Új szkript kép áttetszövé tételéhez

PCLinuxOS Magazine - 2026. július

Írta: Paul Arnote (parnote)

Egíáltalán nem titok: SOK időt töltök grafikus fájlokkal a The PCLinuxOS Magazine főszerkesztőjeként. Azt hihetné az ember, hogy amilyen sok időt töltök grafikus fájlokkal, bizonyára GIMP-guru lennék... de nem vagyok. Ha értékelnem kellene a GIMP-tudásomat, csak egy hajszálnyival vagyok az átlag felett (szerintem). A The PCLinuxOS Magazine stábjában Meemaw a grafika guru.

Ha *kívánhatnék* készséget, képességet, az lenne, hogy meg tudjak nyitni egy képet GIMP-ben, eltávolítani hátteret a képről, majd a hátteret átlátszóságra cserélni. Ezt a feladatot annyi év után, még ma is csak teljesen véletlenül tudom megcsinálni (és nem azért, mert tudom, hogyan kell). Legtöbbször elküldöm a képet Meemaw-nak, és Ő visszaküldi nekem az átlátszó képet egy e-mail mellékleteként.

Őszintén szólva, biztos meg tudnám tanulni, ha rávenném magamat a tanulásra, de sok más dolgot is nyomon kell követnem, amik lekötik a figyelmemet ... a magazinnal és nem a magazinnal kapcsolatos dolgok. Tehát ez egy tanulási lehetőség, amit hagytam mellékvágányon más, magasabb prioritást élvező kötelezettségvállalások miatt. Hé... legalább beismerem.

Julien Tromeur képe a Pixabay-ről

Miközben nem vagyok csúcs GIMP-felhasználó (örömmel hagyom ezt a címet Meemaw-ra … és örülök, hogy van Ő), de az átlagnál sokkal többet tudok, amikor a grafikus fájlformátumok, azok képességei és hiányosságai kerülnek szóba. Ez az ismeret kifejezetten hasznos volt az új szkript megírásakor.

Még megszámolni sem tudom, hogy a The PCLinuxOS Magazine főszerkesztőjeként eltöltött időm során hányszor kellett, vagy akartam olyan képet, aminek Átlátszó volt a háttere. Végén mindig találtam valamilyen megoldást (vagy csak elküldtem a képet Meemaw-nak, hogy varázsoljon egyet, amikor nem találtam meg).

Egy kis történelem

Szóval lehet, most azon tűnődsz, miért vagy hogyan alakult ez így és állt elő ez a helyzet. Még 2013 márciusában írtam néhány szkriptet, ami megjelent a The PCLinuxOS Magazine-ban. Az egyiket „convert-image.sh”-nak, a másikat pedig „img-resize.sh.”-nak hívták Aztán 2024 februárjában újra elővettem és frissítettem az img-resize szkriptet. 2023 márciusában pedig a convert-image szkript lett frissítve.

Az igazság az, hogy az évek során többször is kisebb mértékben frissítettem a két szkriptet. Őszintén szólva két olyan szkriptről van szó, amiket Meemaw és én is gyakran használunk, amikor a magazin PDF-kiadásának cikkeit szerkesztjük. Szinte mindig. Mivel ezeket a szkripteket már kétszer bemutattuk a magazinban, nem vettem a fáradságot, hogy kirakjam vagy közzétegyem frissített verzióikat.

De valahol, valami megváltozott az ImageMagick-ben (aminek az eszközei végzik a munka dandárját ezekben a szkriptekben). Az egyik, ami más, hogy néha a szkriptek fekete hátterű képet hoznak létre, nem pedig átlátszót, ahogyan az elvárható lenne.

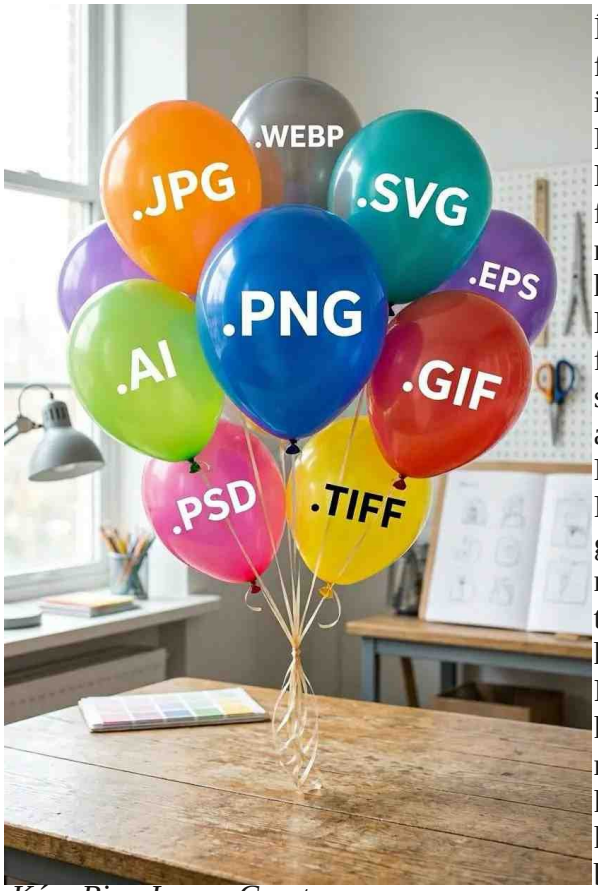
Így a közelmúltban nekiláttam a szkriptek „javításának”. Legnagyobb megdöbbenésemre azt kellett látnom, hogy az ImageMagick meglehetősen „kényes” lett a kiadott parancsok sorrendjét illetően. Ha valaha is elolvastad az ImageMagick parancsainak „használati szabályait”, akkor tudod, hogy milyen mérhetetlenül zavarosak ezek az leírások. Mindent elmondanak, amit tudnia kell az összes parancsról és az összes lehetőségről... kivéve a végrehajtás sorrendjét. Ha mégis ott lennének ezek az információk, egyszerűen nem találtam meg a szó szerinti hegynyi információban. A honlapjukon próbálni valamit megtalálni olyan, mintha egy gombostűt keresnék egy egy hektárnyi méretű sűrű gazban.

Mivel nem találtam olyan opciót, vagy parancs-sorrendet, ami az igényeim szerint működött, más megközelítést alkalmaztam. Készítettem egy olyan szkriptet, ami kizárólag áttetszővé teszi a képet. Elneveztem **remove-bg2.sh**-nak (a remove-bg.sh, az eredeti verzió csak egy „koncepciót igazoló” szkript volt). Noha az eredeti szkript „javítása” továbbra is tervben van, ez az új szkript egyelőre kielégíti az igényeimet.

Röviden a grafikus fájlformátumokról

Az összes elérhető fájlformátum közül csak hat jelentősebb támogatja az átlátszóságot. Ezek közül kettőt azonnal ki is zárhatunk a vizsgálatból: a GIF-et és a HEIC-t. A GIF fájlok csak az egybites átlátszóságot támogatják, így nem sokra megyünk vele (továbbá vannak más, komolyabb gondok is, de ezek a cikk témáján túlmutatnak). A HEIC elsősorban az Apple-eszközökön használatos és (előnyei ellenére) az Apple világán kívül nem nyert túl sok teret.

A fennmaradó többi nagyobb grafikus fájlformátum … a JPG (és a ***legtöbb*** változata), a BMP illetve a többi zöme … nem támogatja az átlátszóságot.



Kép: Bing Image Creator

Így már csak négy grafikus formátum maradt, amivel igazából foglalkoznunk kell: PNG, WEBP, AVIF és TIFF. Bár a TIFF-et a kimeneti formátumok közé soroltam, nem látom, hogy sokan a használnák, mivel NAGYON nagy méretű fájlokat állít elő. Csak azért soroltam ide, mert támogatja az átlátszóságot. Mindannyian ismerjük a PNG fájlokat, és a WEBP grafikák is egyre gyorsabban nyernek teret. Valójában több mint két éve használunk WEBP grafikát a PDF és HTML magazinhoz, köszönhetően kiváló minőségének és lényegesen kisebb fájlméreteinek. A kiugró érték itt az AVIF, ami bár minőségi grafikus formátum, nem kapott akkora elfogadottságot, mint a WEBP grafika. Az AVIF fájlok határozottan vetekednek a WEBP-vel, mivel kiváló minőségű képeket készítenek lényegesen kisebb fájlmérettel. Kilóg a sorból az AVIF, leginkább azért nem használjuk a magazinban, mert a Scribus (a program, amit a magazin PDF-kiadásának elkészítéséhez használunk) még nem támogatja a használatukat. Ha, vagy amikor a Scribus támogatja az AVIF grafikát, nyitottak leszünk ezek használatára. Az összes „jelentősebb” webböngésző

már támogatja az AVIF grafikát, így nem zárkózunk el attól, hogy a HTML-kiadásban használjuk.

Így, most (és a PCLinuxOS Magazine-ban használatban) eléggé leragadtunk a PNG és WEBP grafikus fájlok használatánál, amikor átlátszósággal bíró képpel van dolgunk.

Legyünk átlátszóak

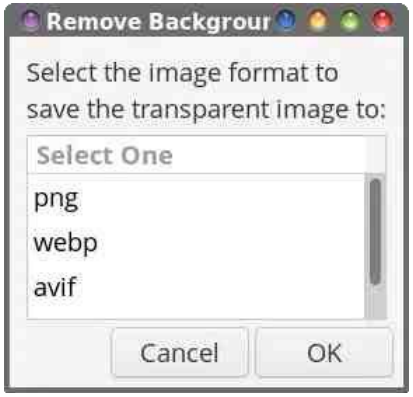
Ezt, mint számos más szkriptet, amiket az évek során írtam, úgy terveztem, hogy parancssorból, vagy Thunar egyéni műveletként fusson. Biztos vagyok benne, hogy más fájlkezelők felhasználói is használhatják a szkriptet hasonló módon. Minthogy, én szigorúan Xfce-felhasználó vagyok, más fájlkezelők esetében a pontos folyamatot nem ismerem jól.

Természetesen, letöltheted a szkriptet a magazin szerveréről. A fájl neve „remove-bg2.sh.txt,”, mérete pedig csak 3,2 KiB. Amint azt más, a magazinban általam bemutatott szkripteknél korábban említettem, abban a könyvtárban tárold, ahová általában mented a bash szkripteket. Távolíts el a „.txt” fájlkiterjesztést, és tedd végrehajthatóvá a fájlt. Remélhetőleg a könyvtár, ahová a bash szkripteket mented, a számítógép $PATH változójában megtalálható. Ha igen, akkor csak ki kell adnod a szkript nevét (remove-bg2.sh) a szkript elindításához. Ellenkező esetben a teljes elérési utat kell megadni a szkript a futtatásához.

Nos, mielőtt a szkriptről beszélnénk, vessünk rá egy pillantást.

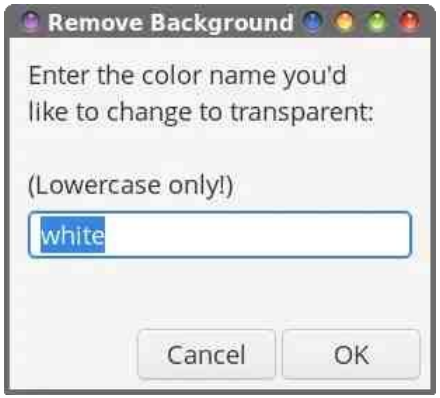
```
1.#! /bin/bash
2.#
3.# Written by Paul Arnote, Chief Editor of The PCLinuxOS Magazine
4.# Read the original article in the July 2026 issue of The PCLinuxOS
5.# Magazine. https://pclosmag.com/html/Issues/202607/links.html
6.#
7.# Released under the GPL 2.0 license
8.# May be freely distributed in accordance to the GPL 2.0 license
9.#
10.# This script will take an image and change the specified color
11.# to transparent, and then resave that image to one of the four
12.# graphic formats supporting transparency (PNG, WEBP, AVIF, or
TIFF).
13.#
14.# This software is offered without warranty of any kind.
15.# Any use is at your own risk.
16.#
17.# Usage: remove-bg2.sh [name of file]
18.#
19.# The new image will be saved in the same directory as the
original,
20.# without any risk of overwriting the original file. Only ONE file
21.# at a time can be processed.
22.#
23.# This script will also function as a Thunar Custom Action. The
24.# command line for the custom action should be remove-bg2.sh %n.
25.# Under "Appearance Conditions," place a checkmark in front of
26.# "Image Files," and leave the file pattern set to *.
27.
28.# Select the image format you want to save the image as
29.EXT=$(zenity --list --column="Select One" --title="Remove
Background" --width=250 --height=250 --text="Select the image format
to\nsave the transparent image to:" png webp avif tiff)
30.# If the 'Cancel' button is selected, exit the script
31.    if [ $? == 1 ]; then
32.      exit
33.    fi
34.# Make sure the file extension is all lowercase text, and save it
35.declare -l EXT
36.EXT=$EXT
37.
38.# Enter the color name or hexadecimal equivalent. To see a list of
39.# acceptable color names and hexadecimal equivalents by visiting
40.# https://imagemagick.org/color/#color_names
41.# The color defaults to "white." The other commonly used color will
42.# most likely be "black."
43.COLOR=$(zenity --entry --title="Remove Background" --width=300 --
height=250 --text="Enter the color name you'd\nlike to change to
```

```
transparent:\n\n(Lowercase only!)" --entry-text="white")
44.# If the 'Cancel' button is selected, exit the script
45.if [ $? == 1 ]; then
46.   exit
47.fi
48.
49.FUZZ=$(zenity --entry --title="Remove Background" --width=300 --
height=250 --text="Enter the fuzz value you'd\nlike to use.
\n\nDefault: 10\nUse smaller numbers only large\nenough to achieve the
results\nyou are seeking!" --entry-text="10")
50.# If the 'Cancel' button is selected, exit the script
51.   if [ $? == 1 ]; then
52.      exit
53.   fi
54.
55.# Make sure the input file actually exists
56.if [ ! -e $1 ]; then
57.continue
58.fi
59.
60.# Strip away the file extension of the input file, and save it in
61.# the variable "name"
62.       name=$( echo $1 | cut -f1 -d.)
63.
64.# The ImageMagick "convert" command performs the creation of the
65.# image with transparency. This command is based on the solution
66.# presented by fmw42 on Stack Overflow, incorporating the
alteration
67.# offered by mobeen, along with some of my own tweaks.
68.# https://stackoverflow.com/questions/69851329/ \
69.# remove-backgroundany-color-from-image-using-image-magick
70.       convert $1 -alpha off -fuzz $FUZZ% -fill none -transparent
$COLOR -draw "alpha 0,0 floodfill" \
71.       \( +clone -alpha extract -blur 0x2 -level 50x100% \) \
72.       -alpha off -compose copy_opacity -composite \
73.       $name-fuzz-$FUZZ-transp.$EXT
74.
75.# Exit the script cleanly after the creation of the transparent
image
76.exit 0
```



A szkript az **1. sorában** a szokásos shebag-gal kezdődik (#! /bin/bash), amivel a bash szkriptek tipikusan indulnak. A következő 25 sor megjegyzéseket tartalmaz, és magától értetődőek.

A **29-36. sor** megjeleníti az első Zenity párbeszédpanelt, ami lehetővé teszi



a felhasználó számára, hogy kiválassza az átlátszó hátterű kép kimeneti formátumát. Ha a Zenity párbeszédpanelen a “Cancel” gombot használjuk, a szkript szépen kilép. Azért, hogy a dolgok igényeink szerint legyenek, gondoskodunk arról, hogy a kimeneti fájl kiterjesztése kisbetűs legyen.
A **43-47. sorban**, a második Zenity párbeszédpanel lehetővé teszi a felhasználónak, hogy megadja azt a színt, amit átlátszóságra szeretne

lecserélni. Az alapértelmezett érték „fehér”, de bármilyen színre módosítható, szükség szerint. Ha a ***színt*** névvel adod meg, ügyelj arra, hogy kisbetűs legyen. A név helyett megadható a hatjegyű hexadecimális RGB színkóddal is (pl. #FFFFFF fehér, vagy #000000 fekete). A színek nevét és azok hatjegyű hexadecimális RGB színmegjelöléseit egyaránt megtalálhatod az ImageMagick honlapján.

Valószínűleg a leggyakoribb a fehér (alapértelmezett) vagy a fekete szín. Ami a többi szín azonosítását illeti, egyszerűen nyisd meg a képet GIMP-ben, és a szemcseppentővel válaszd ki a színt. Ezután a GIMP előtér/háttér színváltójában kattints a színre. Olvasd le szín ott megadott hatjegyű hexadecimális kódját, és használd azt.

Jogos a kérdés, ha megnyitod a képet a GIMP-ben, hogy beazonosítsd a színt, akkor „miért ne használjam a GIMP-et szín átlátszóságra váltásához?” Nos, ha olyan vagy, mint én, talán még nem ismered a folyamatot (tudom, hogy még nem találtam ki teljesen). Tehát ahelyett, hogy a GIMP-ben botladoznánk és babrálnánk, megpróbálnánk kitalálni a

módját (és ***talán*** véletlenül ráérezni a szükséges lépésekre), csak vedd a színkódot, és térj vissza a szkripthez, ami elintézi neked. Kétségtelen, hogy a GIMP képes kezelni a feladatot, de a szkript gyorsabb.

Egy újabb Zenity beviteli párbeszédablakot hozunk létre a **49-53. sorban**, hogy a végfelhasználó megadhassa az „fuzz” (elmosás) szintjét (összehasonlításképpen a GIMP-ben „threshold”-nak nevezett értéket). A fuzz opció lehetővé teszi az adott szín megfelelő árnyalatának eltérését, ezzel segít megbizonyosodni arról, hogy nem maradtak kóbor pixelek az átlátszónak szánt területen. A szkriptben az alapértelmezett értéket 10% -ra állítottam. A 0% azt jelenti, hogy pontosan CSAK a megadott szín változik átlátszóvá. A 10%-os beállítás azt jelenti, hogy a megadott szín és a szín 10%-án belüli egyéb színek változnak átlátszóvá. Tapasztalatom szerint, a 10%-os beállítás a legtöbb esetben meglehetősen jól, az igényeimnek megfelelően működik, ezért választottam az „alap” fuzz szintnek. Nyugodtan játssz ezzel a beállítással, hogy elérd a várt eredményeket. Ne dőlj be a „késztetésnek”, hogy nagy számokat használj! Valószínűleg nem fog tetszeni az eredmény. A fuzz szintje tipikusan 4 és 10 százalék között van. Ha jóval 30 százalék fölé kerül a szint, az eredeti kép túl nagy része válik átlátszóvá ahhoz, hogy felismerhető vagy használható legyen.

Itt egy példa arra, hogy a különböző fuzz szintek hogyan hatnak a kimeneti

képre. A képet véletlenszerűen választottam ki, monokróm háttérrel. Valójában a kép a magazinban máshol is, az ICYMI cikkében megjelenik.

Az eredeti kép

Satheesh Sankaran képe a Pixabay-ről

Akkor itt egy kép a különböző „fuzz”-szintek bemutatására.

A szín megadásánál csak tippeltem (WAG-nak is szoktam hívni … wild a** guess - becslés hasraütéssel) és beírtam „blue” a színhez, amit lecserélnék átlátszóra. Az 5%-nál (balra fönt) látható, hogy a „kék” háttér egy része cserélődött le átlátszóra. 10%-os fuzz-szinten (jobbra fönt) kék háttér még további részletei lettek átlátszóra cserélve. 20%-os fuzz szintet használva (balra lent) majdnem az egész kék hátteret átlátszóra cseréltük, egy nagyon kis rész kivételével a „weight loss” szöveg alatt és a mérlegtől jobbra. 30%-os szintnél (jobbra lent) a teljes kék hátteret elintéztük, lecserélve azt átlátszóságra.

Csak az ilusztáció kedvéért, hogy miért maradj az eredmény eléréséhez szükséges lehető legalacsonyabb fuzz szintnél, íme ugyanaz a kép 50%-os fuzz szinttel.

Ahogy látható, átléptük azt a szintet, ahol csak a megadott háttérszínt alakítjuk át átlátszósággá. A kék egy része újra megjelenik és a mérleg sötétszürke lapja átlátszóvá vált. A sárga szöveg egy része is átlátszóvá lett, a jobb oldali mérleg szürkés területével együtt. Ezért fontos, hogy a kívánt eredmények elérése érdekében a lehető legkisebb fuzz szintet használd. Ha túl nagy a fuzz szint értéke, amivel indítasz, akkor lényegében visszalépsz, és kevésbé közelítesz ahhoz, amit kerestél. ***Még egy-két százalékos eltérés is óriási változást hozhat.*** Próbálj ki több fuzz szintet, hogy megtaláld azt, ami a szükséges eredményt adja. A fenti példákban a 30%-os fuzz szint működik, míg az 50%-os fuzz szint nem.

Az **56-58. sor** ellenőrzi még egyszer, hogy a bemeneti fájl valóban létezik-e. Ha igen, a szkript tovább fut. A **62. sor** leválasztja a fájlkiterjesztést a bemeneti fájlról, és a $name változóban tárolja.

A szkriptben a „kemény munkát” a **70-73. sor** végzi, meghívva az ImageMagick **convert** parancsát. A $FUZZ változót arra használjuk, hogy beszúrjuk a fuzz szintet, amiről az imént beszéltünk. A konvertálás parancs további részében a korábban megadott színt használjuk (-transparent $COLOR) annak beállítására, hogy melyik színt szeretnénk áttetszőre váltani. Ezután a $name változóba mentett bemeneti fájlnevet vesszük (kiterjesztése nélkül), és ahhoz csatoljuk a „-fuzz-” szót, a fuzz szintet ($FUZZ), valamint a „-transp” kifejezést, és megadjuk az új fájlkiterjesztést, amit az első Zenity párbeszédpanelen választottunk ki. Így az eredeti fájl nem lesz felülírva (jobb a biztonság, mint később sajnálkozni!). Ezen kívül, a különböző fuzz szintekkel készült képeket külön fájlokba menti, így nyomon követheted és összehasonlíthatod, hogy az egyes fuzz szintek mennyire működnek az adott képen. Ez az ImageMagick „convert” parancs az **fmw42** által a Stack Overflow-n bemutatott megoldásán alapul, beépítve a **mobeen** által javasolt módosítást, néhány saját módosításommal együtt.

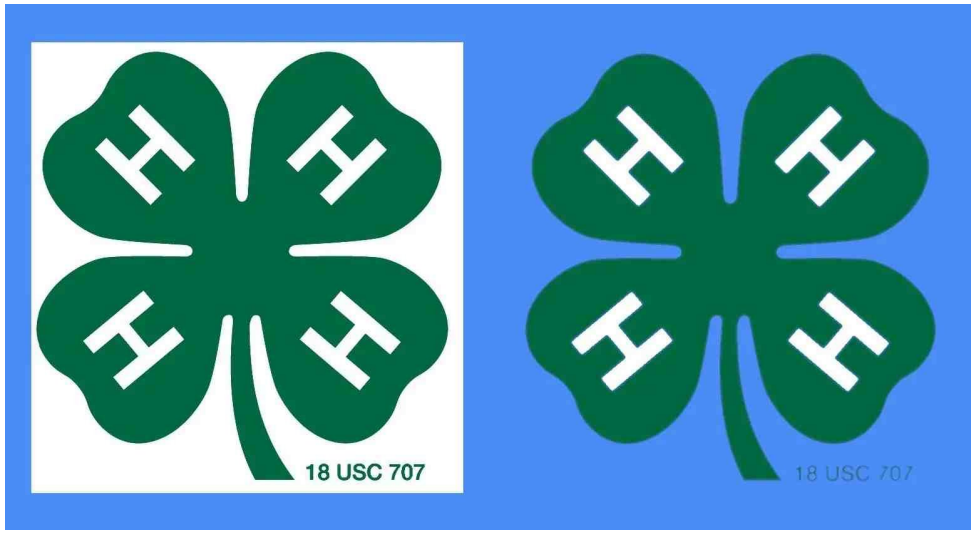
Végül, a **76. sorral** lehetővé tesszük a kilépést a szkriptből tisztán és elegánsan, a kép konvertálása után.

os fuzz szint kellett. Bármilyen más szint átlátszó „lyukakat” csinálna a mellkasi lemezén, a fogain és a szemfehérjén. Ha magasabb fuzz szintet tudtam volna használni, a karaktert körülölelő fehérség eltűnt volna (ahogy

Néhány további példa

Most nézzünk meg néhány további példát, amik közül néhány a cikk kedvéért készült.

Ezen az összetett képen bal oldalt a Pixabay-ről letöltött (a cikkben korábban használt) JPG kép van, a jobb oldalon pedig áttetsző hátterűvé alakított PNG. A képeket színátmenetes kék háttérre helyezték, hogy segítsék az átlátszóság megjelenítését. A bal oldali kép a JPG (és ne feledd, a JPG NEM támogatja az átláthatóságot), míg a jobb oldali kép a szkripttel való feldolgozás után létrehozott PNG. Véleményem szerint a szkript nagyon jó munkát végzett a képpel. Az eredeti JPG teljesen fehér hátterű, a PNG fájlnak pedig nincs háttere. Ez a kép kissé trükkös volt, és csak 1%-



a kezdeti próbálkozásaim során magasabb fuzz szinteken). Tehát mindaddig, amíg olyasvalamin használom fel a képet, aminél a háttér szintén fehér (például a PCLinuxOS Magazine oldalain), senki nem fogja észrevenni a fehér „fényudvart” a kép körül.
Meemaw ezzel, a szkript használatának első próbálkozásai során készült képpel járult hozzá. A JPG fájl a bal oldalon található, tömör fehér háttérrel. A jobb oldali, a szkript által létrehozott átlátszó kép. A fehér háttér eltávolításakor a lóhere levelein lévő „H” betűk is átlátszóvá lettek, de Meemaw a GIMP-be belépve, újra kitöltötte fehérrel. Az átlátszó kép létrehozásakor a „alapértelmezett” 10%-os fuzz szintet használta. Kék alapon mutatjuk, így jobban felismerhető a jobb oldali kép átlátszósága, szemben a bal oldali JPG egyöntetű hátterével.

A remove-bg2.sh használata

A remove-bg2.sh szkript csak egy parancssori argumentumot vesz fel, ez pedig annak a grafikus fájlnak a fájlneve, amihez átláthatóságot szeretnél hozzáadni. Ez lehet bármilyen, az ImageMagick által támogatott grafikus formátum (ami azt jelenti, hogy SOK különböző grafikus formátumot tud olvasni!). A lista Mode oszlopában az „R” jelzésűeket keresd.

Parancssorban valahogy így fog kinézni:

```
remove-bg2.sh [a_fájl_neve]
```

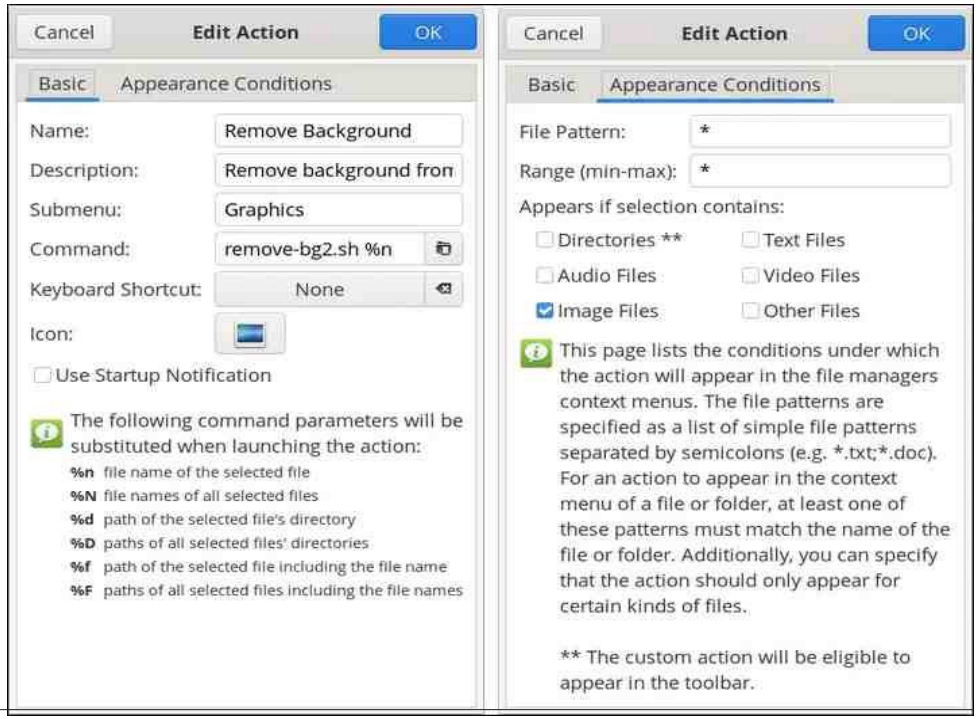
vagy

```
remove-bg2.sh [útvonal_és_a_fájl_neve]
```

Vedd figyelembe, hogy a szkript ***EGYSZERRE CSAK EGY FÁJLON TUD DOLGOZNI***! Ez addig igaz, amíg parancssorból, vagy Thunar egyéni műveleteként használod.

Biztosítania kell, hogy a fájlnevekben **NE** legyenek szóközök. A dolgok egyszerűsége érdekében nem adtam hozzá semmit a szkripthez, ami eltávolítaná a szóközöket a bemeneti fájlnevekből (megpróbáltam, és nem járt sikerrel... a fájlnév szóközeit nem elég a szkripten belül átírni, mivel a konvertálási parancs opciói a merevlemezen ténylegesen létező megadott képet használják, és nem a szkript memóriájában tárolt fájl megváltozott nevére, változóra keres... így egyetlen szokásos „trükköm sem működik). Szóköz a fájlnevekben a szkript hibáját **FOGJA** okozni, **tehát én szóltam**. És mivel olyan jól működik, úgy döntöttem (egyelőre), hogy nem megyek bele, és vacakolok mindenfélével, hogy a szóközökkel bíró fájlnevek ne okozzanak szkripthibát. A fájlnevekben szóköz használata egyébként is borzalmas ötlet, mivel a Linux parancssorban (és így a bash szkriptekben) túlteng, és NAGYON gyakran használják a parancssori opciók leírásának egymástól való elválasztására.

Alapvető szabályként, én igyekszem elkerülni a fájlnevekben a szóközöket, már többször visszaütött, amikor szóköz volt a fájlnévben. Van egy külön Thunar Egyéni műveletem, ami a fájlnevekben a közöket „-”-re, „.”-ra, vagy „_”-ra váltja (kötőjel, pont, aláhúzás), attól függően, hogy melyik műveletet választom. Vagyis, ha olyan fájlt töltök le, aminek a neve szóközt tartalmaz, azonnal eltávolítom azokat valamelyik egyéni műveletemmel. Ez olyasmi, amit rutinból csinálok.



A remove-bg2.sh használata Thunar egyéni műveletként

A „Műveletek létrehozása” párbeszédben, az „Alap” fül alatt írd be a Név mezőbe „Háttér eltávolítása”. A „Leírás” mezőbe én beírtam, hogy „Háttér eltávolítása képfájlból”. Ha az egyéni műveleteknél van almenüd, akkor írd be annak az almenünek a nevét, ahol az egyéni műveletet megjeleníteni akarod. A gépemen a grafikus egyéni műveletek a „Grafika” almenübe vannak csoportosítva. Ezután a „Parancs” mezőbe írd be, hogy „remove-bg2.sh %n”. Nem emlékszem, hogy valaha beállítottam volna gyorsbillentyűt bármelyik egyén művelethez … soha. Ellenben választok ikont, amit a menüben az egyéni művelet mellett megjelenít.

A „Megjelenési feltételek” fül alatt a „Fájlminta” mezőben hagyom a *-ot és a „Képfájlok” elé pipát teszek. Ismét mondom, emlékezz, a szkript úgy készült, hogy egy időben csal egy fájlon tud dolgozni (tekintettel a %n-re a parancs beviteli sorában).

Most már az egyéni műveleted használatra kész. Keresd meg a képet, kattints rá a jobb egérgombbal, és válaszd ki az egyéni műveletet. Kövesd a szkript utasításait, és saját képeket készíthetsz átlátszósággal. Még egyszer, tartsd észben, hogy a szkript ***egyszerre csak egyetlen fájlon*** történő munkára van tervezve (mivel a parancssorban a %n-t használzunk.)

Alternatívák

Persze dolgozhatsz GIMP-el, és próbálhatod megtanulni az átlátszó kép készítésének ottani folyamatát ... ha van időd rá. Nekem eddig nem volt.

Meemaw a kéthavonta megjelenő GIMP oktató anyagaiban még 2021-ben két külön cikket írt arról, hogyan lehet megszabadulni képek háttértől (vagyis átlátszó hátteret létrehozni). Ha van olyan kép, aminek átlátszó hátteret szeretnél, akkor a jól megírt oktatóanyagait követheted, hogy elérd a kívánt eredményt. Csak egy oktatóanyaga támaszkodik maszkok használatára. Valamiért lefagy az agyam, amikor GIMP-ben maszk használatára kerül sor, ezért én átugrom azt. Szerencsére három másik módot is kínál az átlátszó háttér létrehozására.

Vagy beugorsz a https://remove.bg weblapra, feltöltöd a képet, majd letöltöd onnan a képedet átlátszó háttérrel a számítógépedre. Noha ez a folyamat sokkal tovább tart, mint a szkript futtatása, de talán jobb eredményt ***érhetsz el,*** mint a szkripttel. Tudod, ez a webhely vagy mesterséges intelligenciát, vagy pár összetett algoritmust használ (ami JÓVAL túlmutat a kódolási képességeimen), hogy képet átlátszó háttérrel készítse el. Ennek folytán sokkal összetettebb képeknek tud átlátszó hátteret készíteni, mint amiket ez a szkript képes kezelni. De ne feledd, hogy ekkor fájljaidat valamilyen távoli webhelyre töltöd fel, ami mindig felveti a biztonság kérdését. A remove-bg2 szkripttel minden marad a számítógépeden.

Összegzés

Remélem, érdemesnek találod a szkriptet arra, hogy elhelyezd a grafikus eszközöket tartalmazó eszköztáradban. És be kell vallanom, hogy szórakoztató volt elkészíteni a szkriptet (igen, tudom... furcsa dolgok szórakoztatnak?). De az „élvezet” egy része az volt, hogy felfedeztem, mennyivel többre képes a szkript, mint ahogy eredetileg gondoltam. Kezdetben nem akartam belefoglalni a fuzz-szint állítása vagy megadása lehetőségét, csak az alapértelmezett 10%-os értéken hagytam, ahogy a szkriptben be volt kódolva. De miután hozzáadtam a képességet, hogy a fuzz szint testre szabható legyen, sokkal többre lett képes, mint azt valaha is el tudtam volna képzelni, és a szkript sokkal jobban működött, mint reméltem.