***Identify the basic capabilities(****UNIX / Linux****)***

***Network operating system (NOS)*** provides services to clients over a network. Both the client/server and peer-to-peer networking models use network operating systems, and as such, NOSes must be able to handle typical network duties such as the following:

* Providing access to remote printers, managing which users are using which printers when, managing how print jobs are queued, and recognizing when devices aren't available to the network
* Enabling and managing access to files on remote systems, and determining who can access what—and who can't
* Granting access to remote applications and resources, such as the Internet, and making those resources seem like local resources to the user (the network is ideally transparent to the user)
* Providing routing services, including support for major networking protocols, so that the operating system knows what data to send where
* Monitoring the system and security, so as to provide proper security against viruses, hackers, and data corruption.
* Providing basic network administration utilities (such as SNMP, or Simple Network Management Protocol), enabling an administrator to perform tasks involving managing network resources and users.

UNIX / Linux

* UNIX, created originally by Bell Labs (under AT and T), is a powerful server operating system that can be used in peer-to-peer or client/server networks. UNIX was the first operating system written in the C programming language. Due to an antitrust ruling forbidding AT and T from releasing operating systems commercially, AT and T released UNIX upon its completion in 1974 to universities, mostly, enabling people to go in and actually view the source code to the system, which enabled coders to reconstruct the basic functions of the original UNIX operating system.

**UNIX (and Linux by extension) systems offer the following features:**

* ***Fully protected multitasking:*** This means that UNIX can easily switch between tasks without the operating system crashing, because all UNIX processes are separate from those of the operating system. Even if an application crashes, unless it somehow manages to take down the X Windows system with it (which does happen), the operating system just keeps right on humming.
* ***High performance and stability :*** Many servers running UNIX or Linux have run for years without crashing once. The multitasking capabilities of UNIX, along with the rapid rate at which the operating system matures (especially with Linux, which is free and can be changed by anyone), make UNIX or Linux a powerful solution, especially for server systems.
* ***Multiuser capabilities :*** True multiuser systems enable different users to be logged in to the same system simultaneously. In UNIX and Linux, not only can a user log in to the same system at the same time as other users, that user can log in multiple times on the same system as the same user without the operating system batting an eyelash (such things are often necessary when administrating a network, particularly when managing users).
* ***Tons of high-quality software :*** From Apache Server (a Web server that’s used on a whopping 6 in 10 major Web servers on the Internet) to the long-awaited Mozilla.org Mozilla 1.0 open source Web browser/e-mail software (Mozilla is an open source version of the venerated Netscape Communicator) to the powerful free Gimp graphics manipulation software, Linux is packed with tons of free, high-quality software. The trick is that, with UNIX/Linux, you give up compatibility with commercial software that’s available only for Windows and/or Macintosh, currently.
* ***Easy customization :*** While other operating systems seem to offer less and less choice to the user about which applications to install with the operating system (Windows XP is this way), UNIX and especially Linux are the exact counterpoint to that model. With UNIX or Linux, you can actually customize your operating system kernel, stripping it down to just drivers and networking or installing everything possible.
* ***Modular architecture :*** The modular architecture of UNIX (and especially Linux) is directly responsible for how customizable UNIX is. Modular really means just what it sounds like: The operating system is built with a kernel that attaches modules to itself based on what the user needs.
* ***POSIX compliance:*** With a free operating system like UNIX, the different distributions (or flavors) of UNIX quickly became difficult to manage. Currently, hundreds of different implementations of UNIX are available. To enable programmers to have some idea of how to code their software such that it would run on any version of UNIX, the Institute of Electrical and Electronics Engineers, Inc. (IEEE) defined the Portable Operating System Interface (POSIX).
* ***Use of TCP/IP as the standard protocol stack:*** UNIX overwhelmingly uses TCP/IP as the protocol stack of choice. If you consider that the vast majority of the servers that help make up the Internet are UNIX computers of one form or another, you start to get the idea why TCP/IP is so popular.
* ***A shell interface:*** All versions of UNIX (at least those you care about for the exam) include a shell interface of some sort. If you have ever seen your computer use a completely black screen with white words written on it, that’s a shell interface. You simply type in commands at the prompt and hit Enter to execute those commands. The hard part in using these interfaces is simply the effort it takes to learn all of those rather cryptic commands. Making life even more difficult, UNIX is ultimately customizable and can use different shells. The bash shell (likely the most popular shell in use today) and the tcsh shell, for example, have different commands for the same action.
* ***A graphical user interface:*** Although most versions of UNIX (such as Red Hat Linux) include a graphical user interface (GUI) these days, this has not always been the case. Historically, UNIX has been derided for its cryptic interface, and the advent of the GUI into popular UNIX systems was a direct result of this. Popular UNIX GUIs include KDE and GNOME. KDE is mostly used with Linux, but GNOME has versions for the Sun Solaris operating system, and therefore crosses the border from Linux into UNIX proper.
* ***Support for dumb terminals:*** Traditionally, UNIX was used for dumb terminals, and just about all versions of UNIX still include this capability. The traditional dumb terminal model involves one central UNIX server that is used by remote terminals to execute applications. Basically, a user logs in to a UNIX system via Telnet or some other remote connectivity application and uses UNIX commands to tell the remote system what functions to perform. In this way, users can download and check e-mail via a text-based e-mail client such as Pine. The dumb terminal in this form isn’t used much anymore; Web browsers are definitely more than just dumb terminals— and Web browsers are now the clients most often seen by UNIX servers (at least those that make up the Internet). However, wireless devices such as cell phones and mobile Internet e-mail clients such as AOL’s Mobile Communicator device are good examples of modern dumb terminals. The devices have nearly no storage at all, and don’t carry large e-mail clients on the device; the message is simply transferred as text from one end to the other.

**Interoperability**
Open source software such as SAMBA is used to provide Windows users with Server Message Block (SMB) file sharing.

**Authentication**:-Centralized login authentication

**File and Print Services**
Network File System (NFS) is a distributed file system that allows users to access files and directories located on remote computers and treat those files and directories as if they were local.
LPR/LPD is the primary UNIX printing protocol used to submit jobs to the printer. The LPR component initiates commands such as "print waiting jobs," "receive job," and "send queue state," and the LPD component in the print server responds to them.

**Security**
With most Unix operating systems, the network services can be individually controlled to increase security.