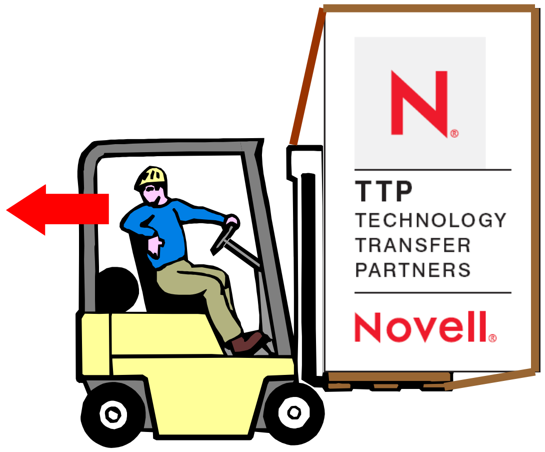


Linux backups

Joe R. Doupnik
MindworksUK and Univ of Oxford
jrd@netlab1.oucs.ox.ac.uk

Elements of this talk

Recognize a fundamental requirement of backing up and restoring

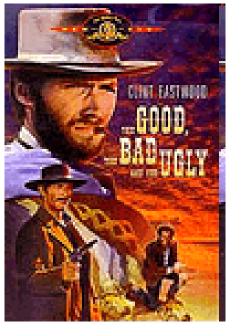
no longer will you wonder what to backup

Discover that Linux provides a major obstacle

Reach out far away and pull in an unrelated little known technique

Combine the works to achieve our goal (at no cost)

The good, the bad, & the ugly



Backing up files is “relatively” easy

Putting them back in their original location, with all their meta-bits, can be a real challenge

Not all files are real files. Some are only names in the file tree, without disk data, such as pipes, device nodes, sockets, and pseudo files. Some are not even visible.

We must consider all of the above

Our toolkit shopping list

Tar (what everyone thinks of first)
Star (an improved tar)

Dump, restore and xfsdump, xfsrestore

Nbackup (part of OES, TSAFS, does NSS and POSIX)

A tricky bit to be introduced later on

Rsync, amanda, bacula et al, plus your commercial product of choice

Tar and star

Not all Tar's are alike, beware

- not great at long filenames
- concerns about sparse files
- GNUtar is NOT POSIX compliant

Star

- POSIX compliant tar-like
- handles long filenames, character sets, sparse files
- artype=exustar is most complete, -xattr for extended attributes

Both record only what they know about, via VFS layer

Can span storage devices or be restricted to one

Both are awkward about restoring selected files from large archives

Dump and xfsdump

Dump **for EXT2/3,** ***xfsdump*** **for XFS**

Records all file detail (knows all internal details)

Talks to the block device driver, not the VFS layer

Records special files (without tripping over contents of /dev/zero and open sockets/pipes)

Records whole partition or a subset, does not span partitions

Interactive file restoration, an excellent feature

Restore **and** ***xfsrestore*** **do file restoration**

No, there is no dump for ReiserFS

Novell’s nbackup

- OES utility, works with eDir
- Tar-like command interface
- Works with POSIX and NSS volumes
- Uses SMS compliant mode:TSAFS, TSAIF, TSAGW
- Can backup/restore over the net using SMS
- Read/writes to file, tape drive, or pipe
- Records only what it knows, but SMS really knows NSS in detail
- Retains owner and trustee information, name spaces
- No interactive restoration mode

A short comparison chart

	tar/star	dump/xfsdump	nbackup	cp -a
Regular files	Yes	Yes	Yes	Yes
Device node, pipe, sockets	Metadata but quirky	Metadata	No	Metadata
Hard linked files	One copy	One copy	Yes, but avoid on NSS	One copy
Sparse files	Yes with -S or --sparse	Yes	Yes	Yes
POSIX xattr	tar -p star -xattr	Yes	Yes	Yes
NSS attributes and trustees	tar -p star -xattr	No (not NSS)	Yes	Yes
Remote src / destinaton	No	No	Yes	No

Nbackup and NSS xattr

29.4 Using Extended Attributes (xAttr) Commands (Linux)

In OES SP2 and later, NSS supports the Linux extended attributes (XAttr) option that allows listing, saving, and restoring the trustee information that is stored in the netware.metadata extended attribute.

Third-party backup software that supports the standard Linux Extended Attributes (xattr) can use this feature for NSS volumes to preserve trustees, trustee rights, file attributes, and quotas in backup and restore.

Support for Linux xattr is disabled by default. It is necessary to set the following NSS switches to enable this behavior:

```
nss /ListXattrNWMetadata
nss /CtimeIsMetadataModTime
```

If issued from the command line (nsscon), support is automatically disabled at the next server reboot. Enable support for Linux xattr across server reboots by adding the switches to file ***/etc/opt/novell/nss/nssstart.cfg*****.**

Note that nbackup transfers trustees regardless of this setting.

http://www.novell.com/documentation/oes2/stor_nss_lx_nw/?page=/documentation/oes2/stor_nss_lx_nw/data/backup.html

Nbackup of remote server

Authenticate to SMS as user *root*

```
# /opt/novell/sms/bin/nbackup --remote-target=129.67.103.181
   --target-type=Linux -cvf /tmp/test.nbk  /home/sys1/Linux
Operation performed as user root
Password:

/home/sys1/Linux/
/home/sys1/Linux/0003.txt
/home/sys1/Linux/0004.txt
/home/sys1/Linux/15523.html
       ...
```

Username *root* works to remote NSS Linux and POSIX sources

Nbackup of remote server

Authenticate to SMS as user *admin.oucs*

```
# /opt/novell/sms/bin/nbackup -U admin.oucs --remote-
   target=129.67.103.181 --target-type=Linux -cvf
   /tmp/test.nbk /home/sys1/Linux
Password:

/home/sys1/Linux/
/home/sys1/Linux/0003.txt
/home/sys1/Linux/0004.txt
/home/sys1/Linux/15523.html
        ...
```

Username *admin* works to source NSS

Nbackup restores, failed

Authenticate to SMS as user *admin.oucs*

```
# /opt/novell/sms/bin/nbackup -U admin.oucs -xvf
   /tmp/test.nbk -r "/etc/init.d /tmp/dummy"
Password:

nbackup: Received error: 0xfffdffe5 ((libtsafs.so 6.50.0 282)
   The data set handle is invalid.) from
   NWSMTSOpenDataSetForRestore
nbackup: Failed to restore: /tmp/dummy/rcS.d/S11microcode
```

Username *admin.oucs* is not known to a POSIX destination, unless that area is made into an NCP share

Nbackup restores, succeeded

Authenticate to SMS as user *root*

```
# /opt/novell/sms/bin/nbackup -U root -xvf /tmp/test.nbk
   -r "/etc/init.d /tmp/dummy"
Password:
    ...
/tmp/dummy/rcS.d/S11splash
/tmp/dummy/rcS.d/S09boot.clock
/tmp/dummy/rcS.d/S11microcode

Total data size: 731.00 KB
Elapsed time:  0.02 minutes

Throughput:    42.83 MB/min
```

User name *root* works to POSIX and NSS file destinations

What about rsync?

Handy for smart local/remote synchronization, as a copy-style program, not an archive filer

Sees files at the VFS level, always the top level of a mounted file system stack

Uses POSIX style rights, sees NSS with typically “long” name space (set by NSS volume mount command)

Has -x single-file-system, -sparse, -X xattrs, -A acls options

plus preservation of hard links option -H

What we see is not what we want

When we look at the file tree it encompasses many storage providers (partitions, file systems etc)

Tar looks at the file tree, but it can recognize when a mount point is encountered

Actually, we do NOT want the tree view of a server

We want only what is <u>on disk</u>, not other parts which are created dynamically as helpful illusions in the file tree

Getting below the tree level to the disk takes finesse

Spaghetti Western part again

Good: backing up things feels right, keeps our job

Bad: simple backups may not produce useful restores

Ugly: not everything we see is a disk file, some parts of a disk may be invisible

Let’s get the ugly part understood first

OES2 SP2, mounted file systems

File system types are in colour, **bold print** are disk based systems

```
$ mount
/dev/sdb3 on / type xfs (rw,noatime,nodiratime,logbufs=8)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
debugfs on /sys/kernel/debug type debugfs (rw)
udev on /dev type tmpfs (rw)
devpts on /dev/pts type devpts (rw,mode=0620,gid=5)
/dev/sdb2 on /boot type ext2 (rw,acl,user_xattr)
/dev/sdb4 on /home/extra type xfs (rw,noatime,nodiratime,logbufs=8)
/dev/sdc1 on /home/patches type xfs (rw,noatime,nodiratime,logbufs=8)
fusectl on /sys/fs/fuse/connections type fusectl (rw)
proc on /var/lib/ntp/proc type proc (rw)
novfs on /var/opt/novell/nclmnt type novfs (rw)
/dev/evms/USERPOOL on /opt/novell/nss/mnt/.pools/USERPOOL type nsspool
    (rw,name=USERPOOL)
/dev/evms/SYS1POOL on /opt/novell/nss/mnt/.pools/SYS1POOL type nsspool
    (rw,name=SYS1POOL)
admin on /_admin type nssadmin (rw)
SYS1 on /home/sys1 type nssvol (rw,noatime,name=SYS1,ns=long)
USER on /home/user type nssvol (rw,noatime,name=USER,ns=long)
/home/user/anonftp/pub on /home/ftp/pub type none (rw,bind)
```

Types in red are special and pseudo file systems mounted into the file tree, not easily backed up, not restorable, **tar has big problems**

Careful tar of a sensitive area

We need this option to stay out of trouble

```
# tar --one-file-system -cvf test.tar /var/lib/ntp
tar: Removing leading `/' from member names
/var/lib/ntp/
/var/lib/ntp/dev/
/var/lib/ntp/drift/
/var/lib/ntp/drift/ntp.drift
/var/lib/ntp/etc/
/var/lib/ntp/etc/ntp.conf.iburst
/var/lib/ntp/etc/localtime
/var/lib/ntp/etc/ntp.conf
/var/lib/ntp/var/
/var/lib/ntp/var/lib/
/var/lib/ntp/var/lib/ntp
/var/lib/ntp/var/run/
/var/lib/ntp/var/run/ntp/
/var/lib/ntp/var/run/ntp/ntpd.pid
/var/lib/ntp/proc/
tar: /var/lib/ntp/proc/: file is on a different filesystem; not dumped
#
```

Pseudo file system /proc is re-mounted here; it cannot be tar’d without very major problems

MindWorks UK

Some areas are most awkward

xfsdump view of
/dev area of root (/)

```
# xfsdump -l 0 -f test2.dmp  -s dev  /
# xfsrestore -i -f test2.dmp  /tmp
-> cd dev
-> ls
   100663466 zero          100663443 md7
    100663465 watchdog     100663442 md6
    100663464 ttyS7        100663441 md5
    100663463 ttyS6        100663440 md4
    100663462 ttyS5        100663439 md3
    100663461 ttyS4        100663438 md2
    100663460 ttyS3        100663437 md15
    100663459 ttyS2        100663436 md14
    100663458 ttyS1        100663435 md13
    100663457 ttyS0        100663434 md12
    100663456 tty1         100663433 md11
    100663455 tty          100663432 md10
    100663454 stdout       100663431 md1
    100663453 stdin        100663430 md0
    100663452 stderr       100663429 kmsg
    100663451 skip         100663428 fwmonitor
    100663450 rtc          100663427 fd
    100663449 route        100663426 core
    100663448 ptmx         100663425 console
    100663447 ppp           28129394 mapper/
    100663446 null         100745158 blog
    100663445 md9          100689050 initctl
    100663444 md8            8388737 shm/
                                 133 net/
                           109052032 pts/
```

These are on-disk

tar -clvf of /dev   -l is --one-file-system

```
# tar -clvf test2.tar /dev
/dev/
tar: /dev/log: socket ignored
/dev/vcsa2
/dev/vcs2
/dev/vcsa6
/dev/vcs6
/dev/vcsa5
/dev/vcsa4
/dev/vcs5
/dev/vcs4
/dev/vcsa3
/dev/vcs3
/dev/zapi
/dev/dm-4
/dev/dm-3
/dev/dm-2
/dev/dm-0
/dev/dm-1
/dev/evms/
/dev/evms/SYS1POOL
/dev/evms/USERPOOL
/dev/evms/.nodes/
/dev/evms/.nodes/sda1
/dev/evms/.nodes/SYS1POOL
/dev/evms/.nodes/USERPOOL
/dev/evms/.nodes/sda1.2
/dev/evms/.nodes/sda1.1
/dev/evms/.nodes/sdc
/dev/evms/.nodes/sdb
/dev/evms/.nodes/sda
/dev/evms/dm/
/dev/evms/dm/control
/dev/userModeNebDrv
/dev/ndp
/dev/nsscmd
/dev/admindrv
/dev/xconsole
/dev/fuse
```

/dev has file system "udev" tmpfs mounted on it

tar of /dev shows the udev contents, all dynamic

tar -clvf of / shows an empty /dev area (tar can not see into it with -l)

Shows what we do not want, hides what we do

and so on for 480 entries

These are not on-disk

Dump/xfsdump look beneath mount points

Both programs look at their primary file systems and ignore those mounted on top

The root of the primary file system must be exposed to get started

For example, if we were to mount an iso on top of *\/mnt*, which contained files, then

```
xfsdump -l 0 -f /home/extra/test.dmp /
```

will archive files in /mnt, ignoring the iso sitting on top

Nbackup of mounted upon area

Does not record what is beneath a mount point (hidden files)

Does not record files added above a starting file system by a mount operation

Records only what is on the starting point file system (real storage), <u>excluding</u> mount points and files hidden beneath mounts (skips things)

Rocks and shoals

Tar

- rides over the top of mount points
- cannot see underneath for reading or writing
+ not POSIX compliant but -p keeps trustees

Dump/xfsdump

+ reads underneath mount points
- restore/xfsrestore cannot write underneath
+/- file system kind dependent (EXT, XFS)

Nbackup (TSAFS)

- shuns mount points entirely
- neither reading nor writing above or below them
+ retains NSS trustee information

There is a way forward

The magic part, a taster

How to make subdirectories, or just individual files, appear at other places in the user level file tree -

- without copying
- without symbolic links or hard links
- without CIFS Junctions
- without Shadow Volumes/DST
- without revealing names of superior directories just to get to the interesting area
- without extra mounts on top
- without complexity or side effects
- without system loading

For both NSS and POSIX file systems (source and destination)

For all methods of access, users and applications alike

From my paper MountBind.[pdf | ppt]

How we do it: magic is explained

***Mount* man page excerpt --**

Since Linux 2.4.0 it is possible to remount part of the file hierarchy somewhere else.

The call is

```
mount --bind olddir newdir
```

After this call the same content is accessible in **two places**. One can also remount a **single file** (on a single file).

This call attaches only (part of) a single file system, not possible submounts.

The entire file hierarchy <u>including submounts</u> is attached to a second place using

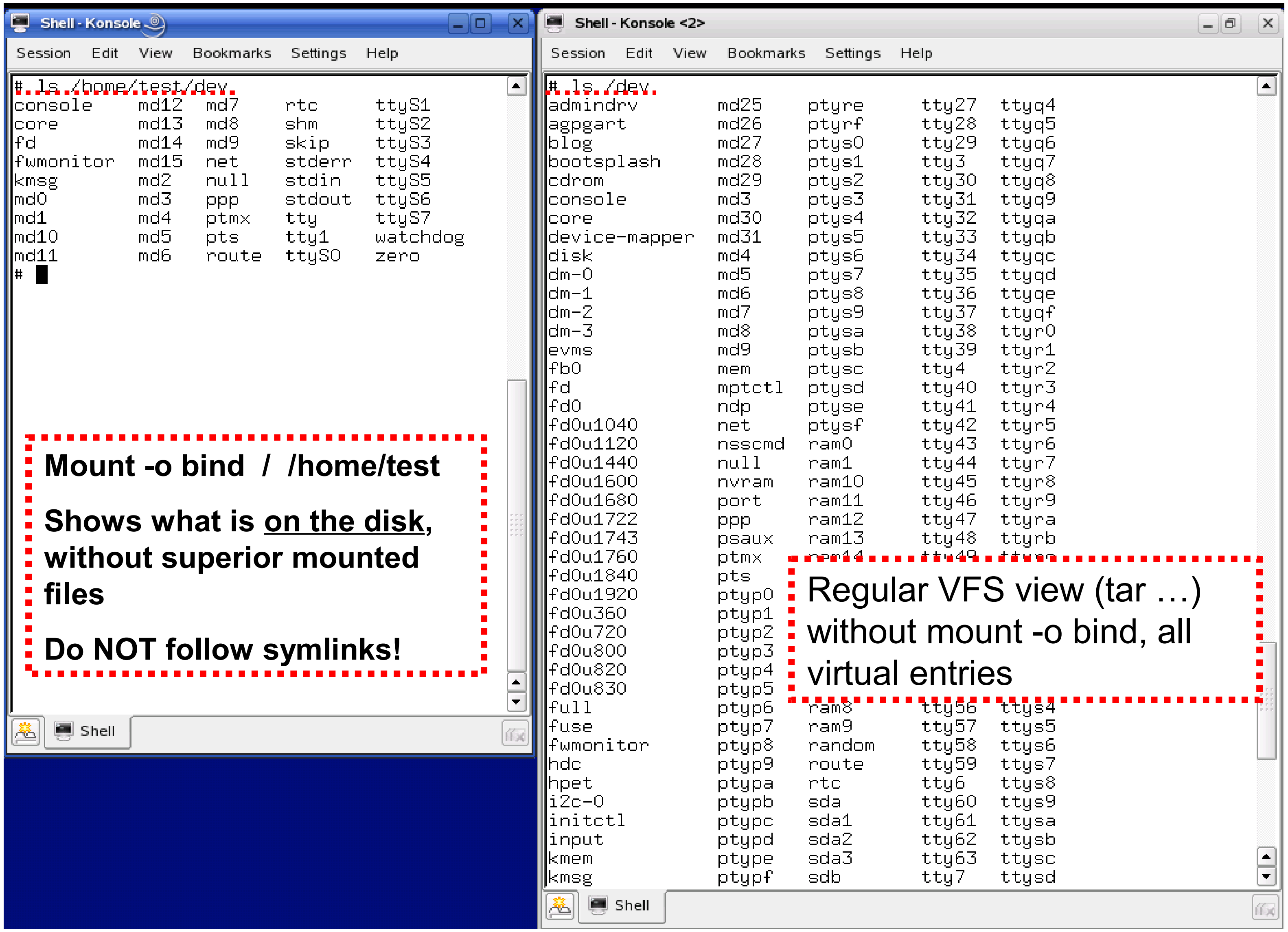
```
mount --rbind olddir newdir
```

Note that the file system <u>mount options will remain the same</u> as those on the original mount point, and <u>cannot be changed</u> by passing the -o option along with --bind/--rbind.

mount -o bind source destination   is better syntax

From man mount and my paper MountBind.ppt

Seeing the disk with mount bind


Shell - Konsole
Session Edit View Bookmarks Settings Help
Shell
Shell - Konsole <2>
Session Edit View Bookmarks Settings Help
Shell

```
# ls /home/test/dev
console     md12  md7    rtc     ttyS1
core        md13  md8    shm     ttyS2
fd          md14  md9    skip    ttyS3
fwmonitor   md15  net    stderr  ttyS4
kmsg        md2   null   stdin   ttyS5
md0         md3   ppp    stdout  ttyS6
md1         md4   ptmx   tty     ttyS7
md10        md5   pts    tty1    watchdog
md11        md6   route  ttyS0   zero
#
```

Mount -o bind / /home/test

Shows what is <u>on the disk</u>, without superior mounted files

Do NOT follow symlinks!

```
# ls /dev
admindrv        md25     ptyre      tty27  ttyq4
agpgart         md26     ptyrf      tty28  ttyq5
blog            md27     ptys0      tty29  ttyq6
bootsplash      md28     ptys1      tty3   ttyq7
cdrom           md29     ptys2      tty30  ttyq8
console         md3      ptys3      tty31  ttyq9
core            md30     ptys4      tty32  ttyqa
device-mapper   md31     ptys5      tty33  ttyqb
disk            md4      ptys6      tty34  ttyqc
dm-0            md5      ptys7      tty35  ttyqd
dm-1            md6      ptys8      tty36  ttyqe
dm-2            md7      ptys9      tty37  ttyqf
dm-3            md8      ptysa      tty38  ttyr0
evms            md9      ptysb      tty39  ttyr1
fb0             mem      ptysc      tty4   ttyr2
fd              mptctl   ptysd      tty40  ttyr3
fd0             ndp      ptyse      tty41  ttyr4
fd0u1040        net      ptysf      tty42  ttyr5
fd0u1120        nsscmd   ram0       tty43  ttyr6
fd0u1440        null     ram1       tty44  ttyr7
fd0u1600        nvram    ram10      tty45  ttyr8
fd0u1680        port     ram11      tty46  ttyr9
fd0u1722        ppp      ram12      tty47  ttyra
fd0u1743        psaux    ram13      tty48  ttyrb
fd0u1760        ptmx     ram14      tty49  ttyrc
fd0u1840        pts
fd0u1920        ptyp0
fd0u360         ptyp1
fd0u720         ptyp2
fd0u800         ptyp3
fd0u820         ptyp4
fd0u830         ptyp5
full            ptyp6    ram8       tty56  ttys4
fuse            ptyp7    ram9       tty57  ttys5
fwmonitor       ptyp8    random     tty58  ttys6
hdc             ptyp9    route      tty59  ttys7
hpet            ptypa    rtc        tty6   ttys8
i2c-0           ptypb    sda        tty60  ttys9
initctl         ptypc    sda1       tty61  ttysa
input           ptypd    sda2       tty62  ttysb
kmem            ptype    sda3       tty63  ttysc
kmsg            ptypf    sdb        tty7   ttysd
```

Regular VFS view (tar …) without mount -o bind, all virtual entries

Which tool to use?

Backup material actually <u>on disk</u>, partition by partition
Ignore extra mount points and pseudo file systems; these are, after all, created dynamically after booting

Green is good Blue is not	One file system	Read hidden	Write to hidden	
tar cpvf	**No**	**No**	**No**	cp -a is like tar cpvf
tar clpvf or star -xdev	**Yes**	**No**	**No**	l is single-file-system
nbackup	**Yes**	**No**	**No**	Wise choice for NSS & remote OES2 file systems
dump & xfsdump	**Yes**	**Yes**	**No**	Wise choice for XFS/EXT
mount bind +above three	**Yes**	**Yes**	**Yes**	Avoids mount effects

None of the tools alone restores files to spots beneath mount points, but use of *mount -o bind* can provide mount-free access for proper restoration

MindWorks Inc. Ltd
210 Burnley Road
Weir
Bacup
OL13 8QE UK

Telephone: +44 (0) 170 687 1900

Fax: +44 (0) 170 687 8203

Web: www.mindworksuk.com

Email: training@mindworksuk.com