Linux

Void Linux distribution and general Linux stuff.

XBPS Packages

I have hosted packages under https://voidlinux.jpastuszek.net.

Usage

/etc/xbps.d/jpastuszek.conf

```
repository=https://voidlinux.jpastuszek.net
```

Building packages

I keep void-packages repo on GitHub and in private repo:

- For upstream updates: https://github.com/void-linux/void-packages
- ssh://git@git.hexadust.net/user/brombek/void-packages
- git@github.com:jpastuszek/void-packages.git

1. The master branch is used as base and is kept in sync with official repo.
2. I create new branches from the master branch for each package.
3. Each package branch should be rebased on the current master branch.

Building:

```
# morgana
./xbps-src -j 10 pkg <package>
# aarch64
./xbps-src -j 10 -a aarch64 pkg <package>
```

Indexing new packages

.ssh/config

```
Host www
	HostName 192.168.50.159
	ProxyJump igor.lan
	User hxd
```

repoindex

```
#!/bin/sh -x
cd public && \
xbps-rindex --privkey ~/.ssh/voidlinux.jpastuszek.net.pem --sign-pkg *.xbps && \
XBPS_TARGET_ARCH=x86_64 xbps-rindex --add *.x86_64.xbps && \
XBPS_TARGET_ARCH=x86_64 xbps-rindex --clean . && \
XBPS_TARGET_ARCH=aarch64 xbps-rindex --add *.aarch64.xbps && \
XBPS_TARGET_ARCH=aarch64 xbps-rindex --clean .
```

```
# Mount
exec sshfs www:/var/www ~/net/www/
net/www/voidlinux.jpastuszek.net
cp <package> public/
./repoindex
```

> This needs to be done from morgana since it has the key for indexing: ~/.ssh/voidlinux.jpastuszek.net.pem

OS Setup

Swap file

On Btrfs:

```
btrfs subvolume create /.swap
truncate -s 0 /.swap/swapfile
chattr +C /.swap/swapfile
fallocate -l 2G /.swap/swapfile
chmod 600 /.swap/swapfile
mkswap /.swap/swapfile
echo "/.swap/swapfile none swap rw,nofail 0 0" >> /etc/fstab
swapon -a
```

On ext4:

```
truncate -s 0 /swapfile
fallocate -l 2G /swapfile
chmod 600 /swapfile
mkswap /swapfile
echo "/swapfile none swap rw,nofail 0 0" >> /etc/fstab
swapon -a
```

Chroot to Linux root

```
cd /mnt/<root>
mount -o bind /dev dev
mount -o bind /proc proc
mount -o bind /sys sys
mount -o bind /sys/firmware/efi/efivars sys/firmware/efi/efivars
mount -o bind /run run
```

Void Linux

- ISO: https://voidlinux.org/download/
- Use x86_64 Live image glibc

Basic packages

```
xbps-install xtools fish-shell helix
chsh -s /usr/bin/fish root
chsh -s /usr/bin/fish hxd
ip addr
```

Now SSH to host:

```
ssh hxd@192.168.0.x
sudo -i
```

Custom repository and mascot

- Repository URL: https://voidlinux.jpastuszek.net/
- Fingerprint: `63:d7:85:df:db:75:a3:3e:e8:96:9f:78:49:ee:43:ff`

```
echo "repository=https://voidlinux.jpastuszek.net" > /etc/xbps.d/jpastuszek.conf
xi -S mascot mascot-delegate-xbps-package mascot-delegate-runit-service
```

Get `sudo` working without password so you can run `mascot-converge` with `sudo`:

```
echo "%wheel ALL=(ALL:ALL) NOPASSWD: ALL" > /etc/sudoers.d/wheel
```

Converge with mascot:

```
www/base-access | ssh hxd@192.168.0.x sudo mascot-converge
www/base-hw | ssh hxd@192.168.0.x sudo mascot-converge
www/rc | ssh hxd@192.168.0.x sudo mascot-converge
www/xbps | ssh hxd@192.168.0.x sudo mascot-converge
www/syslog-dmesg | ssh hxd@192.168.0.x sudo mascot-converge
```

Finalize with:

```
xbps-reconfigure -fa
reboot
```

PKI

Setup

Put `openssl.conf`:

openssl.conf

```
distinguished_name  = $ENV::DN
x509_extensions     = $ENV::EXTENSIONS

string_mask = nombstr         # This sets a mask for permitted string types
prompt      = no              # Don't ask questions

default_bits      = 4096      # Key length to use (due to req bug this is also specified
in cmd line)
default_md        = sha256    # The message digest to use

[ ca ]
default_ca    = CA_default    # The default ca section

[ CA_default ]
database      = index.txt     # The text database file to use. Mandatory. This file must
be present though initially it will be empty
serial        = serial        # A text file containing the next serial number to use in
hex

unique_subject  = no          # If the value no is given, several valid certificate
entries may have the exact same subject
email_in_dn     = no          # If you want the EMAIL field to be removed from the DN of
the certificate simply set this to 'no'
preserve        = no          # keep passed DN ordering
name_opt        = ca_default  # Subject name display option
cert_opt        = ca_default  # Certificate display option

policy          = ca_policy   # Policy section
```

```
[ ca_policy ]
countryName             = supplied
stateOrProvinceName     = supplied
localityName            = supplied
organizationName        = supplied
organizationalUnitName  = supplied
commonName              = supplied

[ ca_extensions ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid:always,issuer:always
basicConstraints        = CA:true
keyUsage                = cRLSign, keyCertSign
nameConstraints         = critical,permitted;DNS:.$ENV::FQDN

[ cert_extensions ]
subjectKeyIdentifier    = hash
authorityKeyIdentifier  = keyid,issuer:always
basicConstraints        = CA:FALSE
keyUsage                = digitalSignature, keyEncipherment
extendedKeyUsage        = serverAuth, 1.3.6.1.5.5.8.2.2
subjectAltName          = $ENV::ALT
nsCertType              = server
nsComment               = "HxD Internal Certificate"

[ ca_dn ]
countryName             = IE
stateOrProvinceName     = Dublin
localityName            = Dublin
0.organizationName      = Hexa Dust
organizationalUnitName  = Ops
commonName              = "HxD Internal Certification Authority"

[ cert_dn ]
countryName             = IE
stateOrProvinceName     = Dublin
localityName            = Dublin
0.organizationName      = Hexa Dust
```

```
organizationalUnitName  = Ops
commonName              = $ENV::FQDN
```

Create `certs` dir:

```
mkdir certs
```

Certificate request

Use `./mkreq <fqdn>` to create request file.

mkreq

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export ALT=

openssl req \
    -new -newkey rsa \
    -keyout "certs/$FQDN.key" \
    -out "certs/$FQDN.csr" \
    -config openssl.conf
```

Self-sign certificate request

Request file can be self-signed using `./selfsign <fqdn>`.

selfsign

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export ALT="DNS:$FQDN"

openssl req -x509 \
-days 7120 \
-nodes \
-new -newkey rsa \
-keyout "certs/$FQDN.key" -out "certs/$FQDN.crt" \
-config openssl.conf
openssl x509 -text -in "certs/$FQDN.crt"
```

CA certificate and signing

For CA signed certificates we need CA certificate and key and then use it to sign the certificate requires.

Create CA certificate and key

Use . `/mkca <domain>` to create CA for given domain. The `<domain>.crt` file can be installed in a browser or system wide as trusted CA.

mkca

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=ca_extensions
export FQDN="$1"
```

```
export ALT=

test -f "$FQDN.key" || openssl genrsa \
  -aes256 -out "$FQDN.key" 4096
test -f "$FQDN.crt" || openssl req \
  -x509 \
  -new -nodes -key "$FQDN.key" \
  -sha256 -days 14240 \
  -out "$FQDN.crt" \
  -config openssl.conf
```

CA-sign certificate request

Use `./casign <fqdn>` to create CA-signed certificate.

casign

```
#!/bin/sh
set -u

export DN=cert_dn
export EXTENSIONS=cert_extensions
export FQDN="$1"
export CA=`echo "$FQDN" | sed -r 's/.*\.([^\.]+\.)/\1/'`
test "$CA" = "$FQDN" && export CA=`echo "$FQDN" | sed -r 's/.*\.//'`
shift
ALT="DNS:$FQDN"
for A in "$@"; do
  ALT="$ALT,DNS:$A"
done
export ALT

openssl x509 -req \
  -in "certs/$FQDN.csr" \
  -CA "$CA.crt" -CAkey "$CA.key" -CAcreateserial \
```

```
-sha256 -days 14240 \
-extensions $EXTENSIONS \
-extfile openssl.conf \
  -out "certs/$FQDN.crt"
openssl x509 -text -in "certs/$FQDN.crt"
```

Decrypt certificate key file

Use `./dec <fqdn>` to decrypt key file to `stdout`.

dec

```
#!/bin/sh
set -u
FQDN="$1"

openssl rsa -in "certs/$FQDN.key"
```

i3blocks

Requires `font-awesome` (4) and `font-awesome5` OR `font-awesome6` OR all of them installed.

/etc/i3blocks.conf

```
# Global properties
separator=true
separator_block_width=15

[battery]
command=i3blocks-rustlets-battery
interval=30
signal=1
markup=pango

[battery-refresh]
command=acpi_listen | while read F; do pkill -x -SIGRTMIN+1 i3blocks; done
interval=persist

[wifi]
command=i3blocks-rustlets-wifi
interval=30
markup=pango
signal=2

[wifi-refresh]
command=dbus-monitor "interface=org.freedesktop.portal.NetworkMonitor" | while read F; do
pkill -x -SIGRTMIN+2 i3blocks; done
interval=persist

[cpu]
command=i3blocks-rustlets-cpu
interval=persist
markup=pango

[mem]
command=free -g | tail +2 | tr "\n" ' ' | awk '{print "□ " $7 "Gi " $11 "Gi"}'
```

```
interval=4

[disk]
command=df -hH / /boot/ | tail +2 | tr "\n" ' ' | awk '{print "  " $4 " " $10}'
interval=60

[mullvad]
command=echo -n "  "; curl -s https://am.i.mullvad.net/json | jq '[.city , .ip] | join("  ")'
-r
interval=30

[time]
command=echo "  <span color='#19f5af'>`date '+%a %Y-%m-%d %H:%M:%S'`</span>  `date --utc
'+%H:%M'`"
markup=pango
interval=1

[greetings]
command=echo "  <span color='#f5af19'>$USER</span>"
interval=once
markup=pango
```

PipeWire

Tools

- `pw-record` & `pw-play` - capture and play sound
- `qpwgraph` - visual tool for making links between ports
- `pw-dot` - dump graph as dot file: `pw-dot && dot -T svg -O pw.dot && firefox pw.dot.svg`
- `pavucontrol` - PulseAudio mixer GUI
- `pw-dump` - writes JSON representation of nodes, links and ports
- `pw-top` - stats

Playback over SSH

```
pw-record --target 56 - | ssh ann XDG_RUNTIME_DIR=/run/user/hxd pw-play -
```

Where `56` is ID of output monitor named `alsa_output.pci-0000_00_1f.3.analog-stereo` in my case.

```
pw-dump | jq '.[] | select(.info.props."node.name" == "alsa_output.pci-0000_00_1f.3.analog-stereo").id'
pw-dump | jq '.[] | select(.info.props."node.description" == "Built-in Audio Analog Stereo" and .info.props."api.alsa.pcm.stream" == "playback").id'
pw-dump | jq '.[] | select(.info.props."device.profile.name" == "analog-stereo" and .info.props."api.alsa.pcm.stream" == "playback").id'
```

Full script can look like this:

```
#!/bin/fish
pw-record --target (pw-dump | jq '.[] | select(.info.props."device.profile.name" == "analog-stereo" and .info.props."api.alsa.pcm.stream" == "playback").id') - | ssh ann XDG_RUNTIME_DIR=/run/user/hxd pw-play -
```

Mount image files with loop dev

Mounting file systems from loop device:

```
losetup -f <path to .img file>
dmesg | tail -n 1 # see which loop device got allocated for the image
partprobe /dev/loopX # probe partitions
mount /dev/loopXp1 /mnt/img1/ # mount partition
```

Once done:

```
umount /mnt/img1/ # umount all mounted partitions from the image loop devices
losetup --detach /dev/loopX # get rid of loop device
```

Copying and imaging disk drives

Copy image to disk with Disk Destroyer

```
dmesg -w # and plug in your drive; see dev name that was added
lsblk # verify device name
fdisk -l /dev<output disk> # verify device name, product name (Disk model) and size

# WARNING: this will overwrite output (of) disk data
dd if=<input imag file img> of=/dev/<output disk> bs=1M oflag=direct status=progress
```

Backup disk to image file

Uses `dd` for copy and `pv` for progress bar, applies compression with `zstd`.

```
#!/bin/sh -eu
INPUT_DEV=$1

sudo fdisk -l "${INPUT_DEV:?}" 1>&2
BYTES=$(sudo fdisk -l "${INPUT_DEV:?}" | head -n 1 | cut -d' ' -f5)

echo 1>&2
echo "Backup from ${INPUT_DEV:?} (${BYTES:?} bytes)" 1>&2
echo "Continue? [N/y]" 1>&2

read Y
test "$Y" = "y" || exit 0

sudo dd iflag=direct if="${INPUT_DEV:?}" bs=256K \
| pv -ptebar --average-rate-window=120 -T --buffer-size 512K -s "${BYTES:?}" \
| zstd -14
```

> Make sure to tweak `zstd`'s compression rate (max is 19) if you get CPU bound (100% CPU and no IO wait time) to reduce imaging time.

>

> Use `-L15M` or similar option on `pv` to limit backup rate if source device is misbehaving under high load (e.g. USB drives).

BBS

BBS

Z-Modem

Use `screen` command to connect to BBS with `telent`.

.screenrc

```
zmodem catch
```

```
!!! rz -vv -b -E
```

Monitoring

Tools

Network

- nload - interface traffic graphs
- bwm-ng - interface traffic stats (no graph)
- jnettop - IP flows, TCP and UDP
- tcptrack - TCP only
- btop - also shows network stats
- conntrack - tracked connection by kernel
- netstat or ss - show established local connections and listening ports

List netfilter connection tracking state:

```
conntrack -L
```

Zabbix

Custom parameters

Reload configuration:

```
zabbix_agentd -R userparameter_reload
```

Put config files in `/etc/zabbix/zabbix_agentd.d`.

Example configuration to add additional keys to agent to respond to:

```
UserParameter=ping.rtt.avg[*],ping -c 10 -A -n -q -w 10 "$1" | awk -F/ '/rtt/ { print $$5 }'
UserParameter=ping.loss.pct[*],ping -c 10 -A -n -q -w 10 "$1" | awk '/packets transmitted/ {
print 100 - int($$4) * 10 }'
```

Testing:

```
zabbix_get -s 127.0.0.1 -k 'ping.rtt.avg[127.0.0.1]'
```

Rsync

Copy content preserving attributes of `/foo` to `/bar`:

```
rsync -aXHA --info=progress2 --stats /foo/ /bar
```

To exclude specific paths from `/foo`, for example: `--exclude=/.snapshots --exclude=/.swap`

To use `.rsync-filter` files as exclude info use `-F` flag. For example:

```
- /.snapshots/
- /.swap/
```

LUKS encryption

Set up new encrypted volume

Format

```
cryptsetup luksFormat --type luks2 /dev/<dev>
cryptsetup luksDump /dev/<dev>
```

Note the UUID.

Open

```
cryptsetup luksOpen --allow-discards /dev/<dev> <name>
```

Creates new device `/dev/mapper/<name>`.

Format

```
mkfs.btrfs /dev/mapper/<name>
```

Note UUID.

Boot from unencrypted boot partition into encrypted root

Update GRUB configuration

In `/etc/default/grub`:

```
GRUB_CMDLINE_LINUX_DEFAULT="bgrt_disable loglevel=4 rd.luks.uuid=<LUKS UUID> rd.luks.allow-discards"
```

Update boot configuration

Make sure `/boot` and `/boot/efi` are mounted:

```
mount -a
```

Update grub:

```
update-grub
```

Update *initramfs*:

```
xbps-reconfigure -f linux6.6
```

(Re)install grub; make sure `/sys/firmware/efi/efivars` is bound if using `chroot` and `/boot` and `/boot/efi` are mounted:

```
grub-install /dev/<dev>
```

> If running into `symbol `grub_is_shim_lock_enabled` not found` error try removing `/boot/efi/EFI` and `/boot/grub` directories before running grub-install.

Where `<dev>` is the main device (not partition).

Check that grub had good configuration:

```
  cat /boot/grub/grub.cfg | egrep 'crypt|luks'
```

Verify that `root=UUID` point to FS UUID and `rd.luks.uuid=` to LUKS UUID.

Git

Default config

```
[user]
	email = hxd@hexadust.net
	name = Hexa Dust
	signingkey = FF16E5FDA4287A4B018EB2DFD41D2F6DCBCB743C
[pull]
	rebase = true
[init]
	defaultBranch = main
[commit]
	gpgsign = true
```

Sign commits

```
[user]
	email = hxd@hexadust.net
	name = Hexa Dust
	signingkey = FF16E5FDA4287A4B018EB2DFD41D2F6DCBCB743C
[commit]
	gpgsign = true
```

Reset author and sign all commits

```
git rebase -r --root --exec 'git commit --amend --no-edit --reset-author -S'
```

Replacing Switch SD card

Migration to bigger SD card

> No tool under Linux can extend ExFAT filesystem. Files have to be copied.

Replicate partition layout from old SD card to new cardt. Disk Destroyer can be used to initialize the partition with same IDs etc. by coping some data over. There is no point in DD'ing all the data as we can't resize the filesystem afterwards. Alternatively `sfdisk -d` can be used to dump the partition table for old SD card and restore it to the new one.

Make sure that `fdisk -l` shows same details for both SD cards: new and the original.

On new SD card delete the partition and create new. Observe the starting sector number - use same as the original partition (it is not the suggested default value in my case).

Check the output of `blkid` to see the *UUID* of the original SD card filesystem.

Format new partition and set ID to be the same as original `mkfs.exfat -i <UUID> <new SD card p1>`. ID can also be set later with `tune.exfat -I 0x<UUID in hex> <new SD card p1>`.

> Check again with `blkid` to see if *UUID*s are matching. If they are not, the new card will "work" in Switch but no game will start, complaining to go back to Home screen and start it again.

Now mount both cards and copy file over to the new card.

After `umount`'ing you can put new card in Switch and it should work and you should have more space for games.

Nvidia GeForce RTX 3050

TL;DR: Utter piece of hot garbage - literally. Don't buy yesterday, today or ever. Linus was right!

H/W

GeForce RTX 3050 low profile for compact Dell PC. I needed something below 80W of power drain due to limited power supply (that is rated for 80W GPU).

- Model: MSI GeForce RTX™ 3050 LP 6G
- GeForce RTX 3050 - NV176 (GA106) - Ampere family
- 6GB RAM
- Low profile
- Rated 70W

Void Linux driver setup

```
xi linux-headers nvidia nvtop
```

1. Make sure `linux-headers` are installed and system is up-to-date (same kernel and headers version).
2. Install `nvidia` package. Also `nvtop` comes handy - should how your card if driver is loaded correctly.
3. Use `xbps-reconfigure -f linux<version>` to see if DKMS driver was build correctly (no warnings about it).

Sway

Sway requires KMS. For some reason this requires flag for kernel to be enabled:

In `/etc/default/grub` add to kernel list and reload grub with `update-grub`:

```
nvidia-drm.modeset=1
```

You can run `cat /sys/module/nvidia_drm/parameters/modeset` after reboot to see if it got enabled.

Sway needs to be started with `--unsupported-gpu`.

Minecraft

Minecraft under Sway works with GeForce RTX 3050 but with higher power usage I get glitching on the screen, I can use shades that or FPS limit that does not go over 60W or so. This card is within power budget of the power supply but some people say that this GPU series tend to have spiky power drain! Hopefully this is a driver issue that gets resolved...

Without drivers

Running GPU without installing drivers and using integrated GPU is not a good idea as by default it will run fans and heat up as it does no do power management by VBIOS! It required blobs to manage power!

Nouveau

Driver work fine (tested with Minecraft) but GPU is not power managed due to lack of blobs! This makes it 6x slower than internal GPU (Intell UHD Graphics P630)!

Update: There is hope with the GSP firmware: https://www.phoronix.com/news/Nouveau-GSP-Merged-Linux-6.7

Batocera

Worked out of the box with Batocera so no power management issues.

Steam game (Spore) I tried failed to start! Works well with Intell UHD Graphics P630 - could be just this game so further testing needed.

Kodi

LibreELEC does not support Nvidia on Wayland - stuck at booting. I was able to run Kodi on X11 on Batocera but you don't get HDR that I get from integrated GPU!

Current status: https://libreelec.tv/2025/08/15/libreelec-omega-12-2-0/

> Future nVidia GPU support remains a grey area and we continue our long-running advice to avoid purchasing nVidia GPU cards for LibreELEC use.

Worse is that now if I want to use Kodi with integrated GPU, the Nvidia card heats up and becomes noisy and it cannot be disabled in any way!

Summary

- Kodi - no HDR possible (X11), worse than integrated Intel card! Running hot when using integrated GPU (no power mgmt without driver loaded)!
- Batocera - works fine, but Steam game did not work!
- Linxu in general:
  - Without drivers (unused card, using integrated) - drains power and loud due to lack of power management!
  - Nouveau - usable but way slower than integrated card and as above!
  - DKMS - driver works fine but have serious glitching at higher power levels - not very usable!

Expanding paritions

Backup partition table

```
fdixk -x /dev/<dev>
sfdisk -d /dev/<dev> > partition.dump
```

Shrinking EXT(2,3,4) file system to minimum size

```
e2fsck -f /dev/<dev partition>
resize2fs -M /dev/<dev parition>
```

Editing partition table and restore

After shining the last partition you need to:

1. Image it (see the other pages) and any other portions that need to be moved.
2. Use `fdisk` to delete the resized last partition.
3. Move other partitions (keeping exact size).
4. Expand the target partition (keep starting sector unchanged).
5. Copy data from images into moved and last partition.
6. Resize the target partition and the last partition to max size with `resize2fs /dev/<dev parition>`.

> Use `fdisk`'s expert mode (`x`) to print full partition information and to restore partitions UUID, name and attributes.

> To resize `vfat` (FAT32) partition you can use `fatresize -s max /dev/<dev partition>` (from `parted` package).