**NAME**

talk − allow user to listen and talk to one or more other users simultaneously

**SYNOPSIS**

**talk** [ **−n** alias ] [ user user ... ]

**DESCRIPTION**

*Talk* allows a group of users to simultaneously talk and listen to each other. If you run *talk* without specifying any users with whom you would like to speak, you immediately join the first conversation currently in progress which is not restricted in such a way that you cannot join it. All other participants in the conversation are alerted to the fact that you have arrived by "*The Chair*". "*The Chair*" is the name of the central distributor process, which oversees the conversation and distributes lines of text typed in by each participant to all the other users. If there is no conversation in progress when you type *talk*, a new central distributor process is created with you as the first member of the conversation.

If you run *talk* and specify a list of users with whom you would like to speak, *talk* attempts to put you into a single conversation with these users. If some or all of the users specified are already running *talk* and are in more than one conversation, *talk* will ask you to pick the conversation you wish to join. If those users who are running talk and with whom you wish to speak are all in the same conversation, *talk* attempts to join you to that conversation. All users you listed who are not running *talk* at the time are alerted to the fact that you wish to speak to them if you successfully join the conversation.

When you first join a conversation, "*The Chair*" tells you the names and aliases of all the other people in the conversation, and which tty line they are logged in on. Any time a user leaves the conversation all the remaining members are apprised on the fact. If the owner of the conversation leaves, ownership is passed to the first person that *the Chair* finds who is still in the conversation and they are alerted to the fact that they now own the conversation. If there is no one left in the conversation, it becomes available again. When the last user leaves *talk*, the central distributor process terminates.

*Talk* is structured in such a way that people are not interrupted if more than one person is talking at the same time. If there is a terminal type specified (see *stty*(1) and *ioctl*(2)), *talk* divides the terminal screen into two sections, reserving the top two lines for your typing. Every partially typed line appears at the top. When it is complete it is sent to all the participants of the convervation.

If there is no terminal type, then *talk* saves up any output from other users while you are typing in a line. When you either finish your line or erase it, *talk* then sends you all the queued output. In this way you are not interrupted while typing.

Normally you are identified by your login name. This name can be changed to an *alias* at the time *talk* is invoked with the **−n** flag. It can also be changed during the conversation. See ~**name** below. This is useful if more than one person with the same login id is involved in a conversation.

*Talk* can be asked to *alert* other users that you would like to talk to them. This is the only time that *talk* sends anything to a user's terminal uninvited. If you specify a list of other user ids at the time *talk* is invoked, it will send out the following message to each of those users who it finds logged in:

"*your_login_id*" wants to "talk" to you.

*Talk* will report any user listed, who it doesn't find logged in. Also any user whose terminal is set so that *talk* can't send a message will be reported.

*Talk* has a number commands that can be typed once you have started *talk*. Any line beginning with a ~ (tilda) is assumed to be a command to *talk*. The commands follow. Abbreviations

appear in ()'s.

**~!***cmd*

Escape to the shell and perform *cmd*. All messages from other users will continue to arrive, but nothing typed by you will be transmitted to the other participants while you are escaped.

**~help (~h)** [ cmd cmd ... ]

Type a help message. If one or more of the *talk* commands is specified, then type a help message for the specific command. There are help messages for **!**, **help(h)**, **alert(a)**, **name(n)**, and **users(u)**.

**~alert (~a)** user [ user ... ]

Send message to the specified users, if they are logged in, saying that you would like to *talk* to them. This results in the same action as specifying users when originally starting *talk*.

**~name (~n)** [ alias ]

If no argument is specified, your current name and alias, if any, is printed out. If a new alias is specified, then the alias is changed to the new one and all other participants in the conversation are alerted to the fact that you have changed your alias.

**~users (~u)**

List all the current participants in the conversation, their aliases, and what terminal they are logged in on. You are marked with an ->. The owner of a conversation is denoted by an *. If the conversation is restricted and if you are the owner, the various restrictions to the conversation are listed. If the conversation is restricted, but you are not the owner, only the word *Restricted* appears after the conversation number.

**~join (~j)** cnum

**~knock (~k)** cnum

Attach to the conversation numbered *cnum*. The conversation must already be in existence. If you can join immediately, you are detached from your current conversation and attached to the new conversation, with appropriate warnings to the members of the old and new conversations. If you must get permission to join the conversation, *talk* puts you into a wait loop and requests permission from the owner of the conversation. If you get tired, a <*del*> will exit you from the waiting state.

**~create (~c)** [−**l**] [ [+**u**|−**u** |+**g**|−**g**] name [...] ] ...

Create a new conversation and make you the owner of it. If there is room for another conversation this command will succeed. At the time of creation it is possible to specify restrictions on who may join the new conversation. See ~*lock* for the details on the restriction switches.

**~lock (~l)** [−**l**] [ [+**u**|−**u** |+**g**|−**g**] name [...] ] ...

The owner of a conversation may restrict entry to the conversation. If no arguments are given to the lock command or when the conversation is created with the **~create** command only the −**l** switch is specified, the conversation will be totally locked and each new participant will be required to get specific permission from the owner of the conversation to join.

Specific restrictions can be made with the **u** and **g** switches. +**u** followed by a list of user ids means that these users are allowed to enter the conversation without having to ask permission. −**u** means that these users must ask permission. It is not permissible to mix +**u** and −**u** or +**g** and −**g** switches. If you specify that some people can enter

without knocking, then all other users will have to ask specific permission. If you specify that some users are to ask permission, then all other users will be allowed to enter without asking. The +g and −g switches work the same way for groups.

**~unlock (~unl)**

If you are the owner of the conversation, the conversation is unlocked and all restrictions against joining the conversation are removed.

**~admit (~ad)** name [name ...]

Admit any person having one of the listed names to this conversation if they are waiting for permission. Only the owner of a conversation can admit people. If no one is waiting with the names listed, a message from *the Chair* will be sent to you.

**~deny (~d)** name [name ...] [−r reason ]

Deny any person having one of the listed names permission to enter your conversation. Only the owner of a conversation can deny people entry. If no one is waiting with the names listed, a message from *the Chair* will be sent to you. A reason for denying permission can be included with the −r switch. Everything after the −r switch to the end of the line is taken as a reason and is appended to the end of the "Permission denied." message that is sent back to a user who is being denied entry.

If at any time the divider of "*******"s one the second line of the screen gets scribbled, as might happen during a shell breakout if something is run that alters the variable scrolling, typing a control L ( ^L) will cause the divider on line two to be redrawn.

**SEE ALSO**

write(1)

**BUGS**

A maximum of 16 people can use *talk* at one time.

Echo can be slow since *talk* runs in **RAW NOECHO** mode. When running on a terminal without a terminal type set, the first character of a line causes the typing of a newer header, if required, as well as echoing of the character. It can also be slower than any other character of the line since before it is echoed, the local process informs the central distributing process not to send any more messages and waits for an acknowledgement. When a line is complete, the local process tells the central process that it is okay to send lines again.