###Linux = kernel ### gnu.org linuxdistro.org kernel.org apache.org 1. Example Version 2.6x kernel (2.6.18) > 'uname -a' returns OS/Kernel information Note: 'uname -a' returns the following useful info: 1a. OS - Linux 1b. Fully Qualified Domain Name (FQDN) 1c. Kernel version - 2.6.18... 1c 2.6 = major version 1c .18 = minor version 1c anything else after the minor version indicates that the kernel was patched by the distributor 1d Date and time that the kernel was compiled 2. Supports multiple versions: Basic - Red Hat Enterprise Linux Server and Debian or ubuntu supports 2 physical (Socket) CPUs Up to 4 virtual guests Advanced Platform supports unlimited physical CPUs supports unlimited virtual guests Note: Virtualization limits pertaining to the virtualization technology included with Red Hat Enterprise Linux. NOT third-party software (VMWare) 3. Linux Supports the following platforms: a. Intel 32/64-bits b. AMD 32/64-bits c. IBM - POWER and z-series, S/390 Note: Memory limitation is based on hardware Note: Common uses of the various versions of RHEL or Debian or Suse d. RHEL Basic Version or the minimal server e. File & Print f. Web server g. Infrastructure server (DHCP, DNS, Proxy, etc.) h. RHEL Advanced Version i. Application server (Apache Tomcat, JBOSS, Weblogic, WebSphere, etc.) j. Database server (MySQL, PostgreSQL, Oracle, Ingres, etc.) k. Clustering ###BASIC LINUX COMMANDS### 1. tty - reveals the current terminal 2. whoami - reveals the currently logged-in user 3. which - reveals where in the search path a program is located 4. echo - prints to the screen 4a. echo $PATH - dumps the current path to STDOUT 4b. echo $PWD - dumps ths contents of the $PWD variable 4c. echo $OLDPWD - dumps the most recently visited directory 5. set - prints and optionally sets shell variables 6. clear - clears the screen or terminal 7. reset - resets the screen buffer 8. history - reveals your command history 8a. !690 - executes the 690th command in our history 8b. history command is maintained on a per-user basis via:~/.bash_history 8c ~ = users's $HOME directory in the BASH shell 9. pwd - prints the working directory 10. cd - changes directory to desired directory 0a. 'cd ' with no options changes to the $HOME directory 0b. 'cd ~' changes to the $HOME directory 0c. 'cd /' changes to the root of the file system 0d. 'cd Desktop/' changes us to the relative directory 'Desktop' 0e. 'cd ..' changes us one-level up in the directory tree 0f. 'cd ../..' changes us two-levels up in the directory tree 11. Arrow keys (up and down) navigates through your command history 12. BASH supports tab completion:type unique characters in the command and press 'Tab' key 13. You can copy and paste in GNOME terminal windows using: a. left button to copy or ctrl-C to b. right button to paste OR Ctrl-Shift-v to paste 14. ls - lists files and directories a. ls / - lists the contents of the '/' mount point b. ls -l - lists the contents of a directory in long format:Includes: permissions, links, ownership, size, date, name c. ls -ld /etc - lists properties of the directory '/etc', NOT the contents of '/etc' d. ls -ltr - sorts chronologically from older to newer (bottom) e. ls --help - returns possible usage information f. ls -a - reveals hidden files. e.g. '.bash_history' g. ls ?a* h. ls ??? i. ls a* starting with a OR anywhere a *a OR in betw*a* Note: files/directories prefixed with '.' are hidden. e.g. '.bash_history' 15. cat - catenates files a. cat 123.txt - dumps the contents of '123.txt' to STDOUT b. cat 123.txt 456.txt dumps both files to STDOUT c. cat 123.txt 456.txt > 123456.txt - creates new catenated file d. cat 123.txt 456.txt >> appends a file 16. mkdir - creates a new directory a. mkdir test5 - creates a single directory with the name specified c. mkdir -p test1/test2/test3/ will create nested directories 17. cp - copies files a. cp 123.txt test/ By default, 'cp' does NOT preserve the original modification time b. cp -v 456.txt test 18. mv - moves files and renames files a. mv 123456.txt test - moves the file, preserving time 19. rm - removes files/directories a. rm 123.txt b. rm -rf 456.txt - removes recursively and enforces 20. touch - creates blank file/updates timestamp a. touch test.txt - will create a zero-byte file, if it doesn't exist b. touch 123456.txt - if exists update the timestamp c. touch -t 200801091530 123456.txt - changes timestamp 21. stat - reveals statistics of files a. stat 123456.txt - reveals full attributes of the file 22. find - finds files using search patterns a. find / -name 'fstab' Note: 'find' can search for fields returned by the 'stat' command 23. alias - returns/sets aliases for commands a. alias - dumps current aliases b. alias copy='cp -v' ###Linux Redirection & Pipes### Ability to control input and output Input redirection '<': 1. cat < 123.txt Note: Use input redirection when program does NOT default to file as input Output redirection '>': 1. cat 123.txt > onetwothree.txt Note: Default nature is to: 1. Clobber the target file 2. Populate with information from input stream Append redirection '>>': 1. cat 123.txt >> numbers.txt - creates 'numbers.txt' if it doesn't exist, or appends if it does 2. cat 456.txt >> numbers.txt Pipes '|': Connects the output stream of one command to the input stream of a subsequent command 1. cat 123.txt | sort 2. cat 456.txt 123.txt | sort 3. cat 456.txt 123.txt | sort | grep 3 ###Command Chaining### 1. Permits the execution of multiple commands in sequence 2. Also permits execution based on the success or failure of a previous command 1. cat 123.txt ; ls -l - this runs first command, then second command without regards for exit status of the first command 2. cat 123.txt && ls -l - this runs second command, if first command is successful 3. cat 1234.txt && ls -l 4. cat 123.txt || ls -l - this runs second command, if first command fails 24. more|less - paginators, which display text one-page @ a time 1. more /etc/fstab 2. less 1thousand.txt 25. seq - echoes a sequence of numbers a. seq 1000 > 1thousand.txt - creates a file with numbers 1-1000 26. su - switches users a. su - with no options attempts to log in as 'root' 27. head - displays opening lines of text files a. head /var/log/messages 28. tail - displays the closing lines of text files a. tail /var/log/messages 29. wc - counts words and optionally lines of text files a. wc -l /var/log/messages7 b. wc -l 123.txt 30. file - determines file type a. file /var/log/messages