DEC
2015

BSD Associate | *Exam Objectives*

ABSTRACT

The official Exam Objectives for the BSD Associate (BSDA) certification exam, including detailed descriptions and study aides.

Version 2.0

Cover design by Jenny Rosenberg

Preface

Welcome! This document introduces the BSD Associate (BSDA) examination and describes in considerable detail the objectives covered by the exam. The exam covers material across all four major projects of BSD Unix - NetBSD, FreeBSD, OpenBSD and DragonFly BSD.

While the testing candidate is expected to know concepts and practical details from all four main projects, it is not necessary to know all the details of each one. A thorough reading of this document is recommended to understand which concepts and practical details are expected to be mastered.

Throughout this document, a clear distinction is placed on 'recognizing' and 'understanding', versus 'demonstrating' and 'performing'. Certain objectives call for the mere understanding of certain topics, while others call for the ability to demonstrate performance level knowledge of the topic. The difference is an important one, and should be remembered.

Successful mastery of the BSDA examination will, in most cases, require study and practice. The requirements for the exam encompass more background in BSD than is common among casual users or those new to BSD. This is a deliberate decision by the BSD Certification Group-to encourage more cross learning among BSD systems so that breadth of understanding of BSD is as heavily tasked as depth of understanding. The result will be a more well-rounded BSD advocate and a more knowledgeable system administrator.

The BSD Certification Group

www.bsdcertification.org

December 21 , 2015

Table of Contents

Executive Summary

This document describes the requirements of the BSD Associate (BSDA) certification developed by the BSD Certification Group. It outlines seven *study domains*, each of which expands to about a dozen or more exam objectives. These objectives form the essentials that must be mastered by prospective candidates for this exam.

The BSDA exam covers topics in system administration for several versions of BSD Unix:

- NetBSD: 7.x or higher
- FreeBSD: 10.x or higher
- OpenBSD: 5.x or higher
- DragonFly BSD: 4.x or higher

The BSDA exam is version-agnostic in that exam questions are not specific to a particular version of BSD. Any version that is equal to or higher than that listed above is suited to study purposes.

The seven study domains cover installation and upgrades, security, files and disks, user accounts, networking, and several other basic skills in BSD Unix system administration. This document also includes a useful cross reference tool (in Appendix A) that contains the complete list of the commands referenced in the exam objectives.

The successful candidate can be expected to have a solid grasp of BSD Unix administration and be able to complete many common tasks working under or alongside more senior administrators.

1 Introduction

This document describes the requirements of the BSD Associate (BSDA) certification developed by the BSD Certification Group. The BSDA certification is the first in a series of certifications being developed by the Group to promote BSD Unix system usage around the world.

This document serves two purposes. It describes the certification itself, including the name, description, operating system and version coverage, and recertification requirements. It also introduces seven s*tudy domains*, which are the focus of the exam. Various exam objectives under each study domain are then elaborated in more detail. Taken as a whole, the study domains define the minimum essential knowledge needed to confidently administer BSD systems at the associate level.

This document will be of interest to:

- Potential testing candidates for gauging their current experience against the exam requirements.
- Trainers and educational institutions who wish to include BSD training in their offerings.
- Courseware and study guide developers who plan to create materials to assist candidates in learning the skills needed to meet the exam requirements.

Since there are no officially recommended study materials or training institutions, each exam objective details both the theory and commands the BSDA candidate can expect to be tested on. It is likely that many candidates who take the examination will already have most of the required skills and will be able to use self-study to brush up on the rest of the needed skills. Self-study will include each BSD project's handbook and online documentation, man pages, existing tutorials and how-tos, and the results of Internet searches.

There will also be candidates who wish to enter the field of BSD administration and who would benefit from specific study guides or a training program. For this reason, the BSD Certification Group makes the following recommendations:

- That trainers and developers of instructional material include all exam objectives. In addition, the material should provide easy-to-understand background concepts to allow the student to gain understanding of "why" as well as hands-on labs so the student can master the "how".
- That candidates considering purchasing either training or study materials carefully compare their purchase to the exam objectives found in this document. Verify before purchase that each objective is covered to the level of detail required for the candidate to gain mastery of the BSDA examination requirements.

1.1 Definition of Audience for BSDA

The BSDA certification is designed to be an entry-level certification on BSD Unix systems administration. Testing candidates with a general Unix background, but less than six months of work experience as a BSD systems administrator (or who wish to obtain employment as a BSD systems administrator) will benefit most from this certification. Human resource departments should consider the successful BSDA certified applicant to be knowledgeable in the daily maintenance of existing BSD systems under the direction and supervision of a more senior administrator.

The successful BSDA candidate is able to complete common administrative and troubleshooting tasks and has a good understanding of general BSD Unix and networking principles. In addition, the successful candidate demonstrates basic skills with these BSD operating systems: DragonFly BSD, FreeBSD, NetBSD and OpenBSD.

This does not mean that the candidate needs to learn the complete details of four operating systems. It does mean that the candidate is aware of the basic utilities common to these operating systems, and where specified in the exam objectives, of features unique to some of the BSD operating systems. It is recommended that the testing candidate have access to each of these freely available operating systems while preparing for the exam. Online resources such as manual pages, wikis, and email lists also exist for all these BSD versions.

Note: The completion of the BSDA certification is not a requirement for the BSDP (BSD Professional) examination. If you feel that you already have the skills needed as a senior administrator, you can instead consider taking the BSDP examination.

1.2 Official BSDA Examination Description

The following is the official BSDA examination description, suitable for display by training and testing centers and for use within training materials:

“The BSDA (BSD Associate) Certification is for those with light to moderate skills in BSD Unix system administration. The successful candidate understands and uses basic system administration commands and demonstrates proficiency in the following areas:

1. Installing and Upgrading the Operating System and Software
2. Securing the Operating System
3. Files, Filesystems, and Disks
4. Users and Accounts Management
5. Basic System Administration
6. Network Administration
7. Basic Unix Skills

Each area contains detailed exam objectives that are fully described in the 'BSDA Certification Requirements Document', available at the BSD Certification Group website www.bsdcertification.org."

1.3 Operating System Versions Covered by BSDA

In order for a certification to be practical, it must cover working knowledge in versions of BSD operating systems most likely to be seen in the workplace. The BSDA exam objectives cover material applicable to the following versions of these operating systems:

- NetBSD: 7.x or higher
- FreeBSD: 10.x or higher
- OpenBSD: 5.x or higher
- DragonFly BSD: 4.x or higher

The testing candidate should keep in mind that the base utilities encountered in the BSDA examination rarely change from one operating system revision to another. Any operating system features which are not specified in the exam objectives will not be encountered on the exam.

1.4 Recertification Requirements

One of the requirements of a psychometrically valid certification program is that each certification must have an expiration date. A certified individual must periodically provide evidence that they still have the necessary skills required by the certification.

The BSDA certification expires at the end of five years. In order to remain BSDA certified, the certificate holder must apply for recertification using *one* of the two methods listed below. These methods reflect standard psychometric practices as well as the community-based approach of the BSD Certification Group.

BSDA certified individuals can apply for recertification up to six months before their certification expires. Individuals with a unique circumstance requiring an earlier recertification should contact the BSDCG. There is a six month renewal grace period after a certification expires. When a successful application for recertification occurs, the certification will be renewed for five years starting at the expiration date of the prior certification. If recertification is not attempted during this period, the individual is no longer considered certified and must meet the BSDA certification requirements as a new certification applicant should they wish to pursue certification at a later time.

When applying for a recertification, select *one* of the two renewal methods:

Method 1: Retake and pass the BSDA exam at either a hosted exam event or at a testing center. The BSDA Certification Requirements document should be reviewed as preparation for the exam.

Method 2: Provide evidence that you continue to use the skills covered by the BSDA Certification Requirements during the five years of certification. A list of acceptable activities, their BSDA Recertification Unit (BRU) values, and examples of verifiable evidence are provided in "Table 1: Evidence of Continuing BSD Skills Chart". In order to recertify, you must provide a total of ten BRUs and demonstrate how your listed activities relate to the exam objectives within the BSDA Certification Requirements document.

If you choose the second method of recertification, send an email which details your evidence to chair@bsdcertification.org and pay the recertification fee. Once your email and recertification payment are received, the BSDCG review committee will review your evidence. If clarification is needed, an email will be sent requesting further information. Once the evidence is validated as meeting the ten BRUs requirement, you will receive an email indicating the new expiry date and a new certificate will be shipped to the mailing address that you have provided to the BSDCG.

Table 1: Evidence of Continuing BSD Skills Chart

Activity	BRU Value	Example Evidence
Full-time employment as a BSD system administrator or consultant.	2 BRUs per year of full-time employment up to a maximum of 10 BRUs.	Provide employer name(s), dates of employment, the job duties which relate to the BSDA exam objectives, and contact information of a manager to verify employment dates and job duties. Employment must be within the 5 year period between certification date and certification expiry date.
Part-time employment as a BSD system administrator or consultant (10-30 hours per week).	1 BRU per year of part-time employment up to a maximum of 5 BRUs.	Provide employer name(s), dates of employment, the job duties which relate to the BSDA exam objectives, and contact information of a manager to verify employment dates and job duties. Employment must be within the 5 year period between certification

Activity	BRU Value	Example Evidence
		date and certification expiry date.
Provide BSD training.	1 BRU per day of training up to a maximum of 10 BRUs.	Include URL to course description, your trainer biography, and teaching dates. If this information is not available online, include employer's name and contact information.
Publication, article, or how-to directly related to BSD.	1 BRU per publication (minimum 1500 words).	Include URLs to online publications or page numbers within paper publications, In the event that a publication is only available in print periodicals or behind a "pay wall," attach a copy of the article.
BSD presentation at a conference, user group, or post-secondary institution.	1 BRU per presentation (minimum of 30 minutes).	Include URLs to presentations or your speaker biography at the event's website. If the presentation is not available online, attach a copy of the presentation.
Diploma course.	2 BRUs per semester.	Provide a course outline and a copy of the diploma or course credit. List the BSDA exam objectives covered in the course.
Industry certification.	2 BRUs per certification.	Include a copy of the certification. List the BSDA exam objectives covered by the certification.
BSD technical blog.	1 BRU for every 5 posts averaging 300 words each.	Include URL to blog.
System administration of BSD server (e.g. mail server, webserver).	1 BRU for each yearly summary.	Server must run on BSD platform. Provide a yearly summary of administrative activities, such as the maintenance log, history of software patching, upgrades, DNS renewal, and scripting activities. Include URL of public server or contact information

Activity	BRU Value	Example Evidence
		of employer if server is private.
Contributions to a BSD project as a non-committer.	1 BRU per year of active contributions to a BSD project. This may include code, documentation, patches, system administration assistance to the project or for the project community.	Include URLs to patches, code, documentation or a summary of your contributions.
Code committed to a BSD project as a committer.	2 BRUs per year of active commits. Commits can be to code or documentation.	Include your valid BSD project email address and a summary URL of your commits.

2 Using the BSDA Study Domains

The BSDA certification exam covers seven study domains, each containing several detailed exam objectives:

1. Installing and Upgrading the Operating System and Software (10 objectives)
2. Securing the Operating System (13 objectives)
3. Files, Filesystems, and Disks (14 objectives)
4. Users and Accounts Management (9 objectives)
5. Basic System Administration (24 objectives)
6. Network Administration (15 objectives)
7. Basic Unix Skills (17 objectives)

Testing candidates should understand all exam objectives prior to taking the exam, even though not every exam objective may be tested on every version of the BSDA exam.

Note that **exam objectives** are not the same as **exam questions**. Exam objectives are publicly available for distribution whereas exam questions are the private intellectual property of the BSD Certification Group. Those who have access to exam questions- psychometricians, SMEs (Subject Matter Experts), translators, testing centers, and the testing candidate during the certification exam- are subject to an NDA (Non Disclosure Agreement) and are subject to penalties if the NDA is broken.

To assist the testing candidate in their studies as well as those who wish to provide training and study materials, each exam objective has three components:

1. Number and name of exam objective.
2. **Concept** which describes the background material the candidate should master in order to effectively put the exam objective into administrative practice. The level of depth should reflect the BSDA audience described in Section 1.1.
3. **Practical** which provides a list of commands and other content which may appear on the certification exam. Where applicable, the section of the manual is included and candidates are encouraged to familiarize themselves with the contents of specified man pages.

Note: When using the exam objectives, keep the following points in mind:

- Remember that the BSDA certification exam tests practical skills: don't just memorize man pages. Instead, practice the commands until you understand the man pages.
- This is an entry-level examination. You are not expected to know everything, but you are expected to demonstrate that you have experience completing common administrative tasks.
- Watch for the word "recognize" in an exam objective. This indicates that you need to know something exists but not necessarily how to set it up. An example of this is Objective 2.2.11 where the candidate should recognize that BSD systems have the ability to restrict a service and which utilities are available to do so on each BSD. As a BSDA candidate, however, you are not expected to demonstrate experience in configuring a jail.
- Where major differences in the utilities used to accomplish a task on the various BSDs exist, it may be noted in the Practical component but the details regarding all differences are not. Instead, you are encouraged to refer to the cross reference table in Appendix A as you prepare for the exam. This table contains an alphabetical list of the required commands and their availablity on the four listed BSD operating systems.

2.1 Domain 1: Installing and Upgrading the OS and Software

An important aspect of system administration is tracking installed versions of both the operating system and third-party applications. An advantage of using BSD systems is the availability of multiple tools to assist the system administrator in determining software versions and their dependencies. These tools indicate which software is out-of-date or has existing security vulnerabilities. They also assist in upgrading or patching software and its dependencies.

When and how installations and upgrades are done is specific to each organization. The successful BSDA candidate knows how to use the tools which are available for these purposes, and the cautions that are necessary when working on production systems under the supervision of a more senior administrator.

2.1.1 Recognize the installation program used by each operating system.

Concept: While BSDA candidates are not expected to plan an installation, they should be able to start and complete an installation according to a provided list of requirements. Since the install procedure is operating system dependent, it is recommended that the candidate have prior experience in the default install routine for each tested BSD operating system.

Practical:

DragonFly BSD: installer

FreeBSD: bsdinstall(8)

NetBSD: sysinst(8)

OpenBSD: bsd.rd

2.1.2 Recognize which commands are available for upgrading the operating system.

Concept: The BSDA candidate should recognize the utilities which are used to keep the operating system up-to-date.

Practical:

DragonFly BSD/FreeBSD: build(7), make(1) including the buildworld and installworld targets, etcupdate(8), mergemaster(8)

NetBSD: sysinst(8), etcupdate(8), postinstall(8)

OpenBSD: bsd.rd, sysmerge(8)

2.1.3 Understand the difference between a pre-compiled binary and compiling from source.

Concept: The candidate should be familiar with using the ports, packages, and pkgsrc collections In addition, the candidate should be aware of the advantages and disadvantages of installing a pre-compiled binary and the advantages and disadvantages of compiling a binary from source.

DragonFly BSD: dports(7)

FreeBSD/OpenBSD: ports(7)

OpenBSD: packages(7)

2.1.4 Understand when it is preferable to install a pre-compiled binary and how to do so.

Concept: The BSDA candidate should be aware that while pre-compiled binaries are quick and easy to install, they don't allow the customization of the binary to a system's particular needs. The candidate needs to know how to install a pre-compiled binary, as well as how to uninstall a pre-compiled binary.

Practical:

DragonFly BSD/FreeBSD: pkg-install(8), pkg-delete(8)

NetBSD/OpenBSD: pkg_add(1), pkg_delete(1)

2.1.5 Recognize the available methods for compiling a customized binary.

Concept: Many applications used by servers support make(1) options to compile a binary with the feature set required by a particular installation. While the BSDs all use make(1), the BSDA candidate should recognize that each BSD uses different mechanisms to use and preserve make(1) options.

Practical:

DragonFly BSD: make.conf(5), PKG_OPTIONS, CFLAGS

FreeBSD: -DWITH_* or WITH_*=, make.conf(5)

NetBSD: PKG_OPTIONS.<pkg>, CFLAGS, mk.conf(5), PKG_DEFAULT_OPTIONS

OpenBSD: bsd.port.mk(5), mk.conf(5)

2.1.6 Determine which software is installed on a system.

Concept: The BSDA candidate should recognize that on BSD systems, package managers track software and dependencies if the software was installed using packages, ports or pkgsrc. The candidate should be familiar with querying the package manager to determine which software and their versions are installed on the system.

Practical:

DragonFly BSD/FreeBSD: pkg-info(8)

NetBSD/OpenBSD: pkg_info(1)

2.1.7 Determine which software requires upgrading.

Concept: The candidate should recognize the importance of balancing the need to keep software up-to-date while minimizing the impact on a production system. The candidate should recognize which tools are used to determine which third-party software requires updating on each BSD system.

Practical:

DragonFly BSD/FreeBSD: pkg-version(8)

NetBSD: pkgsrc(7), pkgtools

OpenBSD: pkg_add(1)

2.1.8 Upgrade installed software.

Concept: The BSDA candidate should recognize the built-in and third-party commands which are available for upgrading installed software on BSD systems.

Practical:

DragonFly BSD/FreeBSD: pkg-upgrade(8)

NetBSD: pkg_add(1), make update

OpenBSD: pkg_add(1)

2.1.9 Determine which software have outstanding security advisories.

Concept: The BSDA candidate should recognize the importance of being aware of software security vulnerabilities. The candidate should also recognize the utilities that can be used to determine which installed software have outstanding vulnerabilities.

Practical:

DragonFly BSD/FreeBSD: pkg-audit(8)

NetBSD: pkg_admin(1)

OpenBSD: https://www.mtier.org/solutions/apps/openup/

2.1.10 Follow the instructions in a security advisory to apply a security patch.

Concept: The BSDA candidate should be aware that each BSD project maintains security advisories which are available both on the Internet and via mailing lists. The candidate should be able to follow the instructions in an advisory when asked to do so by a supervisor.

Practical:

patch(1)

2.2 Domain 2: Securing the Operating System

The mark of a good system administrator is the awareness of and adherence to best security practices. The BSDA candidate is expected to be familiar with common security practices. BSD systems are designed with security in mind and provide many mechanisms which allow the system administrator to tune systems to the security requirements of an organization. While the BSDA candidate won't always be responsible for implementing these mechanisms, the candidate should be able to recognize the features and commands available for securing BSD systems.

2.2.1 Determine or set the system's security level.

Concept: BSD systems provide security profiles known as securelevels. The BSDA candidate should be able to recognize the restrictions set by each securelevel for each BSD operating system. The candidate should also understand under what circumstances a securelevel can be raised or lowered.

Practical:

init(8), sysctl(8)

DragonFly BSD/FreeBSD/NetBSD: rc.conf(5)

OpenBSD: rc.conf(8)

2.2.2 Configure an SSHD server according to a set of requirements.

Concept: The BSDA candidate should be aware that the sshd(8) built into BSD systems can be configured to limit who can access a system via SSH.

Practical:

sshd_config(5) including these keywords: AllowUsers, Banner, MaxStartups, PermitRootLogin, X11Forwarding

2.2.3 Configure an SSHD server to use a key pair for authentication.

Concept: The candidate should understand private/public key theory including: which protocols are available for generating key pairs, choosing an appropriate bit size, providing a seed, providing a passphrase, and verifying a fingerprint. In addition, candidates should be able to generate their own keys and use them for authentication.

Practical:

ssh-keygen(1), .ssh/authorized_keys

2.2.4 Preserve existing host keys during a system upgrade.

Concept: In addition to knowing how to generate a system's SSH keys, the BSDA candidate should know where host keys are located and how to preserve them if the system is upgraded or replaced.

Practical:

/etc/ssh/

2.2.5 Recognize alternate authentication mechanisms.

Concept: The BSDA candidate is not expected to know how to configure an alternate authentication mechanism. However, candidates should understand basic authentication theory and be aware that providing a username and password is only one way to authenticate on BSD systems. Candidates should have a basic understanding of PAM and know it is available on DragonFly BSD, FreeBSD and NetBSD. They should also understand basic theory regarding Kerberos, OTP and RADIUS.

2.2.6 Recognize alternate authorization schemes.

Concept: Candidates should understand basic authorization theory and which frameworks extend the features provided by the standard Unix permissions.

Practical:

DragonFly BSD: acl(3)

FreeBSD: mac(4), acl(3)

FreeBSD/NetBSD: extattrctl(8), getextattr(1)

NetBSD: kauth(9)

OpenBSD: systrace(1)

2.2.7 Recognize basic recommended access methods.

Concept: BSDA candidates should be familiar with standard system administration practices used to minimize the risks associated with accessing a system. These include using ssh(1) instead of telnet(1), denying root logins, using the possibly third-party sudo utility instead of su(1), and minimizing the use of the wheel group.

Practical:

ttys(5), sshd_config(5)

DragonFly BSD: sudo(8), visudo(8), sudoers(5)

DragonFly BSD/FreeBSD/NetBSD: ftpusers(5)

OpenBSD: doas(1), doas.conf(5)

2.2.8 Recognize BSD firewalls and rulesets.

Concept: Each BSD comes with at least one built-in firewall. The BSDA candidate should recognize which firewalls are available on each BSD and which commands are used to view each firewall's ruleset.

Practical:

pf(4), pfctl(8)

DragonFly BSD/FreeBSD: ipfw(8)

FreeBSD/NetBSD: ipf(8), ipfstat(8)

NetBSD: npfctl(8), npf(7)

2.2.9 Recognize BSD mechanisms for encrypting devices.

Concept: The BSDA candidate should be aware that it is possible to encrypt devices on BSD systems and which utilities are available on each BSD system.

Practical:

DragonFly BSD: tcplay(8)

FreeBSD: gbde(4), gbde(8), geli(8)

NetBSD: cgd(4), vnd(4)

OpenBSD: vnd(4), softraid(4)

2.2.10 Recognize methods for verifying the validity of files.

Concept: BSDA candidates should recognize the usefulness of file integrity utilities. They should also recognize the built-in checks provided on the BSDs.

Practical:

mtree(8)

DragonFly BSD/FreeBSD/NetBSD: security(7)

NetBSD: security.conf(5), veriexecctl(8)

OpenBSD: security(8)

2.2.11 Recognize the BSD methods for restricting a service.

Concept: The BSDA candidate should recognize the advantages of restraining a service on an Internet facing system and which utilities are available to do so on each of the BSDs.

Practical:

chroot(8), inetd(8), the possibly third-party Xen

DragonFly BSD: jail(8)

FreeBSD: jail(8), bhyve(8), xen(4)

NetBSD: kauth(9)

OpenBSD: systrace(1)

2.2.12 Change the encryption algorithm used to encrypt the password database.

Concept: The BSDA candidate should be able to modify the encryption algorithm used by the password database. The candidate should also have a basic understanding of DES, MD5, SHA1, and Blowfish.

Practical:

DragonFly BSD/FreeBSD/OpenBSD: login.conf(5)

NetBSD: passwd.conf(5)

2.2.13 Modify the system banner.

Concept: The candidate should be aware of the banner(s) that may be seen, depending on how a user accesses a system, and which files are used to configure each banner.

Practical:

motd(5), login.conf(5), gettytab(5), sshd_config(5)

2.2.14 Protect authentication data.

Concept: To prevent attacks against system security with password cracking attacks, BSD systems keep encrypted passwords visible to system processes only. The BSDA candidate should have an understanding of the location of the password database files.

Practical:

passwd(5), pwd_mkdb(8)

2.3 Domain 3: Files, Filesystems, and Disks

The usefulness of any computing system is related to the accessibility of the data stored on it. The BSDA candidate is expected to thoroughly understand how to make data available both locally and remotely and how to use permissions to ensure authorized users can access that data. The candidate should be experienced in backing up data and in resolving common disk issues.

2.3.1 Mount or unmount local filesystems.

Concept: The BSDA candidate should be familiar with all aspects of mounting and unmounting UFS and ZFS filesystems, including: how to mount or unmount a specified filesystem, how to mount all filesystems, configuring filesystems to be mounted at boot, passing options to mount(8), and resolving mount(8) errors.

Practical:

mount(8), umount(8), fstab(5)

FreeBSD/NetBSD: zfs(8)

2.3.2 Configure data to be available through NFS.

Concept: BSDA candidates should be aware of the utilities associated with NFS (Network File System), the configuration file for defining NFS mount points, and the security risks associated with allowing RPC through a firewall.

Practical:

exports(5), nfsd(8), mountd(8), rpc.lockd(8), rpc.statd(8)

DragonFly BSD/FreeBSD/NetBSD: rpcbind(8), rc.conf(5)

OpenBSD: portmap(8), rc.conf(8)

2.3.3 Determine which local filesystems are currently mounted and which will be mounted at system boot.

Concept: The candidate should be able to determine which local filesystems are currently mounted and which will be mounted at boot time.

Practical:

mount(8), df(1), fstab(5)

FreeBSD/NetBSD: zfs(8)

2.3.4 Determine disk capacity and which files are consuming the most disk space.

Concept: The BSDA candidate should be able to combine common Unix command line utilities to quickly determine which files are consuming the most disk space.

Practical:

du(1), df(1), find(1), sort(1), systat(1)

2.3.5 Create and view symbolic or hard links.

Concept: Candidates should know the difference between symbolic and hard links as well as how to create, view, and remove both types of links. In addition, the candidate should be able to temporarily resolve a low disk space issue using a symbolic link.

Practical:

ln(1), ls(1), rm(1), stat(1)

2.3.6 View ACLs.

Concept: The BSDA candidate should be able to determine if a FreeBSD system is using ACLs, and if so, on which filesystems. In addition, the candidate should be able to view a file's ACL on a FreeBSD system.

Practical:

mount(8), ls(1), getfacl(1)

2.3.7 View file permissions and modify them using either symbolic or octal mode.

Concept: The BSDA candidate is expected to have a thorough understanding of traditional Unix permissions, including: how to view and modify permissions, why the sticky bit is important on /tmp and other shared directories, recognizing and using the SUID and SGID bits, and the difference between symbolic and octal mode. In addition, the candidate should understand that a shell setting determines the default file and directory permissions and, given a umask value, the candidate should be able to determine the default permission set.

Practical:

ls(1), chmod(1), sh(1), csh(1), umask(2)

DragonFly BSD/FreeBSD: builtin(1)

DragonFly BSD/NetBSD/OpenBSD: ksh(1)

2.3.8 Modify a file's owner or group.

Concept: The BSDA candidate should be able to modify a file's ownership as required and be aware of how a file's original ownership is created.

Practical:

chown(8), chgrp(1), su(1), mtree(8)

2.3.9 Backup and restore a specified set of files and directories to local disk.

Concept: Candidates should have experience using common Unix command line backup utilities. In addition, candidates should be able to recognize the device names for tape devices on BSD systems.

Practical:

tar(1), cpio(1), pax(1), cp(1)

DragonFly BSD: cpdup(1)

2.3.10 Backup and restore a filesystem.

Concept: Candidates should recognize the utilities used to backup an entire UFS filesystem and the various dump(8) levels. In addition, the candidate should recognize the commands used to create a snapshot of a ZFS filesystem.

Practical:

dump(8), restore(8), dd(1)

FreeBSD/NetBSD: zfs(8)

2.3.11 Determine the directory structure of a system.

Concept: BSDA candidates should be able to quickly determine the directory layout used by BSD systems.

Practical:

hier(7)

2.3.12 Manually run a filesystem checker and repair tool.

Concept: The BSDA candidate should be aware of the utilities available to check the consistency of a UFS or ZFS filesystem.

Practical:

fsck(8)

FreeBSD/NetBSD: zpool(8)

2.3.13 View and modify file flags.

Concept: The BSDA candidate should understand how file flags augment traditional Unix permissions and should recognize how to view and modify the immutable, append-only, and undelete flags.

Practical:

ls(1), chflags(1)

2.3.14 Monitor the virtual memory system.

Concept: The virtual memory subsystem may have an important impact on a system's overall performance. The BSDA candidate should have an understanding of how to monitor swapping and paging activity to help determine if this subsystem requires tuning.

Practical:

pstat(8), systat(1), top(1), swapctl(8),

DragonFly BSD/FreeBSD/OpenBSD: vmstat(8)

NetBSD: vmstat(1)

2.4 Domain 4: Users and Accounts Management

All systems require at least one user account, and depending upon the role of the system, the BSDA candidate's job duties may include supporting end-users in the maintenance of their accounts. Accordingly, the BSDA candidate should be able to create user accounts, modify account settings, disable accounts, and reset passwords. The candidate should also know how to track account activity and determine which accounts are currently accessing a system.

2.4.1 Create, modify and remove user accounts.

Concept: Managing user accounts is an important aspect of system administration. The BSDA should be aware that the account management utilities differ across BSD systems and should be comfortable using each utility according to a set of requirements.

Practical:

vipw(8), adduser(8)

DragonFly BSD/FreeBSD: pw(8)

DragonFly BSD/FreeBSD/NetBSD: adduser.conf(5)

DragonFly BSD/FreeBSD/OpenBSD: rmuser(8)

NetBSD/OpenBSD: useradd(8), userdel(8), userinfo(8), usermod(8), user(8)

2.4.2 Create a system account.

Concept: Candidates need to understand that many services require an account and that such accounts should not be available for logins.

Practical:

nologin(8); using a * in the password field of passwd(5)

2.4.3 Lock a user account or reset a locked user account.

Concept: The BSDA candidate should know how to recognize a locked account and how to remove the lock.

Practical:

vipw(8); chpass(1), chfn(1), chsh(1)

DragonFly BSD/FreeBSD: pw(8)

NetBSD/OpenBSD: usermod(8), user(8)

2.4.4 Determine identity and group membership.

Concept: In the context of the Unix permission system, determining one's identity and group membership is essential to determine what authorizations are available. The BSDA should be able to determine and, as required, change identity or group membership.

Practical:

id(1), groups(1), who(1), whoami(1), su(1)

2.4.5 Determine who is currently on the system or the last time a user was on the system.

Concept: BSD systems maintain databases which can be queried for details regarding logins. The BSDA candidate should be familiar with the database names and the utilities available for determining current and past login information.

Practical:

w(1), who(1), users(1), last(1), ac(8)

DragonFly BSD/FreeBSD/NetBSD: lastlogin(8)

DragonFly BSD/NetBSD/OpenBSD: lastlog(5), wtmp(5), utmp(5)

2.4.6 Enable accounting and view system usage statistics.

Concept: The BSDA candidate should be aware of when it is appropriate to enable system accounting, recognize which utility is available to do so, and know how to view and analyze the resulting accounting details.

Practical:

sa(8), accton(8), lastcomm(1)

2.4.7 Change a user's default shell.

Concept: Candidates should know the default shells for both user accounts and the superuser account for each BSD. In addition, they should know how to change the default shell for each BSD operating system.

Practical:

vipw(8); chpass(1), chfn(1), chsh(1)

DragonFly BSD/FreeBSD: pw(8)

NetBSD/OpenBSD: usermod(8), user(8)

2.4.8 Control which files are copied to a new user's home directory during account creation.

Concept: BSD systems use a “skel” directory containing files which are copied over to a user's home directory when a user account is made. The BSDA candidate should be aware of the location of the skel directory on each BSD, as well as how to override the copying of its contents during account creation.

Practical:

DragonFly BSD/FreeBSD: pw(8)

DragonFly BSD/FreeBSD: adduser.conf(5)

NetBSD/OpenBSD: useradd(8). usermgmt.conf(5)

2.4.9 Change a password.

Concept: The BSDA candidate should be able to change their own password as well as the passwords of other users.

Practical:

passwd(1), vipw(8)

2.5 Domain 5: Basic System Administration

An important component of system administration is an awareness of its subsystems and their interactions, as well as how to monitor the health of a running system. The BSDA candidate needs to demonstrate experience in interacting with BSD processes, a running kernel, and the BSD boot process. Candidates should also demonstrate familiarity with BSD devices, the disk subsystem, and the mail and print daemons.

2.5.1 Determine which processes are consuming the most CPU.

Concept: The BSDA candidate should be able to view active processes and recognize inordinate CPU usage. In addition, the candidate should know how to end a process or change its priority.

Practical:

top(1), systat(1), ps(1), nice(1), renice(8), kill(1), pkill(1)

2.5.2 View and send signals to active processes.

Concept: The BSDA candidate should be familiar with both the names and numbers of the most commonly used Unix signals and how to send a signal to an active process. The candidate should also recognize the difference between a SIGTERM and a SIGKILL.

Practical:

ps(1), kill(1),pkill(1), pgrep(1)

DragonFly BSD/FreeBSD: killall(1)

2.5.3 Use an rc(8) script to determine if a service is running and start, restart, or stop it as required.

Concept: In addition to directly sending signals to processes, the BSDA candidate should realize that BSD systems provide scripts which can be used to check the status of services and to stop, start, and restart them as required. The candidate should also be aware of the locations of these scripts on each of the BSD systems.

Practical:

rc(8)

DragonFly BSD/FreeBSD/NetBSD: service(8), rc.conf(5)

OpenBSD: rcctl(8), rc.conf(8)

2.5.4 View and configure system hardware.

Concept: BSD systems come with many utilities to determine what hardware is installed on a system. The BSDA candidate should know how to determine which hardware was probed at boot time as well as some BSD-specific utilities which can be used to troubleshoot and manipulate PCI, ATA, and SCSI devices.

Practical:

dmesg(8)

DragonFly BSD/FreeBSD: pciconf(8), camcontrol(8), devinfo(8)

NetBSD/OpenBSD: atactl(8)

NetBSD: scsictl(8), pcictl(8)

OpenBSD: scsi(8)

2.5.5 View, load, or unload a kernel module.

Concept: The BSDA candidate should understand the difference between a statically compiled kernel and one that uses loadable kernel modules. The candidate should be able to view, load, and unload kernel modules and also be aware that kernel modules are not available on OpenBSD systems as of OpenBSD version 5.7.

Practical:

DragonFly BSD/FreeBSD: kldstat(8), kldload(8), kldunload(8), loader.conf(5)

NetBSD: modstat(8), modload(8), modunload(8), module(7)

2.5.6 Modify a kernel parameter on the fly.

Concept: BSD systems maintain kernel MIB variables which allow a system administrator to both view and modify the kernel state of a running system. The BSDA should be able to view and modify these MIBs both at run-time and permanently over a system boot. The candidate should also recognize how to modify a read-only MIB.

Practical:

sysctl(8), sysctl.conf(5)

OpenBSD: config(8), boot.conf(8)

2.5.7 View the status of a software RAID mirror or stripe.

Concept: In addition to providing drivers for hardware RAID devices, BSD systems also provide built-in mechanisms for configuring software RAID systems. The BSDA candidate should know the difference between RAID levels 0, 1, 3 and 5 and recognize which utilities are available to configure software RAID on each BSD system.

Practical:

DragonFly BSD/FreeBSD/NetBSD: ccdconfig(8)

DragonFly BSD: vinum(8)

FreeBSD: gmirror(8), gstripe(8), graid3(8), zpool(8)

NetBSD:raidctl(8)

NetBSD/OpenBSD: bioctl(8)

2.5.8 Determine which MTA is being used on the system.

Concept: The BSDA candidate should recognize the role of the MTA (Mail Transfer Agent), recognize which MTA(s) are available during each BSD operating system's install routine and which configuration file indicates the MTA in use on the system. The candidate should also recognize the difference between the mbox or maildir mail destination file format type.

Practical:

mailer.conf(5)

2.5.9 Configure log rotation.

Concept: The BSDA candidate must understand that the system automatically maintains the creation and maintenance of many different logs. The candidate should be able to configure log rotation by either time or size and should be able to view compressed logs.

Practical:

Note that the newsyslog(8) implementations vary by BSD.

newsyslog(8), newsyslog.conf(5), syslog.conf(5), zmore(1)

DragonFly BSD/FreeBSD/NetBSD: bzcat(1)

2.5.10 View, reorder, or modify a job within the print queue.

Concept: The BSDA candidate should be able to view the print queue and manipulate the jobs within the queue. The candidate should also be able to recognize the meaning of /etc/printcap entries.

Practical:

lpc(8), lpq(1), lprm(1), printcap(5)

2.5.11 Review log files to troubleshoot and monitor system behavior.

Concept: Candidates should know the location of the log files and be aware of the importance of reviewing log files on a regular basis as well as how to watch a log file when troubleshooting.

Practical:

tail(1), syslog.conf(5), grep(1), dmesg(8)

2.5.12 Create or modify email aliases for Sendmail or Postfix.

Concept: The BSDA candidate should understand when to create an email alias and how to do so for both Sendmail and Postfix.

Practical:

aliases(5)

DragonFly BSD/FreeBSD/NetBSD: newaliases(1)

DragonFly BSD/NetBSD: postalias(1)

OpenBSD: newaliases(8)

2.5.13 Drop the system to single-user mode.

Concept: The BSDA candidate needs to understand the ramifications associated with dropping a system to single-user mode, recognize when it may be necessary to do so, and how to minimize the impact on a server system.

Practical:

shutdown(8), boot(8)

2.5.14 Recognize the difference between hard and soft limits and modify existing resource limits.

Concept: Candidates need to understand that resource limits are inherited by the shell as well as how to view their limits and change them both temporarily and permanently. In addition, candidates should understand the difference between soft and hard limits.

Practical:

login.conf(5), sysctl(8)

DragonFly BSD/FreeBSD: limits(1)

DragonFly BSD/FreeBSD/NetBSD: limit(1)

2.5.15 Recognize the BSD utilities that shape traffic or control bandwidth.

Concept: Candidates should understand when it is advantageous to create policies controlling the amount of bandwidth available to specified services. In addition, candidates should recognize the utilities available on BSD systems to create bandwidth policies.

Practical:

pf.conf(5)

DragonFly BSD/FreeBSD: ipfw(8), altq(4), dummynet(4), altq(9)

NetBSD: altq(4),altq(8), altq(9), altqd(8), altq.conf(5)

2.5.16 Recognize common, possibly third-party, server configuration files.

Concept: BSD systems are often used to provide Internet services. The BSDA candidate may be asked to view or make a specified change to a service's configuration file and needs to recognize the names of the most commonly used configuration files and which applications they are associated with.

Practical:

sendmail.cf

DragonFly BSD: smb.conf(5)

DragonFly BSD/FreeBSD/NetBSD: dhcpd.conf(5)

DragonFly BSD/NetBSD: master(8), named.conf(5)

DragonFly BSD/OpenBSD: httpd.conf(5)

2.5.17 Configure a service to start at boot time.

Concept: The BSDA candidate should recognize that the BSD boot process does not use runlevels. The candidate must be able to configure essential services to start at boot time to minimize the impact of a system reboot.

Practical:

rc(8), inetd(8)

DragonFly BSD/FreeBSD/NetBSD: rc.conf(5)

OpenBSD: rc.conf(8)

2.5.18 Configure the scripts that run periodically to perform various system maintenance tasks.

Concept: BSD systems provide many scripts that are used to maintain and verify the integrity of the system. The BSDA candidate should be able to locate and run these scripts manually as required as well as configure which scripts run daily, weekly, and monthly on each BSD system.

Practical:

DragonFly BSD/FreeBSD: periodic.conf(5), periodic(8)

NetBSD: security.conf(5), daily.conf(5), weekly.conf(5), monthly.conf(5)

OpenBSD: daily(8), weekly(8), monthly(8)

2.5.19 View the Sendmail or Postfix mail queue.

Concept: The BSDA candidate should be able to view the mail queue to determine if any mail is stuck in the queue, and if necessary, ask the MTA to reprocess or flush the queue.

Practical:

DragonFly BSD/FreeBSD/NetBSD: mailq(1)

DragonFly BSD/NetBSD: postqueue(1)

OpenBSD: smtpctl(8)

2.5.20 Determine the last system boot time and the workload on the system.

Concept: The BSDA candidate should be able to monitor the system's workload using the time since last system reboot, as well as the system load over the last 1, 5 and 15 minutes in order to determine operation parameters.

Practical:

uptime(1), w(1), top(1)

2.5.21 Monitor disk input/output.

Concept: A system's disk input/ouput can have a dramatic impact on performance. The BSDA candidate should know how to use the utilities available on BSD systems to monitor disk I/O.

Practical:

iostat(8), systat(1), vmstat(1), nfsstat(1)

DragonFly BSD/FreeBSD: gstat(8)

2.5.22 Deal with busy devices.

Concept: The BSDA candidate should understand what can cause a process to hang, how to detect related processes and how to fix the situation.

Practical:

ps(1), fstat(1), kill(1) pkill(1),and the third-party lsof utility

DragonFly BSD/FreeBSD: killall(1)

2.5.23 Determine information regarding the operating system.

Concept: The BSDA candidate should be able to determine the type and version of the operating system installed.

Practical:

uname(1), sysctl(8), dmesg(8)

NetBSD: /etc/release

2.5.24 Understand the advantages of using a BSD license.

Concept: The BSDA candidate should recognize the 2-clause BSD license and how the license does not place restrictions on whether BSD licensed code remains Open Source or becomes integrated into a commercial product.

2.6 Domain 6: Network Administration

TCP/IP was originally implemented on BSD systems and BSD systems continue to provide core networking services for a substantial portion of the Internet. The BSDA is expected to demonstrate a strong understanding of both IPv4 and IPv6 addressing as well as basic networking theory. Trainers and material providers should provide conceptual depth similar to that found in the CompTIA Network+ or in the networking theory section of the Cisco CCNA.

2.6.1 Determine the current TCP/IP settings on a system.

Concept: The BSDA candidate should be able to determine a system's IP address(es), subnet mask, default gateway, primary and secondary DNS servers, and hostname.

Practical:

ifconfig(8), netstat(1), resolv.conf(5), route(8), hostname(1)

OpenBSD: hostname.if(5)

2.6.2 Set a system's TCP/IP settings.

Concept: Candidates should be able to modify required TCP/IP settings both temporarily and permanently in order to remain after a reboot.

Practical:

hostname(1), ifconfig(8), route(8), resolv.conf(5), hosts(5)

DragonFly BSD/FreeBSD/NetBSD: rc.conf(5)

OpenBSD: hostname.if(5), myname(5), mygate(5), netstart(8), rc.conf(8)

2.6.3 Determine which TCP or UDP ports are open on a system.

Concept:

The BSDA candidate should be able to use the utilities found on BSD systems as well as third-party programs to determine which ports are open on a system and which ports are being seen through a firewall.

Practical:

netstat(1), services(5), fstat(1), and the possibly third-party nmap

DragonFly BSD/FreeBSD/NetBSD: sockstat(1)

2.6.4 Verify the availability of a TCP/IP service.

Concept: The BSDA candidate should be able to determine if a remote system is available via TCP/IP, and if so, telnet(1) to a particular TCP service to determine if it is responding to client requests.

Practical:

ping(8), traceroute(8), telnet(1), netstat(8)

FreeBSD/OpenBSD: nc(1)

2.6.5 Query a DNS server.

Concept: The BSDA candidate should understand basic DNS theory, including types of resource records, types of DNS servers, reverse lookups, and zone transfers. The candidate should be able to query a DNS server for a particular type of resource record, understand which servers are authoritative for a zone and determine if a DNS server is willing to do a zone transfer.

Practical:

host(1), ping(8)

FreeBSD: drill(1)

DragonFly BSD/NetBSD/OpenBSD: dig(1)

DragonFly BSD/OpenBSD: nslookup(1)

NetBSD: nslookup(8)

2.6.6 Determine who is responsible for a DNS zone.

Concept: The BSDA candidate should be able to perform a reverse DNS lookup to determine the network associated with an IP address and then gather contact information regarding that network.

Practical:

whois(1)

DragonFly BSD/FreeBSD: drill(1)

DragonFly BSD/NetBSD/OpenBSD: dig(1)

2.6.7 Convert a subnet mask between dotted decimal, hexadecimal, or CIDR notation.

Concept: The BSDA candidate should be familiar with IPv4 addressing and how to convert a subnet mask from a given notation to another specified notation.

2.6.8 Gather information using an IP address and subnet mask.

Concept: Given an IPv4 address and subnet mask, the BSDA candidate should be able to determine the subnet address, broadcast address and the valid host addresses available on that subnet address.

2.6.9 Understand IPv6 address theory.

Concept: The BSDA candidate should be able to recognize basic IPv6 addressing theory, including: the components of an IPv6 address; the support for multiple addresses (link, local, global) per interface; address and prefix representation (aaaa:bbbb::dddd/17) and the address format (48bit prefix, 16bit subnet, 64 hostbits). In addition, the candidate should understand the autoconfiguration process where the router sends its prefix or gets queried and the host adds its 64 host-bits which are derived from its MAC address. Finally, the candidate should be able to troubleshoot basic IPv6 connectivity.

Practical:

ifconfig(8), ping6(8)

Dragonfly BSD/FreeBSD/NetBSD: rtsol(8)

2.6.10 Demonstrate basic tcpdump skills.

Concept: Given some tcpdump output, the BSDA candidate should be able to answer basic network connectivity questions. This requires the candidate to recognize common TCP and UDP port numbers, the difference between a TCP/IP server and a TCP/IP client, and the TCP three-way handshake.

2.6.11 Manipulate ARP and neighbor discovery caches.

Concept: Candidates should understand basic ARP theory as well as the neighbor discovery cache used on IPv6 networks. Candidates should be able to view, modify, and clear these caches and recognize when it is necessary to do so.

Practical:

arp(8), ndp(8)

2.6.12 Configure a system to use NTP.

Concept: The BSDA candidate should be familiar with the concepts in RFC 868, the importance of synchronizing time on server systems, and which services in particular are time sensitive. The candidate should also be able to configure NTP and manually synchronize with a time server as required.

Practical:

ntpd(8)

Dragonfly BSD/OpenBSD: ntpd.conf(5)

DragonFly BSD/FreeBSD/NetBSD: ntp.conf(5), ntpq(8), rc.conf(5)

OpenBSD: rc.conf(8)

2.6.13 View and renew a DHCP lease.

Concept: The BSDA candidate should have a basic understanding of DHCP leases and how to configure a client to override the settings received from a DHCP server. In addition, the candidate should be able to view the current lease, release it, and renew a lease on each BSD.

Practical:

dhclient(8), dhclient.leases(5), dhclient.conf(5)

DragonFly BSD/FreeBSD/NetBSD: rc.conf(5)

OpenBSD: rc.conf(8)

2.6.14 Recognize when and how to set an interface alias.

Concept: The BSDA candidate should recognize when it is appropriate to set an interface alias and which commands are available on each of the BSDs to set the alias.

Practical:

ifconfig(8)

DragonFly BSD/FreeBSD/NetBSD: rc.conf(5)

NetBSD: ifaliases(5), ifconfig.if(5)

OpenBSD: hostname.if(5), rc.conf(8)

2.6.15 Change the order of name resolution.

Concept: Candidates should recognize the default order of host name resolution on BSD systems and how to change the order of host name resolution.

Practical:

resolv.conf(5)

DragonFly BSD/FreeBSD/NetBSD: nsswitch.conf(5)

2.7 Domain 7: Basic Unix Skills

BSD has its roots in Unix and many Unix utilities were originally developed on BSD systems. The BSDA candidate needs to demonstrate proficiency in the most commonly used Unix command line utilities. The BSDA examination does not test every Unix utility, but only those specifically covered in this BSDA Requirements Document.

2.7.1 Demonstrate proficiency in using redirection, pipes, and tees.

Concept: The BSDA candidate should be able to to redirect either input or output, use a pipe to send the output of one command to another command or file, and use a tee to copy standard input to standard output.

Practical:

<, >, |, tee(1), >& and |&

2.7.2 Recognize, view, and modify environmental variables.

Concept: The BSDA candidate should be able to view and modify environmental variables, both temporarily and permanently, for each of the default shells found on BSD systems.

Practical:

env(1), sh(1), csh(1), tcsh(1), environ(7)

2.7.3 Be familiar with the vi(1) editor.

Concept: The default editor on BSD systems is often vi(1) and many system utilities require familiarity with vi(1) commands. The BSDA candidate should be able to edit files using this editor, as well as modify a read-only file or exit vi(1) without saving any edits to the file.

Practical:

vi(1) including: :w, :wq, :wq!, :q!, dd, y, p, x, i, a, /, :, :r, ZZ, :set number, :set list

2.7.4 Determine if a file is a binary, text, or data file.

Concept: While BSD systems use naming conventions to help determine the type of file, the BSDA candidate should be aware that these are conventions only and that there is a database to help determine file type.

Practical:

file(1), magic(5)

2.7.5 Locate files and binaries on a system.

Concept: The BSDA candidate should be able to quickly find the location of any file on the system as needed and know which utilities can be used to find binaries, source, manpages, and files. In addition, the candidate should be able to update the locate(1) database.

Practical:

whatis(1), whereis(1), which(1), locate(1), find(1), locate.updatedb(8), sh(1) including “type” built-in, -v and -V

NetBSD: locate.conf(5)

2.7.6 Find a file with a given set of attributes.

Concept: The find(1) utility is invaluable when searching for files matching a specific set of attributes. The BSDA candidate should be comfortable in using this utility and may be asked to locate files according to last modification time, size, type, file flags, UID or GID, permissions, or by a text pattern.

Practical:

find(1)

2.7.7 Create a simple Bourne shell script.

Concept: Most system administration tasks can be automated with shell scripts. The BSDA candidate should be aware of the advantages and disadvantages of using a Bourne shell script rather than a csh(1) or bash(1) shell script. The candidate should be able to recognize a shebang, comments, positional parameters and special parameters, wildcards, the proper use of quotes and backslashes, and the use of for, while, if, case, and exec. In addition, the candidate should know how to make a script executable and how to troubleshoot a script.

Practical:

sh(1), stdin(4), stdout(4), stderr(4), chmod(1)

2.7.8 Find appropriate documentation.

Concept: BSD systems are well documented and there are many detailed resources available to the system administrator. The BSDA candidate should know the location of and be able to use the documentation found on the system itself as well as be aware of the resources available at each Project's website.

Practical:

apropos(1), man(1), whatis(1), info(1), mandoc(1)

FreeBSD/NetBSD/OpenBSD: man.conf(5)

2.7.9 Recognize the different sections of the manual.

Concept: The BSDA candidate needs to recognize what type of information is found in each section of the manual. In addition, the candidate should be able to specify a specific section of the manual, ask to see all sections of the manual, and do a search query within the manual.

Practical:

man (1), intro(1) to intro(9), using / (for search)

2.7.10 Verify a file's message digest fingerprint (checksum).

Concept: The BSDA candidate should be familiar with the theory behind a message digest fingerprint and why it is important to verify a file's fingerprint. In addition, candidates should be able to create a fingerprint as well as verify an existing fingerprint.

Practical:

md5(1), openssl(1), sha1(1), cksum(1)

2.7.11 Demonstrate familiarity with the default shell.

Concept: The BSDA candidate should be comfortable using the sh(1), csh(1), or tcsh(1) shells. The candidate should be able to modify shell behavior both temporarily and permanently, including: prevent the shell from clobbering existing files, use history substitution, and set command aliases to save time at the command line. The candidate should also know how to temporarily bypass a command alias.

Practical:

sh(1), csh(1), tcsh(1)

!, !!, $, 0, h, t, r, p, \

2.7.12 Read mail on the local system.

Concept: The BSDA candidate should be aware that by default, system messages may be emailed to the root user on the local system and that a third-party MUA may not be installed. The candidate should be able to both read and send mail using the built-in mail(1) command. The candidate should also know the location of user mailbox files.

Practical:

mail(1), /var/mail/$USER

2.7.13 Use job control.

Concept: The BSDA candidate should know how to start a process in the background, place an existing process into the background, and return a background process to the foreground. The candidate should also be able to verify if any jobs are currently in the background and be aware of the difference between kill(1) and the shell built-in “kill”.

Practical:

these shell built-in commands: &, ^Z, jobs, bg, fg, and kill

2.7.14 Demonstrate proficiency with regular expressions.

Concept: Regular expressions are part of the daily life of a system administrator and the BSDA candidate needs to be able to match text patterns when analyzing program output or searching through files. The candidate should be able to specify a range of characters within [], specify a literal, use a repetition operator, recognize a metacharacter, and create an inverse filter.

Practical:

grep(1), egrep(1), fgrep(1), re_format(7)

2.7.15 Overcome command line length limitations.

Concept: The command line length is limited, and often a command should be applied to more arguments than fit on a command line. The BSDA candidate should understand how to run the command multiple times with different arguments for each call.

Practical:

xargs(1), find(1)

2.7.16 Understand various domain contexts.

Concept: The term "domain" is used in Unix for several facilities. The candidate should understand the meaning of the term in the context of the Network Information System (NIS), the Domain Name System (DNS), Kerberos, and NTLM domains.

Practical:

domainname(1), resolv.conf(5)

DragonFly BSD/FreeBSD/NetBSD: krb5.conf(5)

DragonFly BSD: smb.conf(5)

2.7.17 Configure an action to be scheduled by cron(8).

Concept: The BSDA candidate should understand the difference between the system crontab and user crontabs. In addition, the candidate should be familiar with using the crontab editor, be able to recognize the time fields seen in a crontab, and understand the importance of testing scripts before scheduling their execution through cron(8). Candidates should also recognize that the files /var/cron/allow and /var/cron/deny can be created to control which users can create their own crontabs.

Practical:

crontab(1), cron(8), crontab(5)

3 Conclusion

The BSD Associate examination covers a lot of ground. The candidate is expected to know several concepts and details from all four major BSD variants. While this may place an initial burden on some candidates, the BSD Certification Group believes that having some knowledge of all four BSDs strengthens candidates and places them in the best position to recommend and maintain the right BSD for the job.

By mastering a broad coverage of topics, the candidate absorbs knowledge and information they may not yet have encountered in their use of BSD Unix. The result of engaging these topics for study is two fold: better qualified BSD Unix users and administrators, and more people knowledgeable of the range of BSD choices available. This strengthens the case for BSD Unix deployment worldwide.

4 Appendix A

To assist the testing candidate, Table 2 provides an alphabetized list of the commands found in the "Practical" sections of the exam objectives.

Key:

X	can be installed with operating system and has an associated man page
P	available as a third party package or port

Table 2: Command Reference

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
ac(8)	X	X	X	X
accton(8)	X	X	X	X
acl(3)	X	X		
adduser.conf(5)	X	X		
adduser(8)	X	X	X	X
aliases(5)	X	X	X	X
altq.conf(5)			X	
altq(4)	X	X	X	
altq(9)	X	X	X	
altqd(8)			X	
apropos(1)	X	X	X	X
arp(8)	X	X	X	X
atactl(8)			X	X
bhyve(8)		X		

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
bioctl(8)			X	X
boot(8)	X	X	X	X
boot.conf(8)				X
bsdinstall(8)		X		
bsd.port.mk(5)				X
build(7)	X	X		
builtin(1)	X	X		
bzcat(1)	X	X	X	P (bzip)
camcontrol(8)	X	X		
ccdconfig(8)	X	X	X	
cgd(4)			X	
chflags(1)	X	X	X	X
chfn(1)	X	X	X	X
chgrp(1)	X	X	X	X
chmod(1)	X	X	X	X
chown(8)	X	X	X	X
chpass(1)	X	X	X	X
chroot(8)	X	X	X	X
chsh(1)	X	X	X	X
cksum(1)	X	X	X	X
config(8)	X	X		X
cp(1)	X	X	X	X
cpdup(1)	X	P		
cpio(1)	X	X	X	X
cron(8)	X	X	X	X
crontab(1)	X	X	X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
crontab(5)	X	X	X	X
csh(1)	X	X	X	X
daily.conf(5)			X	
daily(8)				X
dd(1)	X	X	X	X
devinfo(8)	X	X		
df(1)	X	X	X	X
dhclient.conf(5)	X	X	X	X
dhclient.leases(5)	X	X	X	X
dhclient(8)	X	X	X	X
dhcpd.conf(5)	X	P	X	X
dig(1)	X		X	X
dmesg(8)	X	X	X	X
doas(1)				X
doas.conf(5)				X
domainname(1)	X	X	X	X
dports(7)	X			
drill(1)	X	X		P
du(1)	X	X	X	X
dummynet(4)	X	X		
dump(8)	X	X	X	X
egrep(1)	X	X	X	X
env(1)	X	X	X	X
environ(7)	X	X	X	X
etcupdate(8)	X		X	
exports(5)	X	X	X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
extattrctl(8)		X	X	
fgrep(1)	X	X	X	X
file(1)	X	X	X	X
find(1)	X	X	X	X
fsck(8)	X	X	X	X
fstab(5)	X	X	X	X
fstat(1)	X	X	X	X
ftpusers(5)	X	X	X	
gbde(4)		X		
gbde(8)		X		
geli(8)		X		
getextattr(1)		X	X	
getfacl(1)		X		
gettytab(5)	X	X	X	X
gmirror(8)		X		
graid3(8)		X		
grep(1)	X	X	X	X
groups(1)	X	X	X	X
gstat(8)		X		
gstripe(8)		X		
hier(7)	X	X	X	X
host(1)	X	X	X	X
hostname.if(5)				X
hostname(1)	X	X	X	X
hosts(5)	X	X	X	X
httpd.conf(5)	X			X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
id(1)	X	X	X	X
ifaliases(5)			X	
ifconfig(8)	X	X	X	X
ifconfig.if(5)			X	
inetd(8)	X	X	X	X
info(1)	X	X	X	X
init(8)	X	X	X	X
intro(1) to intro(9)	X	X	X	X
iostat(8)	X	X	X	X
ipf(8)		X	X	
ipfstat(8)		X	X	
ipfw(8)	X	X		
jail(8)	X	X		
kauth(9)			X	
kill(1)	X	X	X	X
killall(1)	X	X		
kldload(8)	X	X		
kldstat(8)	X	X		
kldunload(8)	X	X		
krb5.conf(5)	X	X	X	
ksh(1)	X	P	X	X
last(1)	X	X	X	X
lastcomm(1)	X	X	X	X
lastlog(5)	X		X	X
lastlogin(8)	X	X	X	
limit(1)	X	X	X	

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
limits(1)	X	X		
ln(1)	X	X	X	X
loader.conf(5)	X	X		
locate.conf(5)			X	
locate.updatedb(8)	X	X	X	X
locate(1)	X	X	X	X
login.conf(5)	X	X	X	X
lpc(8)	X	X	X	X
lpq(1)	X	X	X	X
lprm(1)	X	X	X	X
ls(1)	X	X	X	X
lsof(8)		P	P	P
mac(4)		X		
magic(5)	X	X	X	X
mail(1)	X	X	X	X
mailer.conf(5)	X	X	X	X
mailq(1)	X	X	X	
make.conf(5)	X	X		
make(1)	X	X	X	X
man.conf(5)		X	X	X
man(1)	X	X	X	X
mandoc(1)	X	X	X	X
master(8) (postfix)	X	P	X	P
md5(1)	X	X	X	X
mergemaster(8)	X	X		
mk.conf(5)			X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
modload(8)			X	
modstat(8)			X	
module(7)			X	
modunload(8)			X	
monthly.conf(5)			X	
monthly(8)				X
motd(5)	X	X	X	X
mount(8)	X	X	X	X
mountd(8)	X	X	X	X
mtree(8)	X	X	X	X
mygate(5)				X
myname(5)				X
named.conf(5)	X		X	
nc(1) (netcat)	P	X	P	X
ndp(8)	X	X	X	X
netstart(8)				X
netstat(1)	X	X	X	X
newaliases(1)	X	X	X	
newaliases(8)				X
newsyslog.conf(5)	X	X	X	X
newsyslog(8)	X	X	X	X
nfsd(8)	X	X	X	X
nfsstat(1)	X	X	X	X
nice(1)	X	X	X	X
nmap(1)	X	P	P	P
nologin(8)	X	X	X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
npf(7)			X	
npfctl(8)			X	
nslookup(1)	X			X
nslookup(8)			X	
nsswitch.conf(5)	X	X	X	
ntp.conf(5)	X	X	X	
ntpd.conf(5)	X	P		X
ntpd(8)	X	X	X	X
ntpq(8)	X	X	X	
openssl(1)	X	X	X	X
packages(7)				X
passwd.conf(5)			X	
passwd(1)	X	X	X	X
passwd(5)	X	X	X	X
patch(1)	X	X	X	X
pax(1)	X	X	X	X
pcictl(8)			X	
pciconf(8)	X	X		
periodic.conf(5)	X	X		
periodic(8)	X	X		
pf(4)	X	X	X	X
pf.conf(5)	X		X	X
pfctl(8)	X	X	X	X
pgrep(1)	X	X	X	X
ping(8)	X	X	X	X
ping6(8)	X	X	X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
pkg_add(1)			X	X
pkg_admin(1)			X	
pkg-audit(8)	X	X		
pkg_delete(1)			X	X
pkg-delete(8)	X	X		
pkg_info(1)			X	X
pkg-info(8)	X	X		
pkg-install(8)	X	X		
pkg-upgrade(8)	X	X		
pkg-version(8)	X	X		
pkgsrc(7)	X		X	
pkill(1)	X	X	X	X
portmap(8)				X
ports(7)		X		X
postalias(1) (postfix)	X	P	X	P
postinstall(8)			X	
postqueue(1) (postfix)	X	P	X	P
printcap(5)	X	X	X	X
ps(1)	X	X	X	X
pstat(8)	X	X	X	X
pw(8)	X	X		
pwd_mkdb(8)	X	X	X	X
raidctl(8)			X	
rc.conf(5)	X	X	X	
rc.conf(8)				X
rc(8)	X	X	X	X

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
rcctl(8)				X
re_format(7)	X	X	X	X
renice(8)	X	X	X	X
resolv.conf(5)	X	X	X	X
restore(8)	X	X	X	X
rm(1)	X	X	X	X
rmuser(8)	X	X		X
route(8)	X	X	X	X
rpc.lockd(8)	X	X	X	X
rpc.statd(8)	X	X	X	X
rpcbind(8)	X	X	X	
rtsol(8)	X	X	X	
sa(8)	X	X	X	X
scsi (8)				X
scsictl(8)			X	
security.conf(5)			X	
security(7)	X	X	X	
security(8)				X
service(8)	X	X	X	
services(5)	X	X	X	X
sh(1)	X	X	X	X
sha1(1)	X	X	X	X
shutdown(8)	X	X	X	X
smb.conf(5)	X	P	P	P
smtpctl(8)	X			X
sockstat(1)	X	X	X	

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
softraid(4)				X
sort(1)	X	X	X	X
ssh-keygen(1)	X	X	X	X
sshd_config(5)	X	X	X	X
sshd(8)	X	X	X	X
stat(1)	X	X	X	X
stderr(4)	X	X	X	X
stdin(4)	X	X	X	X
stdout(4)	X	X	X	X
su(1)	X	X	X	X
sudo(8)	X	P	P	P
sudoers(5) (sudo)	X	P	P	P
swapctl(8)	X	X	X	X
sysctl.conf(5)	X	X	X	X
sysctl(8)	X	X	X	X
sysinst(8)			X	
syslog.conf(5)	X	X	X	X
sysmerge(8)				X
systat(1)	X	X	X	X
systrace(1)				X
tail(1)	X	X	X	X
tar(1)	X	X	X	X
tcpdump(1)	X	X		
tcpdump(8)			X	X
tcplay(8)	X			
tcsh(1)	X	X	P	P

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
tee(1)	X	X	X	X
telnet(1)	X	X	X	X
top(1)	X	X	X	X
traceroute(8)	X	X	X	X
ttys(5)	X	X	X	X
umask(2)	X	X	X	X
umount(8)	X	X	X	X
uptime(1)	X	X	X	X
user(8)			X	X
useradd(8)			X	X
userdel(8)			X	X
userinfo(8)			X	X
usermgmt.conf(5)			X	X
usermod(8)			X	X
users(1)	X	X	X	X
utmp(5)	X		X	X
veriexecctl(8)			X	
vi(1)	X	X	X	X
vinum(8)	X			
vipw(8)	X	X	X	X
visudo(5) (sudo)	X	P	P	P
vmstat(1)			X	
vmstat(8)	X	X		X
vnd(4)			X	X
w(1)	X	X	X	X
weekly.conf(5)			X	

Command	DragonFly BSD	FreeBSD	NetBSD	OpenBSD
weekly(8)				X
whatis(1)	X	X	X	X
whereis(1)	X	X	X	X
which(1)	X	X	X	X
who(1)	X	X	X	X
whoami(1)	X	X	X	X
whois(1)	X	X	X	X
wtmp(5)	X		X	X
xargs(1)	X	X	X	X
xen(4)		X		
Xen			P	
zpool(8)		X	X	
zfs(8)		X	X	
zmore(1)	X	X	X	X