenetet, és ugyanezzel a módszerrel válaszolhat rá. Habár az IRC egyszerű szöveges alapú protokoll, a biztonság érdekében több dologra is ügyel. Egyrészt az ügyfél csak akkor kapja meg az üzenetet, ha a címzettek között van (ugyanazt az üzenetet több becenévre is el lehet küldeni), másrészt egy kiszolgáló egy adott ügyfél megtalálásához csak egyetlenegy másik kiszolgálónak adhatja tovább az üzenetet. Ez jó, hiszen valamiféle biztonságot nyújt, ha egy IRC-hálózaton a kiszolgálók közül remélhetőleg nem az összes kapja meg az üzenetedet. Ez a felépítés ugyanakkor gátja is a rendszernek. A csatornákon, ahol nemegyszer 50–60 ügyfél is eszmét cserél, gyakran tapasztalható, hogy nem minden ügyfélhez jut el időben az üzenet, és a többiek úgy látják, hogy már megtárgyalt kérdésekről beszél (lag). Az is előfordul, hogy ha a két (földrajzi értelemben vett) hely közti összeköttetést biztosító kiszolgáló leáll, az IRC-hálózat két félre szakad. Legyenek az ügyfelek ugyanazon a csatornán, mindenki csak azt látja, aki vele azonos oldalon van (split). Ilyenkor időbe telik, amíg új útvonalat találva ismét elérhetik egymást.

Mi az a bot?

Egy IRC-bot ahhoz hasonló ügyfél, amelyet csevegésre is használasz. A különbség, hogy a botot valamilyen sokszor ismétlődő feladatra tartják, és nem szükséges előtte embernek ülnie. A `#linux.hu` csatornán létezik egy Rozsdas nevű bot, ami a két kérdőjellel kezdődő sorokat keresi, és ha ilyet lát, a kérdőjeleket követő kifejezésre valamilyen leírást ad. Akad olyan csatorna, ahol egy botnak nincs más dolga, mint feltennie egy kérdést, majd a válaszra várni. Ha valaki a megadott időn belül helyesen válaszol, egy pontot kap. Ezután – vagy az idő lejártával – a bot felteszi a következő kérdést. Egy játék megadott számú kérdésből áll, és a végén az győz, akinek a legtöbb pontja van. Ezekhez hasonló és ezer másik feladat képzelhető el egy bothoz. Természetesen én egy egyszerűbbet írtam, ami adott időközönként felolvas egy az állományból véletlenszerűen kiválasztott sort. Gyakorlásnak megfelelő, viszont hasznavehetetlen. Ezért arra kérek mindenkit, ezt a botot ne küldje fel egyik kiszolgálóra sem! Egyrészt egyes kiszolgálók kifejezetten megtiltják a botok jelenlétét, másrészt nincs olyan csatorna, ahol szívesen látnának egy ilyen botot. A legjobb, ha feltelepítesz egy saját IRC-kiszolgálót, és azon próbálgatsz.

Így gyorsabb a fejlesztés, és biztosan senkinek nem okozol kellemetlenséget. Egy kiszolgáló telepítése és beállítása túlmutat e cikk keretein, ezért kérek, vedd a fáradságot, és válassz egyet a terjesztésedhez megfelelők közül. A Debian használók az csomag telepítésével az ircd két változatához is hozzájuthatnak. Ők ircd néven az Undernet hivatalos kiszolgálóját érhetik el, míg a dancer egy kisebb hálózatokra tervezett változat.

A protokollról nagyvonalakban

Az IRC-hálózaton mind a kiszolgálók, mind az ügyfelek üzenetekkel tartják a kapcsolatot. Egy üzenet a következőképpen fest:

```
[ ':' <elitag> <SZ K Z> ] <parancs>
↳ <param0terek> <crlf>
```

Az előtag (prefix) használata nem kötelező. Az előtag jelenlétét a kettőspont (:) mutatja. A kettőspont és az előtag között nem lehet szóköz. Az előtag az üzenet feladójának azonosítását szolgálja, ha elmarad, az IP-csomag küldője lesz a feladó. Elsősorban kiszolgálók használják; ha ügyfél használja, és nem önmagát tünteti fel feladóként, az első kiszolgáló, amelyikhez eljut az üzenet, visszajelzés nélkül megsemmisíti azt. Én sem használtam sehol az előtagot.

A parancs szabványos IRC-parancs. Habár a parancsnevekben az IRC-alkalmazások elvileg nem különböztetik meg a kis- és nagybetűket, érdemes csupa nagybetűvel írunk.

A kapcsolók a parancstól függenek. Nem tartalmazhatnak szóközt, legfeljebb az utolsó, ebben az esetben kettőspontot kell tennünk elé.

A sort sortörés-kocsivissza zárja, de ha jobban tetszik, a legtöbb nyelvben ez is megteszi: `\n\r`.

Mielőtt ügyfél csatlakozhatna egy csatornához, először be kell jegyeznie magát a kiszolgálónál. Ez két paranccsal történik: USER és NICK.

```
USER <felhasználó név> <m d> <használatlan>
↳ <val di_név>
```

Az első kapcsoló a felhasználói név. A linuxos IRC-ügyfelek legtöbbször annak a felhasználónak a nevét küldik el, akinek a nevében a program fut. Te mint programozó akármit megadhatsz. Arra figyelj, hogy bizonyos felhasználóneveket egyes csatornákon szűrnek. A `#linux.hu`-ra például root felhasználónévvel nem lehet felmenni (illetve fel lehet, de egy bot kidob).

A mód egy szám, ami az ügyfél különleges tulajdonságait írja le. Itt lehet többek között azt beállítani, hogy láthatatlan legyél. Ez azt jelenti, hogy a csatorna tagjainak listájában nem jelensz meg.

A harmadik kapcsoló jelenleg használaton kívül van, ezért csillag (*) karaktert szokás helyette írni.

A valódi név egy tetszőleges karakterlánc. Ide többnyire

a felhasználói névhez tartozó teljes nevet szokták ide írni, ám ez változó. Mivel szóközt is tartalmazhat, kettősponttal kell kezdődnie.

NICK <becenőv>

Ezzel a kapcsolóval határozhatod meg a becenevedet. Bejegyzés után bármelyik kiszolgáló küldhet a botnak egy ping-et. Az IRC-kiszolgálók ugyanis nem ICMP-üzenetekkel ellenőrzik, hogy egy ügyfél fent van-e még a hálózaton, hanem belső ping-gel. Ha nem készíted fel a botot, hogy ezekre válaszoljon, a kiszolgáló bontani fogja a kapcsolatot. A ping parancsot kapcsolóként egy azonosító követi. Az erre válaszoló ügyfélnek egy pong parancsot kell küldenie, kapcsolóként ugyanazzal az azonosítóval.

A PRIVMSG parancsot használhatod, ha hozzá akarsz szólni a beszélgetéshez, és te is ilyeneket fogsz kapni, ha valaki veled beszélget.

PRIVMSG <c mzett> <sz veg>

Ha a címzett egy becenév, akkor magánbeszélgetés indul közted és a címzett között. Ha kettős kereszttel (#) kezdődik, egy csatornát jelöl, és a csatorna összes tagja megkapja az üzenetet. Ha dollárjellel (\$) indul, a dollárjel után álló kiszolgálóhoz csatlakozó összes ügyfél olvasni fogja a szöveget. Helyettesítő karakterek mindhárom esetben használhatók (*, ?). Egy megszorítás létezik, miszerint a kettős kereszttel, illetve dollárjellel kezdődő nevekben legalább egy pontnak (.) szerepelnie kell, és az utolsó pontot nem követheti helyettesítő karakter. Ezt azért találták ki, hogy például a \$*-ra ne lehessen üzenetet küldeni, ezt ugyanis a világ összes IRC-hálózatot használója megkapná.

A szöveg természetesen tartalmazhat szóközt, így kettősponttal kell kezdődnie.

Ha további részletekre vagy kíváncsi a protokollal kapcsolatban, vagy más parancsok is érdekelnek, ajánlom figyelmedbe a kapcsolódó RFC-eket: RFC1459 és RFC2812 (lásd a 39. CD Magazin/Bot könyvtárban).

Egy perlbeli megvalósítás

Ezek után lássunk egy példát! Az én kis IRC-botom nem csinál mást, csak csatlakozik a megadott csatornára, és bizonyos időközönként felolvas egy véletlenszerűen kiválasztott sort egy állományból. Az Interneten ezeket a botokat nagyon idegesítőnek tartják, és végleg meg tudod vele utáltatni magadat. Mindenki elismeri a programozói teljesítményt, azonban nem győzöm elégszer hangsúlyozni: ne használd ezt a botot nyilvános IRC-kiszolgálón! Ha nagyon tetszik, írd egy olyat, amelyik hasznos. És feltétlenül kérdezd meg a kiszolgáló rendszergazdáját, hogy felteheted-e! Tudomásom szerint az *irc.extra.hu* egyáltalán nem engedélyezi a botok használatát a kiszolgálón. Az *elte.irc.hu* napi üzenetéből (motd, message of the day) az derül ki, hogy két hétig elviselik a botokat, utána mindenkit nyomtatékosan megkérnek, hogy távolítsa el kis kedvencét.

A *sote.irc.hu*-ra lehet botot vinni, de erről egy e-mailben (*ircadm@sote.irc.hu*) értesíteni kell a kiszolgáló üzemeltetőjét. (A listát lásd a 39. CD Magazin/Bot könyvtárban).

Mivel feltételezem, hogy programoztál már Perlben, csak nagy vonalakban magyarázom el a program működését. A \$\
különleges változót a Perl minden kimenő adat végéhez hozzáfűzi. Így a print utasítások végére nem kell mindig a kötelező \
r-t odaírni. A \$| szintén a kimenő adatokra

vonatkozik: nem használ átmeneti tárat az íráshoz, minden adatot azonnal kiír.

A \$nick a becenevet, a \$channel a csatorna nevét, a \$wait pedig egy számot tartalmaz, amely másodpercekben fejezi ki azt az időt, amennyinek két bementett idézet között legalább el kell telnie.

A helyi géphez a 6667-es kapun, TCP/IP-protokollon keresztül kapcsolódik. Ha nem sikerült, a program leáll.

Láthatóan semmiféle különleges módot nem adok a botnak, és a teljes neve Zebulon Master.

A @stuff tömböt több dologra is használni fogom (lásd később). Amennyiben a \$joined 0, még nem kapcsolódott, ha 1, az előbbi iterációban kapcsolódott, amennyiben 2, már rég fent van a csatornán. A \$last tárolja az utolsó kiíratás dátumát az 1970. január 1-je óta eltelt másodpercek számában. A \$i általános ciklusváltozó.

A while ciklus minden lefutásakor beolvass egy sort a \$irc foglalatról.

Az első, hogy ellenőrzi, megpingelték-e. Ha igen, a ping kapcsolóként megadott azonosítóval egy pong-ot küld. Ha nem ping volt, és épp most csatlakozott a csatornához, egy üdvözlő szöveget küld. Ha valaki megszólította, és azt mondta, „aludj”, kilép.

Ha letelt a megadott idő, a megadott állományból kiválaszt egy sort és elküldi.

Amennyiben még nem csatlakozott, megteszi.

A ciklusból történő kilépés után a Quit paranccsal búcsút mond a kiszolgálónak, és ha nem szakította volna meg a kapcsolatot, akkor a bot megteszi.

Bothibák

Ez egy nagyon kezdetleges bot, és több gond is akad vele.

Nem kezeli az olyan váratlan eseményeket, ha a becenév foglalt, vagy nem engedik be a csatornára (miért is ne engednék). Továbbá a saját kitzűzött célját sem látja el tisztességesen. Nem 12 másodpercenként olvas fel egy sort az állományból. Ha senki sem szólal meg a csatornán, és nem kap üzenetet, nem fog beszélni. Ez azért van, mert a Perl <> műveletjelét használtam olvasásra. Ez a megadott leíróról egy sort olvas be. Mivel a leíró nyitva van (a foglalat nem zárult le), nincs állományvége (EOF), és addig nem tér vissza, amíg üzenetet nem kap. Valójában az olvasást a sysread() függvényvel kellene megoldani, ahogy más Perlben írt botok is teszik, de nem akartam túlbonyolítani a példát. Léteznek bővítmények az IRC-protokollhoz, mint például a CTCP (Client To Client Protocol). Ezzel a kiszolgálót megkerülve lehet adatot kérni egy ügyféltől. Egy ügyfél a másiktól a használt ügyfél típusát, a rendszeridőt és más hasonló adatot kérdezhet le. Ha a mi botunk kap egy ilyen üzenetet, nem válaszol rá. Én azért szeretem, mert az enyém.

Végezetül

Nagyon izgalmas IRC-botot írni. Remélem, te is örömet lelsz benne, és a tudásodat okosan használod. Sok szerencsét az IRC határtalan világához!



Fülöp Balázs

(xut@freemail.hu) 17 éves, imádja a Túró Rudit, a Debian Linuxot és a teheneket. Az ELTE Radnóti Miklós Gyakorlóiskola tanulója immár ötödik éve. Kedvenc írója Slawomir Mrodek. Leginkább a számítógépes hálózatok biztonsága érdekli.