Multi-operative USB HD: An All-In-One Solution to IT Supports and Forensic Experts

Anil Kumar Karna[1,3] and Yuting Chen[2,3]
[1]Department of Computer Science, Shanghai Jiao Tong University, Shanghai-200240, China
[2]School of Software, Shanghai Jiao Tong University, Shanghai-200240, China
[3]Shanghai Key Laboratory of Computer Software Testing & Evaluation, Shanghai-201112, China
Email: anil.karna@sjtu.edu.cn, chenyt@cs.sjtu.edu.cn

Abstract— **An operating system is the main interface for human beings to interact with computer, use its resources and run their favorite applications or do their professional work. Lots of operating systems are available for a personal computer and end users have their own choice to select one or more out of these systems. Having numerous choices for an operating system selection, and having different editions and distributions of each choice, IT supports are bound to have all these distinct copies of various operating systems' installation resource (CDs/DVDs) in order to fulfill users' requirements. Similarly, forensic experts should also carry these different copies of various operating systems, as the destination computer may require any of them while retrieving the data. These professionals, however, have started to use a usb stick as a replacement of CD/DVD. Using a usb stick to install an operating system is also a solution to reduce installation time by multiple factors. But the existing number of distinct copies of these operating systems force them to carry a number of installation media (CDs, DVDs, or usb sticks), or compel them to remake a usb stick as an installation medium every time for a different operating system. This paper addresses an all-in-one solution to the IT professionals in using a usb hard disk to install every popular operating system. This paper first describes how a usb stick can be an installation medium for different editions and distributions of various popular operating systems, such as Windows, GNU/Linux distributions, and Mac OS X, individually. It also explains an all in one solution to IT support and forensic expert professionals, to use a single external usb hard disk as an installation resource for all the aforementioned operating systems.**

Index Terms— **USB Hard Disk, Operating Systems, Portable OS, Live OS, Boot Loader, OS Installation Media.**

I. Introduction

AN operating system (OS) is the main interface for human being to interact with the existing electronic devices, properly use and utilize the available resources, and run their favorite applications. Along with the release of new and upgraded hardware, advanced in recent and latest technologies, and the research & development of hardware and microprocessor, the necessity of developing advanced OSs has also been increased accordingly. OS developing companies are competing on developing better and effective OS interfaces, adding fancy features with advanced desktop solutions, clarity in visual and graphical effects, less buggy yet more professional, reliable, and secure OS. As a result, a number of OS choices are available for end users to select and install. Some users like using a user friendly Windows (Win) OS, some love using a beautiful and reliable Macintosh (Mac) OS, some are keen on using a free GNU/Linux (Linux) distribution, while some are interested in using a different OS.

In addition, a single OS has number of installation disks for different editions, versions, and distributions. Most of these OSs have a different installation disk for 32 bit and 64 bit computers as well. OS like Win OS even has a separate installation disk for each language. When IT supports have to provide their service to the users of foreigners' colony, they have to collect a number of different Win OS installation disks for each different language separately. Language pack for different languages and other related hacks are available for some editions, but none of them are suitable for all the existing Win OS editions.

On the other hand, Linux OS itself has over six hundred different distributions, and over three hundred are still in active development, constantly being upgraded [3]. Out of these various distributions, Fedora [4], Ubuntu [5], Kubuntu [6], and Xubuntu [7] are common choice for Asian and American users. Other distributions, such as Mandriva [8] and Suse [9] are popular in Europe and Brazil. Language packs are not an issue for these distributions, because most of the languages are available in the single installation disk. Furthermore, being an open source, Linux distributions have upgraded versions available to download frequently. Fedora releases its new version in every six months, which forces IT supports to burn or prepare a new installation medium for installation in every six months. This is a real troublesome.

Specifically, with the number of OSs available, users may select to use an OS or various OSs on a single computer. IT supports have to please users by installing OS/OSs of users' choice. Having numerous choices for OS selection, more precisely, having different editions and distributions of a particular OS itself, IT supports are bound to have a collection of all these distinct copies of various OSs' installation CDs/DVDs in order to fulfill users' requirements. Similarly, forensic experts also have to be prepared with the installation disks of all the existing OSs [1], because they may require any of the existing OSs

Manuscript received May 31, 2013; accepted August 24, 2013.
Corresponding author: Yuting Chen

doi:10.4304/jsw.9.4.847-858

while retrieving the information from the destination computer. Additionally, testing and debugging team needs a similar collection as well, so that a developed application can be tested on different OSs' platform.

However, installing OS from CD/DVD is time consuming, unreliable and over sensitive. Even a small scratch on it compels an installer to restart the installation. The physical availability of CD/DVD drive itself is a major issue with the evolution of advanced, fancy and slim notebooks and netbooks. Lots of computers are launched without CD/DVD drive these days. As a result, either usb CD/DVD drive or another installation medium has to be replaced to install OS on such computers.

In order to address these problems, most of the IT specialists have started to use a usb stick to install OS. Using a usb stick certainly has lots of benefits over CD/DVD. Because of its high speed data transfer, installation time is reduced. This is a big relief for an IT support, who has to install a number of OSs everyday. In addition, a usb stick can be used multiple times, and it is also a solution for computers missing CD/DVD drive. However, IT specialists still need to carry numerous usb sticks because of the distinct copies of these various OSs.

In this scenario, it would be reasonable if a single usb HD contains installation resources of all the popularly used OSs. It is a better replacement of a CD/DVD, than a usb stick, to install OSs. It has all the benefits of a usb stick described above and generally its capacity is much more than a usb stick. Therefore, it may contain a number of partitions in which installation resources of numerous OSs can be copied. It would be a great comfort for an IT specialist, if a required OS is chosen from the boot menu of such usb HD and the rest of the installation is then completed conveniently.

A usb HD can be used as portable OS itself by installing OS on it. Furthermore, some partitions of usb HD can be made for personal data, and thus, OS as well as the data become portable. People can just carry the usb HD, boot computer from it and enjoy the same OS and data everywhere. Some of the OSs provide "Live OS", which is able to replace portable OS installation as well.

The motivation of our previous work [2] was to install three OSs (Win OS, Fedora and OpenSolaris) from and to a single usb HD. This paper explains how eighty different editions of Win 7 OS, five different editions of Mac OS X86 (MacX86) for PC, OracleSolaris version 11, and various Linux distributions can be installed from a single usb HD. Table I depicts the list of OSs that currently can be installed from this single external usb HD.

The main contribution of this paper is to use a single usb HD to install all the OSs depicted in Table I. Using the methods, IT specialist is able to choose an OS installation from the boot menu of this usb HD, or directly boot portable OS or Live OS. In addition, some other useful software are copied on a different partition for future installation. Some OSs installation iso file inside some partition are used for virtual machines' testing. Since this usb HD already has a platform to add and remove more OSs installation resources, it can be added with Backtrack [10], a Linux distribution, which is helpful for computer security professionals. Besides this, sometimes, customized and optimized kernels also need to be tested [11]–[13] and those can be added to this usb HD as well.

TABLE I.
LIST OF OSS, CURRENTLY CAN BE INSTALLED FROM THIS USB HD

OSs	Edition	Bit		Language			
		32	64	En	Cn	Kr	Jp
Win 7	Starter	y	**No**	y	y	y	y
	Home Basic	y	y	y	y	y	y
	H. Premium	y	y	y	y	y	y
	Professional	y	y	y	y	y	y
	Ultimate	y	y	y	y	y	y
	Enterprise	y	y	y	y	y	y
	Starter SP1	y	**No**	y	y	y	y
	H. Basic SP1	y	y	y	y	y	y
	H. Prem. SP1	y	y	y	y	y	y
	Profess. SP1	y	y	y	y	y	y
	Ultimate SP1	y	y	y	y	y	y
	Enterpr. SP1	**No**	**No**	**No**	**No**	**No**	**No**
GNU/Linux	Fedora 18	y	y	Same for all			
	Ubuntu 12.10	y	y	Same for all			
	Kubuntu 12.10	y	y	Same for all			
	Xubuntu 12.04	y	y	Same for all			
OracleSolaris	Version 11	Same		Same for all			
Mac OS X	Lepd 10.5.5	Same		Same for all			
	Lepd 10.5.7	Same		Same for all			
	SnLpd 10.6.3	Same		Same for all			
	Lion 10.7.1	Same		Same for all			
	MtLion 10.8.2	**No**	y	Same for all			

The reminder of this paper is organized as follows. Section II explains the methodology of making a usb stick as a substitute of CD/DVD installation medium for all the aforementioned OSs, individually. Section III describes the method of making a single usb HD as an installation resources for all the aforementioned OSs, along with installed portable OS, software and data partitions. Section IV analyzes the sorted out problems while implementing these techniques. Section V discusses the related and future work. Section VI concludes this paper.

II. USB STICK AS AN OS INSTALLATION RESOURCE

This section explains how a usb stick can be used as the installation resource of all the aforementioned OSs. Software such as *Unetbootin* [14] does the same thing, i.e. it makes a bootable usb installation medium for popular OSs mentioned above. This software is available for most of these OSs. Using *Unetbootin* software is one of the best choices available for an IT specialist, now a days. However, this section still explains how a usb stick is made as an installation medium for an individual OS because the concept is useful while merging all these OSs installation resource to a single usb HD in next section.

A. Windows OS installation resource

Making a usb stick as an installation medium of Win OS is not a tough task [15]. The simple extraction of Win OS installation iso file on a usb stick is unable to boot the installation because of the missing boot loader. So, boot loader needs to be installed by using *bootsect* command

which can be extracted from the *boot* folder of the Win OS installation disk. Once boot sector is written on the usb stick by using one of the following command line, it smoothly boots the related Win OS installation.

*bootsect.exe /NT52 *drive_name:**[1] *OR*
*bootsect.exe /NT60 *drive_name:**

Here, the flag option of *bootsect* command, */NT52* and */NT60*, are the alternate for *NTLDR* and *BOOTMGR* boot loader, used by Win XP and Win Vista/7/8 respectively. Additionally, *drive_name* is the drive letter, for e.g. F, G, that current OS use to represent this usb stick. After this, simple extraction of the installation iso file to this usb stick makes it ready to use for Win OS installation.

The procedure described above is to make a normal usb stick as an installation medium of one out of many editions of existing Win OS. There are six editions of Win 7, which with service pack (SP) become twelve editions. These twelve editions, having separate installation resources for 32 bit and 64 bit computers, make twenty-four different editions altogether. However, Win 7 *Starter* edition for 64 bit version is not available, with or without SP. Furthermore, there are no SP available for Win 7 *Enterprise* edition for both 32 & 64 bit version, so only twenty editions of Win 7 are available in IT world.

Win OS has a separate installation disk for a different language. This paper proposes a single Win OS installation resource available in four languages depicted in Table I. Twenty editions of Win 7 with four different languages, altogether make eighty editions. Practically, this paper is proposing eighty different editions of Win 7 installation resources in a complete single installation resource. The eighty editions can further be increased or decreased technically, as per the requirements of the users.

The basic idea for making all these eighty editions of Win 7 in a single resource is easy enough theoretically. First of all, all the independent installation resources for twenty editions of a language is merged into a single image file. The other languages packs are then added one by one sequentially after mounting this main image file. Finally, unmounting with saving completes the image with eighty different required editions of Win 7.

The *first step* is to merge all the independent twenty editions' installation resources into a single resource. Win 7 *Ultimate* edition, as a matter of fact, contains all the other editions namely *Starter*, *Home Basic*, *Home Premium*, *Professional*, and *Ultimate*, except *Enterprise* edition. In reality, the ability to select the different editions of Win 7 during installation lies in a configuration file named *ei.cfg*. This *ei.cfg* file contains in the *sources* directory of the installation resources inside the installation medium. Since the installation medium explained in this paper is a usb stick, which is not a *read only* mode like CD/DVD, it is not difficult to remove or rename the required file. Once it is removed or renamed, all the other Win 7 editions are visible and can be selected during Win OS installation. Users are able to select the different editions of the Win 7, whichever they like to install. In order to merge twenty different editions of Win 7, we have prepared six of its installation iso as depicted in the Table II.

[1]The phrases, letters, words, sentences or paths etc. in between the two asterisk can be different for different computers.

TABLE II.
LIST OF DOWNLOADED WINDOWS 7 EDITIONS

Windows Edition	Bit	
	32	64
Ultimate	y	y
Enterprise	y	y
Ultimate with SP1	y	y
Enterprise with SP1	**No**	**No**

Specifically, the description about all the editions are indexed in the *install.wim* file, inside the *sources* directory of the installation medium. Table III depicts the index numbers of various editions inside the *Ultimate* edition.

TABLE III.
INDEX NUMBER OF DIFFERENT EDITIONS INSIDE ULTIMATE EDITION

Index Number	Win 7 Version
1	Starter
2	Home Basic
3	Home Premium
4	Professional
5	Ultimate

The *install.wim* file in the *sources* directory has the list of all the other editions, if exist, indexed inside (Table III). So, the index numbers of *install.wim* from the other editions of Win 7 installation medium can also be exported to this *install.wim* file. In this way, the merged resource contains all the exported editions of Win 7 OS. This is done by using *imagex* command [16] in the command prompt. For instance, suppose an *Enterprise* edition is to be merged with the *Ultimate* edition. Then the index number 1 (because the *Enterprise* edition has only one edition inside its installation medium) of the *install.wim* file from the *sources* directory of the Enterprise edition installation resource has to be exported to the destination *install.wim*, i.e. *install.wim* file of the *sources* directory of the *Ultimate* edition. The command below does the same.

*imagex /export *EnterpriseEd./Source/install.wim* 1 *MergedEd./Source/install.wim* "Windows 7 Enterprise"*

The command completes the exporting/merging process of installation image. The command also automatically creates a new index number 6 on the destination *install.wim* file, because five editions are already contained in this *install.wim* file of the *Ultimate* edition or this new all in one edition [17]. Similarly, by using the *imagex* command, the proper index numbers of the *install.wim* file from all the other editions of Win 7 installation resources (Table II) are merged to customize the creation of this all in one, single installation resource of Win 7 OS.

We have chosen 64 bit editions of Win 7 to be exported or merged to the 32 bit editions of Win 7. It is because of the contents of 64 bit version, which sometimes, cause some problems while running under 32 bit version. In addition, 64 bit versions are fewer in numbers than 32 bit

versions, so obviously, less time is taken to execute and merge the installation images if 64 bit editions are merged onto 32 bit editions [18]. After merging, all 32 bit and 64 bit versions of all the editions of Win 7 including and excluding available SP, the current *install.wim* file contains twenty different Win 7 editions indexed inside it. The list of these twenty editions with their name, old and new index numbers, are depicted in the Table IV.

TABLE IV.
NEW INDEX NUMER IN ALL_IN_ONE INSTALLATION RESOURCE

Original			New Index Number
ISO Version	**Index No.**	**Edition**	
Ultimate x86	1	Starter	01
	2	Home Basic	02
	3	Home Premium	03
	4	Professional	04
	5	Ultimate	05
Enterprise x86	1	Enterprise	06
Ultimate x86 SP1	1	Starter	07
	2	Home Basic	08
	3	Home Premium	09
	4	Professional	10
	5	Ultimate	11
Ultimate x64	1	Home Basic	12
	2	Home Premium	13
	3	Professional	14
	4	Ultimate	15
Enterprise x64	1	Enterprise	16
Ultimate x64 SP1	1	Home Basic	17
	2	Home Premium	18
	3	Professional	19
	4	Ultimate	20

The *first step* is completed and all the twenty different editions of Win 7 are merged into a single installation image as depicted in Table IV. The present installation resource contains twenty different editions of Win 7 OS, but the setup installation language is default one, i.e. English language currently. If only one language all-in-one installation resource is required, then the present merging is a completed resource to use. This paper also presents merging of the other three languages into this installation resource, which is the next step of this process.

The basic idea of the *second step* is to mount this merged *install.wim* file with the proper index number, add the related language pack to it, then unmount it after saving, for all the twenty index numbers of the *install.wim* file individually, one by one. The following commands are used to accomplish the *second step*. Once an index of image file is mounted using first command line below, the second command line can be repeated with a different language pack one after another to add various languages, before saving it by using the third command line.

*dism /mount-wim /wimfile:*sourcePath\install.wim* /index:*indexNo.* /mountdir:*Path_to_mount(Dest.)**

*dism /image:*Path_to_mount(Dest.)* /add-package /packagepath:*Language_Pack_Source_Path\lp.cab**

*dism /unmount-wim /mountdir:*Dest.* /commit*

However, language packs are different for 32 bit & 64 bit Win 7 editions. They also differs for SP editions. Thus, four different language packs for a single language are needed to include that particular language in this all-in-one Win 7 installation resource. We have prepared all the different editions of three languages packs, i.e. twelve altogether. In order to convenient and automate the whole process (*first step & second step*), we have coded small batch file. This batch file runs for approximately 24/25 hours to completely make the installation resource, which contains eighty editions of Win 7 OS.

Although the current *install.wim* file contains four languages version of twenty different Win 7 editions, the index numbers inside this file are still one to twenty. As a result, if this medium is used for installation, the installation still displays twenty different editions of Win 7 in a single language. The link or pointer to which the other languages option are selected during installation is still missing, for which a *third step* is needed.

As mentioned earlier, the various language packs of different Win 7 editions are added upon users' requirement accordingly. Previously, this all-in-one installation contains Win 7 setup installation in English and Chinese language. While making such installation resource, very simple, novice but effective trick has been implemented.

A 32 bit Chinese version of *Ultimate* edition is taken, all the other editions are added to make it twenty editions in Chinese language (*step one*). Language packs for English is then added to its *install.wim* file (*step two*). For *step three*, the *boot.wim* file from the *sources* directory of English installation resource is copied to the *sources* directory of this merged resource. The *boot.wim* is renamed to *boot_en.wim* while copying so that the original *boot.wim* file is not affected. A new entry is then added to the boot menu *bcd* in *boot* directory so that this new *boot_en.wim* file can be loaded to install OS in English language during installation. This trick, worked perfectly because Chinese language installation resource also contains most of the English language environment. User gets a boot menu as depicted in Fig. 1. Upon opting a language, the installation begins in that language.

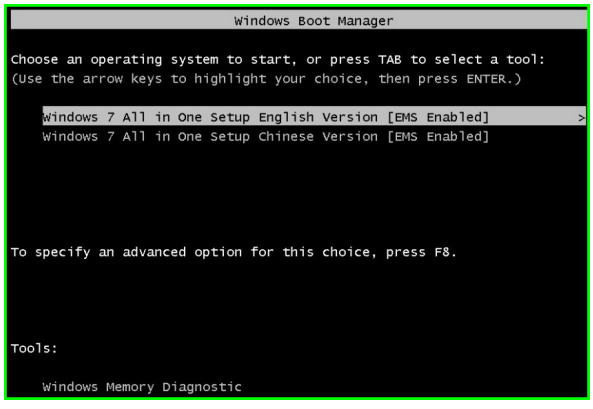

Figure 1. Earlier idea to select language during Win 7 installation

However, the scenario is totally different, when other language except English is added in the similar way. By copying *boot.wim* from another language installation resource cannot boot the installation in that language

because of missing language environment. In addition, it consumes extra space on usb stick as well and it is time consuming and troublesome because *boot.wim* file for all the other languages have to be searched and downloaded. Thus, another idea is used [19] as genuine *third step*.

This *third step* is about polishing the *boot.wim* file. The second index of the *boot.wim* file is mounted using similar commands used in the *second step*, and three language related packages namely *lp.cab*, *winpe-setup_*x*.cab*, and *winpe-setup-client_*x*.cab* from *x* language folder of WinPE Language pack resource, are added to display and select the *x* language during installation. After that, the *lang.ini* file is generated using the command below, before saving and unmounting, to complete the *third step*.

*dism /image:*Path_where_boot.wim_is_mounted* /gen-langini /distribution:*Path_where_boot.wim_is_mounted**

Generally, this idea works for all the other languages, except the East Asian languages like Chinese, Japanese and Korean. Even after adding the related language packs to the *boot.wim* file and selecting these languages as OS language, the installation still continues in English. This is because of the missing font package for East Asian language support [20]. Adding the related *font support package* solves this issue. In short, four files for one East Asian language need to be added. For instance, the following commands are used to add Japanese language packs to the *boot.wim* file of this merged edition.

*dism.exe /mount-wim /wimfile:*g:\sources\boot.wim* /index:2 /mountdir:*g:\mount**
*dism /image:*g:\mount* /add-Package /PackagePath: *g:\Win7\lp_boot\ja-JP\lp.cab**
*dism /image:*g:\mount* /add-Package /PackagePath: *g:\Win7\lp_boot\ja-JP\winpe-setup_ja-jp.cab**
*dism /image:*g:\mount* /add-Package /PackagePath: *g:\Win7\lp_boot\ja-JP\winpe-setup-client_ja-jp.cab**
*dism /image:*g:\mount* /add-Package /PackagePath: *g:\Win7\lp_boot\ja-JP\winpe-fontsupport-ja-jp.cab**
*dism /image:*g:\mount* /gen-langini /distribution: *g:\mount**
*dism /unmount-wim /mountdir:*g:\mount* /commit*

Similarly, all the other related languages are added to the *boot.wim* file. We have coded few lines of batch file to automate the process conveniently. Once this batch file is completely executed, this merged installation resource works as expected and as depicted in Fig. 2.

Basically, making an all-in-one Win OS installation resource has three major steps: the *first step* is merging all the editions to the index of *install.wim* file, the *second step* is adding the related language packs to this merged *install.wim* image, and the *third step* is to add the related language packs to the *boot.wim* image. Once these three major steps are completed, users get proper interface to select their favorite language to be the default language of their Win 7 OS, as depicted in Fig. 2. Likewise, Fig. 3 and Fig. 4 depict the interface to select the different twenty editions of Win 7 installation, when English and Chinese language are selected respectively, as depicted in Fig. 2.

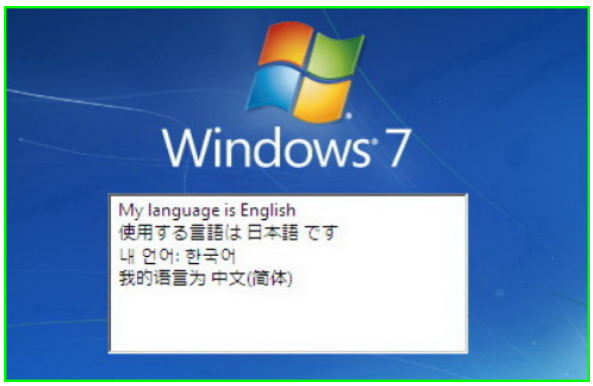

Figure 2. Formal method to select language during Win 7 installation

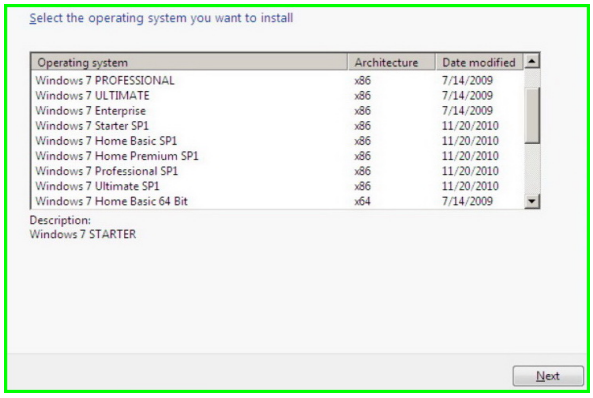

Figure 3. Option to select one of 20 editions of Win 7 OS in English.

Figure 4. Option to select one of 20 editions of Win 7 OS in Chinese.

B. Ubuntu/Kubuntu/Xubuntu OS installation resource

Making usb stick to boot and install Ubuntu, Kubuntu or Xubuntu OS have the similar and easy process. The downloaded iso image from the related sites [21]–[23] is extracted to usb stick using *dd* command as follows:

*dd if=*downloaded.iso* of=/dev/*sdb**

Once the *dd* command copies the iso file into the usb stick, it can be used as the bootable installation resource for these three distributions of Linux OS. Software such as *Ubuntu Live usb creator* [24] also does the same thing.

C. Fedora OS installation resource

Although Fedora is a GNU/Linux distribution, making its usb installation disk is not as easy as Ubuntu, Kubuntu or Xubuntu. The *dd* command copies all the files and folders from installation iso to usb stick, but it does not copy the boot sector. So, this usb is unable to boot the Fedora installation. Software such as *liveusb creator* [25] can make the bootable usb installation disk, however this software is only available after Fedora 10 was released.

The method described in this subsection has favorably been in used before the release of *liveusb creator*. We have used this method to create a usb installation disk for Fedora 5. The same technique is successfully implemented for making the usb installation disk for Fedora 18, the latest release of Fedora OS presently. The basic idea of this method is to install grub boot loader on the usb stick, and then copy some of the required files such as *linux* kernel and *initrd* image from the installation resource so that the boot loader can load the kernel and installation image once computer is booted from it. The installation iso file, likewise need to be copied to it so that the source files of installation can be chosen during installation.

This subsection first describes the making of a bootable usb installation disk for Fedora 7. The easiest way to install grub boot loader is by using the *grub-install* command. Unfortunately, this command cannot guarantee the proper installation of the boot loader on the usb stick, so *grub* command is used instead. First of all, the file system type of the usb stick is checked with the help of *fdisk* command on an already installed Fedora OS. Users may use similar command on other OSs as well.

Generally, after formatting, a usb stick is given FAT16 file system type with partition id "0x6". This partition id creates problem while installing grub on it, or booting this partition with grub boot loader. Hence, the usb stick is converted to FAT32 file system with partition id "0xC". Some files *stage1*, *stage2*, *fat_stage1_5*, *splash.xpm.gz* and *grub.conf* from the *grub* folder inside *boot* directory of the currently installed Fedora system (the files can be downloaded as well) are copied to the *grub* folder inside the *boot* directory of the usb stick. The following command is then used to install grub on it [26].

*grub>root (*hd1*,0)*
*grub>setup (*hd1*)*

The disk counting starts from 0, so *hd0* represents the HD of the installed Fedora OS and *hd1* represents usb stick, assuming no other disks are connected to the system during execution of these commands. After successful execution of these commands, the usb stick has grub boot loader on it and thus it is bootable. Then, the kernel file *vmlinuz* and the image file *inirtd.img* from the *isolinux* folder of the iso installation file, together with installation iso file itself, are copied to the usb stick. Modifying the *grub.conf* file inside the *boot/grub* folder of usb stick, as follows, boots the required Fedora 7 setup installation.

title "Fedora 7 Setup Installation"
root (hd0,0)
kernel /vmlinuz
initrd /initrd.img

The path given for the kernel file and the booting image file for installation, are already copied to the usb root directory earlier. So, the grub configuration file loads the proper files, and so the installation process starts smoothly. During installation, users need to provide the iso source file location, from where Fedora is going to be installed. At this point, selecting hard disk and providing proper device, */dev/sdb1* in our case, continues the rest of the installation process. However, this method only works until the release of Fedora 9. Later releases of Fedora edition start to search another file named *install.img* inside the installation medium, i.e. usb stick, during installation.

The solution to make a bootable Fedora 10 usb installation disk is not very tough. By copying the required file to the required directory of usb stick sorts out this issue. The file *install.img* needs to be copied from the *images* directory of installation iso image to the *images* directory of the usb stick. Rest of the settings are similar to the earlier editions of Fedora OS. During installation, user needs to provide the proper device, */dev/sdb1* in our case, as installation source location and proper directory location (blank in our case as *install.img* file is copied to the *images* directory of usb stick) to load the *install.img* file. The installation iteself searches *install.img* file in the *images* directory of the usb stick and continues the rest of the installation process. However, this method also starts giving troubles after the release of Fedora 15.

Slight changes have been made on Fedora 15. Even installation iso file is not required on usb stick to install it for the computers connected to internet. Just copying *vmlinuz* and *initrd.img* to the bootable usb stick are enough. The same *grub.conf* file coded above boots and installs it from usb stick in the absence of installation iso file on the usb stick. It downloads and installs the packages from the internet. With the grub configuration coded above, even if installation iso file is present, the installation still uses internet to install the OS. Since this paper describes making a bootable Fedora 15 installation usb stick, some changes are made on grub configuration file to forcefully install the OS locally from usb stick. For that, *repository* is provided on the grub configuration file while loading installation kernel by using *repo* flag option. The modified grub configuration is as follows:

title "Fedora 15 Setup Installation"
root (hd0,0)
*kernel /vmlinuz repo=hd:/dev/*sdb1*:/*
initrd /initrd.img

However, loading installation kernel without providing the *repo* option only works for Fedora 15 and Fedora 16 installation. With the release of Fedora 17, this *repo* flag is mandatory. If the *repo* flag is not provided while loading the setup installation, boot loader is not able to load the proper installation of Fedora 17 and Fedora 18 OS.

Fedora 18, on the other hand, starts to create a new, totally unexpected problem. The file size of the installa-

tion iso for the Fedora 18 is more than 4GB and FAT file system does not support any file which is bigger than 4GB. So the installation iso image cannot be copied to this FAT file system formatted usb stick. Furthermore, grub legacy boot loader[2] is unable to mount NTFS partition, thus the usb stick cannot be formatted as NTFS file system as well. Linux file system supports these types of bigger files, hence the usb stick is formatted as ext2 file system. Accordingly, *stage1*, *stage2*, *e2fs_stage1_5*, *splash.xpm.gz* and *grub.conf* files from */boot/grub* directory of already installed Fedora system, Fedora 15 in our case, are copied to the *boot/grub* directory of this usb stick.

The grub boot loader is again installed as the file system of the usb stick is changed to ext2. The *e2fs_stage1_5* file exists in the boot folder of the usb stick, so the *grub* command, used earlier in this subsection, can successfully install grub boot loader. Rest of the Fedora 18 installation process is same as Fedora 17. Providing *repo* flag option installs the system. Table V lists the files that are required to be extracted from the iso file and copied to the usb stick to install the different editions of Fedora OS locally. This copying process is done after the grub is installed on the usb stick so that the *boot* folder in it is not affected.

TABLE V.
FILES INSIDE A USB STICK TO INSTALL VARIOUS FEDORA EDITIONS

Edition	Files to copy from iso	Files in usb, besides /boot
7	isolinux/vmlinuz isolinux/initrd.img	vmlinuz initrd.img Fedora_version.iso
8		
9		
10	isolinux/vmlinuz isolinux/initrd.img images/install.img	vmlinuz initrd.img images/install.img Fedora_version.iso
11		
12		
13		
14		
15	isolinux/vmlinuz isolinux/initrd.img	vmlinuz initrd.img Fedora_version.iso provide repo=hd:/dev/sdb1:/
16		
17		
18		

D. Solaris OS installation resource

Installing OpenSolaris from usb stick has various methods [27], [28]. The method used in our previous paper [2] was to use Solaris terminal from already installed Solaris system or its installation CD/DVD. Running the command *pkg install SUNWdistro-const* on such terminal installs the *usbgen* and *usbcopy* commands. By using *usbgen* command, usb installation image file is generated from the installation iso image. Then by using *usgcopy* command, the image file is copied to a usb stick to make it OpenSolaris installation disk, as shown below.

*usbgen *installation_image.iso* *usbimage.img**
*usbcopy *usbimage.img**

The usb stick is not bootable when *dd* command is used to copy the generated image file (*usbimage.img*) to the usb stick. The *usgcopy* command must be used for copying. The command overwrites the entire usb stick. Once its execution is completed, the usb stick is ready to use as OpenSolaris installation disk. Although *dd* command does not work perfectly while copying generated image to usb stick, it can be used to copy this usb stick to another usb stick, so that another usb stick becomes bootable OpenSolaris installation disk. The later releases of OpenSolaris, i.e. OracleSolaris, has direct download link of usb installation images [29]. Thus, its bootable usb installation disk is easily created by copying the downloaded usb image using *usgcopy* command.

E. Mac OS X for PC (MacX86) installation resource

Ever since Apple started to ship its computers with intel chip, lots of end users show their interest on trying the Mac OS X system on their PCs. Some of the groups have already released some versions of Mac OS X86, that can be installed on PC after some certain troublesome and slight modification on BIOS of PCs [30]–[33]. Some groups release their hacks, soon after the official Mac OS X is released by Apple. We have also tried and successfully installed the Tiger (10.4.x), Leopard (10.5.x), Snow Leopard(10.6.x), Lion (10.7.x), and latest Mountain Lion (10.8.x) editions of MacX86 on different PCs.

Making bootable MacX86 installation usb stick is not a difficult task. Downloading the proper iso image or dmg file, which can be installed on a particular system, is a tough task. However, this is not the related research area this work is focused on. This paper assumes IT specialists or users already have proper installation iso or dmg file of the related MacX86 OS, and thus by using *disk utility* application in an already installed Mac OS, the installation image of it is easily restored to the usb stick.

Unfortunately, this usb stick faces the same problem, which Fedora installation usb stick has faced initially, i.e. missing boot loader. Hence, some of the existing boot loader for MacX86, such as Chameleon [34] is downloaded to install on this usb stick. The files also contain the instructions of installing the boot loader as well as the related boot files. Assuming only one hard disk is present and a single usb stick is connected to the Mac OS X, and also assuming that the user is in the extracted directory of this boot loader, the following commands on terminal successfully make this usb stick as a bootable medium for a MacX86 OS setup installation.

sudo fdisk -f boot0 -u -y /dev/rdisk1
sudo dd if=boot1h of=/dev/rdisk1s1
*sudo cp boot /Volume/*Mount-point-of-usb*/*

Once, the boot loader is installed, the usb stick is able to load the MacX86 installation. This usb stick is able to install MacX86 on some PCs, not all. Some packages for a boot loader installer are also available [35] in GUI, which directly installs the MacX86 boot loader on the usb stick after installation iso or dmg file is restored on it.

III. ALL OSS IN A SINGLE USB HD

A usb HD can be used as portable OS itself by installing an OS to it. This helps people to enjoy the

[2]Although the grub legacy boot loader is updated sequentially after the release of Fedora 15, this paper still uses grub legacy instead of grub2, as grub legacy, compared to grub2, is simple and easy to use.

same OS everywhere, like a laptop. It is not a complete substitute of a laptop, but somehow for some obvious cases, it may be a substitute. We have installed Fedora OS on this usb HD as a portable OS. Out of this experiment, we have found that no matter whether the HD is connected directly or via usb, if the transfer rate and disk RPM is better, the OS works as it is installed on an internal HD.

The basic idea to make a usb HD to install various OSs from it, is to copy the installation resources of the OSs to the different partitions of this usb HD. As per the experiments done while making a single usb HD as an installation resource for several OSs, we found that Win OS, MacX86 OS, Solaris OS can only be installed when the related installation resources are in a primary partitions of a usb HD. Some other Linux distributions, such as Fedora, Ubuntu, Kubuntu and Xubuntu search the required files on any partition that is readable to the boot loader. Thus, they can be extracted to any boot loader readable primary or logical partitions of a usb HD.

Practically, a hard disk can only have four primary partitions when partitioned with master boot record (MBR) scheme. In this scheme, one of these primary partitions is extended to contain lots of other logical partitions. On the other hand, GUID Partition Table (GPT) scheme can have 128 primary partitions at maximum [36], and is already available to use. This paper uses a usb HD that is partitioned with MBR scheme. So it only has four primary partitions. Out of these four partitions, only three partitions are used as primary partition.

As mentioned above, a MacX86 installation resource must reside in a primary partition. We have five distinct editions of MacX86, and restoring each of them to a primary partition of a usb HD is not possible using MBR scheme. However, if an edition of the MacX86 installation resource resides in a primary partition and the remaining MacX86 editions are in the logical partitions of the same usb HD, the remaining MacX86 editions installation resources are easily booted for MacX86 OS installation. The boot loader of MacX86 installation resources in the primary partition recognizes the other MacX86 boot loaders in the logical partitions. When all the MacX86 OS installation resource reside in logical partitions, none can be booted. So at least one MacX86 installation resource is to be restored on a primary partition of this usb HD.

Thus, the three primary partitions of the usb HD need to be allocated for the OSs namely Win OS, one of the MacX86 editions, and a Solaris edition. Besides the OSs installation resources, Fedora is to be installed on this usb HD. In addition, couple of more partitions are needed for other general purpose software and other related personal files with iso installation files. Hence, the disk is partitioned to the four primary partitions, where the fourth partition is the extended partition that contains other logical partitions, as depicted in the Fig. 5.

Figure 5. Disk partition of this all OSs installation in one usb HD

The partition type, id and purpose of the usb HD partitions are listed in Table VI. As depicted in Fig. 5 and Table VI, the first partition is NTFS partition, which contains the all-in-one Win 7 installation resource. This partition contains all the eighty editions of Win 7 that is explained in last section. The partition is made bootable using *bootsect* command, also described in last section.

TABLE VI.
ALL OSS INSTALLATION IN ONE EXTERNAL USB HD'S PARTITIONS

Partition	No.	File System		Purpose
		Type	ID	
Primary	01	NTFS	07	Win 7 installation resource
	02	Solaris	BF	OracleSolaris installation resource
	03	HFS+	AF	Mac 10.8.2 installation resource
Extended fourth primary partition that contains other logical partitions.	05	ext4	83	Boot part of portable Fedora OS
	06	L swap	82	Swap part of portable Fedora OS
	07	ext4	83	Root part of portable Fedora OS
	08	FAT32	0B	GNU/Linux installation resources
	09	NTFS	07	OSs' related installation iso
	10	FAT32	0B	Other software for all OSs
	11	HFS+	AF	Mac 10.7.1 installation resource
	12	HFS+	AF	Mac 10.6.3 installation resource
	13	HFS+	AF	Mac 10.5.7 installation resource
	14	HFS+	AF	Mac 10.5.5 installation resource

Although OpenSolaris is a Linux distribution, it uses ZFS as its file system. The installation disk of OpenSolaris OS is a UFS file system. Its installation resources cannot be simply extracted to any folder of any partition for booting purpose, which is generally done for installing other Linux distributions. Upon doing so for this OS, the installation shows errors on booting, because it is not able to find the resources that contains inside the UFS file system partition. Thus, the installation resources of OpenSolaris need to be on primary partition, so the second primary partition is allotted for it.

The *usbcopy* command, which is used for copying generated image to usb stick, cannot be used to copy the generated image to the usb HD. The command copies files to the entire disk, it is not for copying to a partition of the usb HD. Using this command overwrites the entire usb HD, losing all the other partitions. Hence, another trick is used here. A usb installation stick of the OpenSolaris OS is first made by using *usbcopy* command, and then the first partition of this usb stick is copied to the allotted second partition of the usb HD by using *dd* command. The usb stick is bootable, but the second partition of usb HD is still unbootable due to the missing boot loader on it. Since the grub boot loader used by OpenSolaris system is different from that of other Linux distributions mentioned in this paper, we have installed OpenSolaris compatible grub boot loader on this partition by using the command below on the terminal of an already installed OpenSolaris OS. Terminal of Live OS can be used as well.

*installgrub *path*/stage1 *path*/stage2 /dev/rdsk/c*s0*

The command installs an OpenSolaris supported grub boot loader on the first primary partition which it finds is a Solaris file type with partition id BF, second primary partition in our case. Now the OpenSolaris can be booted if proper command is given. The *mount* command can

be used to know the value of "*" in "c*s0" of the above command. This method works well for OpenSolaris last edition, but sometimes creates problem fot booting OracleSolaris version 11 installation because of the configuration problems in its boot menu. Even when grub is installed using older editions, i.e. OpenSolaris, the OracleSolaris installation is not able to boot the installation properly. However, OracleSolaris supported grub is already installed on this partition, so the installation can be loaded manually using following commands on sequence on its grub text mode.

root (hd0,1,a)
kernel$ /platform/i86pc/kernel/$ISADIR/unix
module$ /platform/i86pc/$ISADIR/boot_archive
boot

It boots the OracleSolaris installation, however, it is troublesome to type these commands every time. Since OracleSolaris starts its Live OS when installation is booted, we have installed grub again using the *installgrub* command on the terminal of this Live OS. This reinstalls the boot loader and boot menu is also configured properly.

On the other hand, the dmg file of the Mountain Lion is restored to the third primary partition of this HD, and then boot loader package for Mountain Lion is installed on this partition. Basically, all these three primary partitions have the installation resources and proper boot loader to load the particular partition now. However, the proper configuration that makes hard disk boot the related installation from the respected partition is still missing.

Nevertheless, all the releases of MacX86 are not compatible for each PC, so we have kept most of the MacX86 editions on this usb HD. We have restored all the earlier editions of MacX86 installation resources on the last few partitions of the partition table of this usb HD, as depicted in Fig. 5. It is planned that, once a new edition of MacX86 is available, another partition is created before the first logical partition of the MacX86 installation resource on this usb HD. All the MacX86 logical partitions are restored with the related iso or dmg file, and the related boot loader is installed on each partition individually.

All the other Linux distributions are loadable from a logical partition if the boot loader can mount the file system type. Any folder of any partition, which is mounted by the boot loader, is used as an installation medium for different versions of Linux distributions [37]. As depicted in Table VI, the first three logical partitions are allotted for portable Fedora OS installation destination. The next logical partition is the FAT partition, and the related files to install Fedora, Ubuntu, Xubuntu, Kubuntu are copied or extracted in the respected folders of this partition. The FAT file system is used because legacy grub boot loader is unable to mount a NTFS partition on boot. So it fails to load the related kernel and image file to boot these Linux distributions from a NTFS partition. This concept works perfect until Fedora 18 is released. The installation iso file of Fedora 18, as described in previous section, is bigger than 4GB. It cannot be copied to its allotted FAT partition, hence it is copied to the root partition of portable OS (Fedora). Root partition of portable OS uses ext2˜4 file system, which can easily be accessed by grub.

Making bootable usb stick as an installation medium for OSs such as Ubuntu, Kubuntu & Xubuntu is easier than that of Fedora OS. However, if the same idea is implemented, it mounts and copies the installation iso of related OSs to the entire usb HD. As mentioned earlier, for Linux distribution installation resources, no extra partitions are needed. The resource files can be extracted to any folder. The boot loader is able load the related kernel and image, and continues the installation process without asking for the installation iso file. However, more information or flag options must be provided during kernel loading. For instance, if a 32 bit installation iso of Ubuntu12.10 is extracted to *32b/ubuntu12.10* directory, say *path*, of its allotted partition, the following grub configuration boots and continues the rest of its installation.

title "Ubuntu 12.10 32 Bit Installation"
root (hd0,7)
kernel /(32b/ubuntu12.10)-as-path*/casper/vmlinuz file=/*path*/preseed/ubuntu.seed boot=casper only-ubiquity iso-scan/filename=/*path*/ quiet splash –*
*initrd /*path*/casper/initrd.lz*

Besides this FAT partition allotted for these GNU/Linux distributions, couple of more partitions are created. Since different OSs use different file systems, another FAT partition is created as a common buffer space to save and store software used by the all OSs. Another partition is made for backup data, and to store the installation iso files of various OSs. Such iso files can be bigger than 4GB size limit of the FAT file system, hence NTFS partition is alloted for them, as depicted in Fig. 5 and Table VI.

At this stage, all the installation resources for the aforementioned OSs reside inside the different partitions of this usb HD. But the appropriate loader to load the proper OS installation and the proper configuration of boot menu are still missing. Finally, Fedora OS is installed on this usb HD and the grub boot loader is installed during its installation. By editing the boot configuration *grub.conf* file of this Fedora OS, the various editions of Fedora OS installation are loaded from the various folders of their residing partition. Similarly, by editing the same *grub.conf* file, and providing more kernel flag options, the installation for other Linux distributions, such as Ubuntu, Kubuntu, and Xubuntu, are loaded as well. The installation resource and their boot loader reside inside their containing partitions for all the other OSs, such as Win OS, OracleSolaris OS & Mac OSs. Hence, chain loading the partition boots the respected OSs installation. Chain loading the partition is done by adding the following commands on the same *grub.conf* file.

root (hd0,0)
chinloader +1

The command lines above loads the Win OS installation from this usb HD, because the Win OS installation resource is contained in the first partition (0) of the first hd (hd0) when this usb HD is booted. After the *grub.conf*

file is added with the related command lines to load all the OSs installation resources residing inside this usb HD, a boot menu is displayed when computer is booted from this usb HD. The boot menu is depicted in Fig. 6.

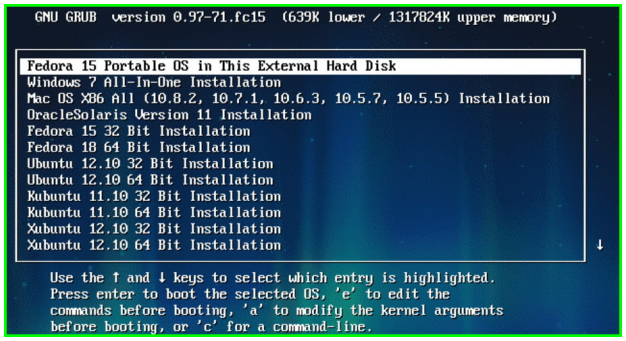

Figure 6. Displayed boot menu on booting a PC with this usb HD

This boot menu contains the portable OS as default OS loader and also the installation resource of all the other aforementioned OSs. If Win OS installation option is selected, the installation of eighty different editions of Win OS can be selected for further installation, which is earlier explained and depicted in Fig. 2. Then after selecting one of the four given languages, users can select their favorite edition of Win 7 OS. Similarly, if the MacX86 installation is selected, the installation of various MacX86 appears, which is depicted in Fig. 7.

Figure 7. Displayed MacX86 boot menu on selecting MacOS setup

Fig. 7 is a partial snapshot of the Mac OS installation boot menu in this HD. Users can select the other MacX86 installation options by moving the right/letf key. In a similar way, if OracleSolaris installation option is selected, it is redirected to the installation boot menu of OracleSolaris OS itself, which is depicted in Fig. 8.

Once Oraclesolaris is selected from its own menu as depicted in Fig. 8, its Live OS is started. Users can enjoy using this Live OS or can start the installation of the OracleSolaris OS anytime from this Live OS. The rest of the OSs, such as Fedora, Ubuntu, Kubuntu and Xubuntu once opted from the main menu (Fig. 6), start to load the installation directly. In addition, Ubuntu, Kubuntu, and Xubuntu also start their Live OS edition, if installation is quit in middle, and user can start installation of respected OSs any time while enjoying the related Live OSs.

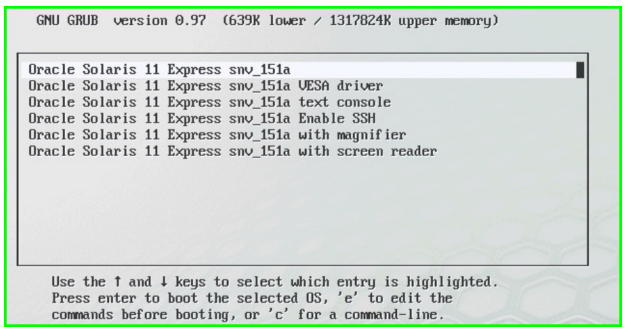

Figure 8. Displayed OracleSolaris grub on selecting Solaris setup

IV. Existing Problems, Solutions & Discussion

The basic concept of this paper is to copy the different OSs installation resources onto the different partitions of the same external usb HD and then patching the proper boot loader of the related OS to their respected partitions. Finally, it uses a boot configuration file to boot the related partition to start the particular OS installation of users' choice. However, practical implementation is not an easy task as described in the related sections this paper.

First of all, searching the proper iso or dmg installation file of MacX86 is time consuming. Similarly, finding the proper boot loaders for various MacX86 editions are troublesome. Boot loader which works for one edition of a MacX86, may not work for a different edition. Moreover, a boot loader able to load the installation process on one computer, may not be able to load the installation process on a different computer, because of the various configurations, hardware and BIOS setup differences on the different computers. The root cause and its probable solution discussion is beyond the scope of this paper.

On the other hand, Fedora OS changes its installation option from time to time, and hence we need to try most of the editions in order to check their successful installation using usb stick. Furthermore, installation of Ubuntu, Kubuntu & Xubuntu cannot be booted properly if fewer kernel flag options are provided. After various tries, the boot menu configuration for these OSs are checked in their original installation iso image, and then the related kernel flags are added to the main boot menu configuration of this external usb HD to make it load the proper installation from this usb HD.

Installing related grub boot loader for Solaris OSs (OpenSolaris & OracleSolaris) is different from other Linux distributions. Solaris uses its own UFS file system type as a source installation partition. So, it cannot be extracted together with other Linux distribution. By copying the usb installation disk to the installation partition of usb HD, then installing boot loader using *installgrub* command with proper device name and partition slice somehow solves this problem. However, making it boot from a partition of usb HD has certain troubles.

Partitioning a usb HD is not a big issue as various disk partitioning software are available to download [38]–[42]. However, partitioning it with suitable size for installation resources of all the aforementioned OSs needs proper

calculation. We have partitioned this usb HD several times to make the proper partition in order with suitable size. However, with the increase in number of Linux distributions, their allotted FAT file system partition has been increased many times. Our experimented usb HD is only 250GB, in which all the other normal application software for all the different OSs, portable OS, OS installation resources and its iso files are copied. At present, the capacity of this usb HD seems to be small to keep all the installation resources of Linux distributions. Thus, we have moved some of the older editions to another partition or to some other HD, and decided to keep fewer editions of similar Linux distributions (Fedora, Ubuntu, Kubuntu, & Xubuntu) on their allotted partition. Our final partition table framework is depicted in Fig. 5 and Table VI.

Besides this, a usb HD cannot be in a *read only* mode. Therefore, the OS installation files residing under the different partitions of the disk can be easily altered to make it non workable, which sometimes may create a disaster. This external usb HD is supposed to be used by the IT professionals, but a normal user can create some unexpected problems while using this bootable all OSs in one usb HD. The scenario can be more crucial when a user by mistake, formats or deletes the partitions of this usb HD, instead of formatting or deleting the internal HD partition of a computer during his/her OS installation.

V. Related and Future Work

Researchers and IT professionals have done numerous works related to the OS installation, from and to the usb stick. Some of the methods are described in our earlier paper [2]. This paper work describes how popular aforementioned OSs' setup installations are merged to make all OSs installation in a single bootable usb HD.

Number of researchers have proposed their method to make a usb stick as an installation medium for a single OS installation medium. Number of software are available in IT world that can make a usb stick as a bootable OS installation disk. All these works have been done to install a single OS from a usb stick though. Very few papers have been published in making all OSs in a single bootable medium. Most of research solutions related to this paper, are found in IT forums, blogs or IT companies' bug discussion sites, but their solutions are mostly for merging similar editions of same OS. We have used these ideas of single OS and elaborated to use it as multipurpose, multi-operative, all OSs installation in one usb HD. This paper proposes a usb HD, which can be treated as an installation medium for most of the popular OSs, along with portable OS itself, a full fledged solution to IT specialists.

At present, MBR partitioning scheme is used to partition this usb HD. The maximum size of a hard disk and the number of primary partition is the main limitation for this scheme. Future work can be done in this area by partitioning it with GPT scheme, in which the number of primary partition is limited to 128 at maximum. As a result, beside Win 7 all in one installation resource, Win XP, Win Vista, Win 8, and its future releases, along with other OSs, such as several editions of MacX86, Solaris can be extracted or restored to the different primary partitions of the usb HD. Key idea for making a bootable OS installation partition is to install suitable boot loader and to copy the proper OS installation related files.

The installation iso file, bigger than 4GB, can neither be copied to a FAT file system partition because of the file size limit, nor be copied to a NTFS partition because the grub boot loader is unable to mount it during boot. Recent advances in computer technology may increase the upgraded version of the popular OS installation iso file in future. Hence, working out with a boot loader can be another future work related to this paper work.

Because of the easy set up configuration, this paper uses legacy grub boot loader, although grub2 is already available on the later releases of Fedora OS. With the release of Fedora 16, the legacy boot loader starts to use sequential upgraded version. Some of the good improvements have been implemented on grub2, such as booting iso image directly from hard disk, dynamic module loading, scripting support including conditional statements and functions, rescue mode, custom menus, themes, graphical boot menu support, non-x86 platform support, universal support for UUIDs [43]. More related information regarding grub2 can be found on its official site [44] and it is always good to start with an available tutorial [45]. However, being familiar with the legacy grub configurations and its command line options, this paper still uses legacy grub, which can be improved in future by direct loading the installation iso file on booting grub2.

Grub2 boot loader is a better option for a GPT scheme partitioned usb HD as well. Most of the current OSs support GPT. Some OSs, including Win OS and Mac OS, only support booting to GPT partitions on systems with EFI firmware [36] though. Thus, this paper still uses MBR partition scheme, as GPT partition scheme can, not necessarily, be bootable by all the systems.

Another main concern and improvement is in the direction of partition locking of a usb HD. During installation people may format or delete the partitions of this all OSs installation in one usb HD by mistake. This will make some of the installation of the OS or OSs missing from the boot menu of this usb HD, and it causes the whole process be repeated again. Although recovery software [46], such as Handy Recovery [47] and Easeus [39], are available to recover the data from the deleted or formatted partitions, these software suggest to recover the data to another partition to avoid data being overwritten. With the number of lost partitions and the data inside these partitions, recovering process is of certain troublesome. Files and folders are still on the deleted or formatted partitions, so it would be better if the related file table, for e.g. Master File Table (MFT) for NTFS file system, can be recovered or recreated rather than recovering each file and folder of each partition to another location and repeating the major steps. Some software, such as Testdisk [48] and EaseUS [39], recreate MFT if MFT mirror record is available. However, in the absence of MFT mirror, they

are unable to repair or recreate it as well. This can be another research direction towards the data recovery.

Furthermore, most of the methods described in this paper are done by manual coding. We have coded few lines of batch file to automate the merging of eighty editions of Win 7 OS. Our future work includes automating the command line code to create the boot loader configuration file for all the aforementioned OSs. The OSs may increase in future and they may use a different file system structure or may create their own file system. Hence, it would be a nice research area to develop a common file system or a simple common method so that the installation of any OS can be started from boot loader itself.

VI. Conclusion

This paper presents the methods and processes on using a single usb HD to install different copies of various OSs such as various editions of Win OS with different languages set up options, for 32 bit and 64 bit computers, including and excluding service pack option, various editions of Mac OS X86 OS, different editions of various distributions of GNU/Linux such as OpenSolaris, Fedora, Ubunutu, Kubuntu and Xubuntu. Besides this, the usb HD itself is used as a portable OS by installing an OS on it, and by booting it on other computers.

Additionally, some common used software, open source software are contained in the different partition of the same usb HD, which is useful to be installed together with the installation of OS. The all in one solution will also be very helpful for IT professionals who have to install different OS frequently to fulfil users requirement. The usb HD saves time wasted over OS installation process or making installation medium for new OS frequently. The methods in this paper can also be implemented for new OSs and for new edition of existing OSs as well.

Acknowledgment

The authors are grateful to the anonymous referees for their valuable comments and suggestions to improve the presentation of this paper. The authors are also thankful to the overseas students at Shanghai Jiao Tong University. Because of their frequent different requirement requests, authors are motivated to build this single USB hard disk as installation medium of all the popular operating systems. This work was supported by National Natural Science Foundation of China (NSFC) (Grant No. 91118004, 61272102, 61100051) and Shanghai Key Laboratory of Computer Software Testing & Evaluating (Grant No. SSTL2011_02).

References

[1] Michael G. Solomon, K Rudolph, Ed Tittel, Neil Broom, and Diane Barrett, "Computer Forensics JumpStart", *Wiley Publishing, Inc., Indianapolis, Indiana*, 2011, pp. 30-31.
[2] Anil Kumar Karna, "Multipurpose usb Hard Disk: Your Mini Laptop", *In proceedings of ITCS'10*, IEEE Computer Society, Washington, DC, USA, 2010, pp. 357-360.
[3] http://en.wikipedia.org/wiki/Linux_distribution
[4] http://www.fedoraproject.org/
[5] http://www.ubuntu.com/
[6] http://www.kubuntu.org/
[7] http://xubuntu.org/
[8] http://www.mandriva.com/
[9] http://www.suse.com/
[10] http://www.backtrack-linux.org/
[11] Wang S., Ji Y., and Yang S., "A Micro-Kernel Test Engine for Automatic Test System", *JCP 6 (1)*, pp. 3-10, 2011.
[12] Ju J., Wu J., Fu J., Lin Z., and Zhang J., "A Survey on Cloud Storage", *JCP 6 (5)*, pp. 1764-1771, 2011.
[13] Li C., and Yue D, "A Model to Select System Core and Its Application", *JCP 6 (9)*, pp. 1913-1919, 2011.
[14] http://unetbootin.sourceforge.net/
[15] http://www.intowindows.com/how-to-install-windows-7vista-from-usb-drive-detailed-100-working-guide/
[16] http://technet.microsoft.com/en-us/library/cc749447(v=ws.10).aspx
[17] http://en.wikipedia.org/wiki/Windows_Imaging_Format
[18] http://www.askvg.com/how-to-create-a-single-dvd-for-windows-vista-x86-and-x84-editions/
[19] http://www.sevenforums.com/installation-setup/88596-win7-multi-language-all-one-dvd-error-0x8000001.html
[20] http://technet.microsoft.com/en-us/library/hh824900.aspx
[21] http://www.ubuntu.com/download/
[22] http://www.kubuntu.org/getkubuntu/
[23] http://xubuntu.org/getxubuntu/
[24] https://launchpad.net/usb-creator/
[25] https://fedorahosted.org/liveusb-creator/
[26] http://wiki.centos.org/TipsAndTricks/TroubleshootGRUB
[27] http://zhidao.baidu.com/question/70996291.html
[28] http://hi.baidu.com/neoli/blog/item/300754098c0a4b226b60fbee.html
[29] http://www.oracle.com/technetwork/server-storage/solaris11/downloads/index.html
[30] http://www.iatkos.me/
[31] http://www.hackintosh.com/
[32] http://www.insanelymac.com/
[33] http://www.osx86.net/
[34] http://chameleon.osx86.hu/
[35] http://iatkos.me/forum/viewtopic.php?f=77&t=23197
[36] http://en.wikipedia.org/wiki/GUID_Partition_Table
[37] http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/install-guide/s1-begininstall-hd.html
[38] http://www.minitool.com/
[39] http://www.easeus.com/
[40] http://gparted.sourceforge.net/
[41] http://www.acronis.com/homecomputing/products/diskdirector/
[42] http://www.paragon-software.com/downloads/home/
[43] https://help.ubuntu.com/community/Grub2
[44] http://www.gnu.org/software/grub/
[45] http://www.dedoimedo.com/computers/grub-2.html
[46] https://en.wikipedia.org/wiki/List_of_data_recovery_software
[47] http://www.handyrecovery.com/
[48] http://www.cgsecurity.org/wiki/TestDisk/

Anil Kumar Karna received a B.E. degree in Computer Engineering from Huazhong University of Science and Technology in 2004, and received a M.S. degree in Software Engineering from Shanghai Jiao Tong University in 2010. He is now a Ph.D. student in the Department of Computer Science and Engineering at Shanghai Jiao Tong Unviersity. His current research interests include testing & debugging.

Yuting Chen received a B.S. and a M.S. degrees in Computer Science from Nanjing University, China, in 2000 and 2003, respectively. He received a Ph.D. degree in Computer Science from Hosei University in 2007. He joined the School of Software at Shanghai Jiao Tong University in 2008. Yuting Chens research interests include program analysis and testing, and software verification and validation.