



AZP Paket Hazırlama Funksiyaları

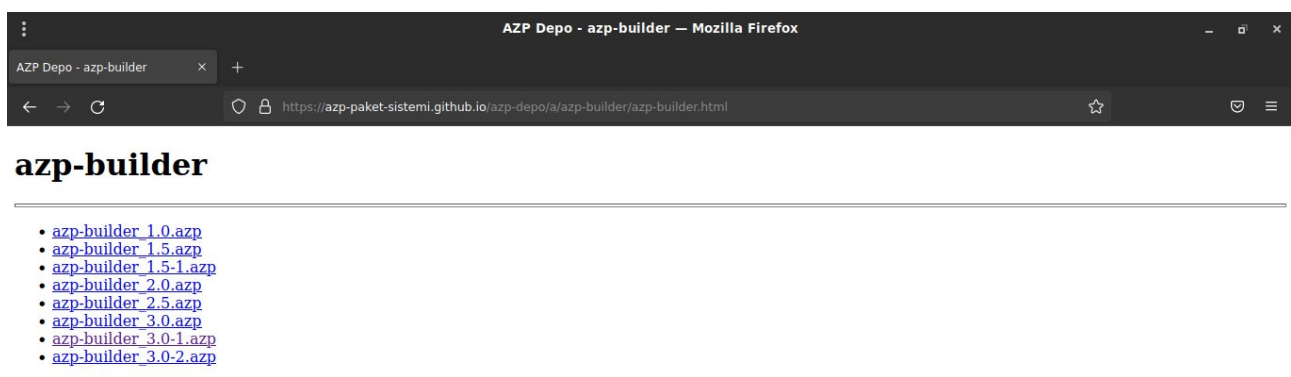
o AZP paketi necə hazırlanır ?

Azp üçün 3 paket növü (bunlardan bir dənəsi paket sayılmır, script'dir) vardır. Bunlardan istifadə edərək istədiyiniz növdə paket hazırlaya bilərsiniz.

Bunun üçün ilk öncə AZP Depo'sundan 'azp-builder' proqramı endirilib quraşdırılmalıdır.

Azp Builder linki:

<https://azp-paket-sistemi.github.io/azp-depo/a/azp-builder/azp-builder.html>



Ən altdaki son versiyayı yüklədikdən sonra terminala 'sudo azp -i azp-builder_(versiya).azp' daxil edib əmri həyata keçirin və proqramı quraşdırın.

Sonra, terminala 'azp-builder -h' daxil edin. Seçənlər qabağınıza çıxacaq:

```
turan@turan: ~  
Fayl Düzənlə Görünüş Axtar Terminal Kömək  
turan@turan:~$ azp-builder -h  
I: << AZP Builder -- 3.0-2 >>  
I: azp-builder -build [folder] == AZP paketi yığ  
I: azp-builder -build-azpsrc [folder] == .azpsrc mənbə paketi yığma  
I: azp-builder -create-data [folder] == 'azp_data' data faylı yaratmaq  
I: azp-builder -create-azpsrc [folder] == .azpsrc mənbə paketi yaratmaq  
I: azp-builder -create-azpsh [name] == .azpsh mənbə skripti yaratmaq  
I: azp-builder -create-azpweb [name] == .azpweb veb yükləmə faylı yaradır  
I: azp-builder -package-open [name] == .azp paketi çıxarır  
turan@turan:~$
```

AZP paket hazırlama növləri və funksiyaları:

- 1) -create-data = Bu funksiya normal azp paketi yığmaq üçün paketin ehtiyac duyacağı proqram məlumatlarının yazılacağı data faylını düzəldir.
- 2) -build = Bu funksiya normal .azp uzantılı paket hazırlamaq üçündür.
- 3) -build-azpsrc = Bu funksiya mənbə kodları (məsələn c, c++) açıq olan, 'compile' edilməmiş proqramları .azpsrc uzantısına paketləmək üçündür.
- 4) -create-azpsrc = Bu funksiya mənbə kodları açıq olan bir layihənin təməlini düzəltmək üçündür. Yəni, seçdiyiniz layihənin içinə əsas olan 'azpbuid' və digər faylları qoyur.
- 5) -create-azpsh = Bu funksiya .azpsh uzantılı mənbə kodu yığma skriptini düzəldir. Bu sadəcə bir skriptdir, hər hansı bir paket deyil. Bu script AZP tərəfindən oxunaraq paketi uyğun şəkildə hazırlayır.
- 6) -create-azpweb = Bu funksiya .azpweb uzantılı bir data faylı hazırlayır. Bunun içərisində siz uyğun yerlərə proqramın məlumatlarını və endiriləcəyi mənbəni daxil edirsiniz. Bu .azpweb data faylınında məqsədi budur. Başqa bir depoda olan paketin asan şəkildə yüklənib quraşdırılmasını təmin etmək.
- 7) -package-open = Seçilən bir AZP paketini çıxarır

AZP Builder ilə hər funksiyada paket hazırlama

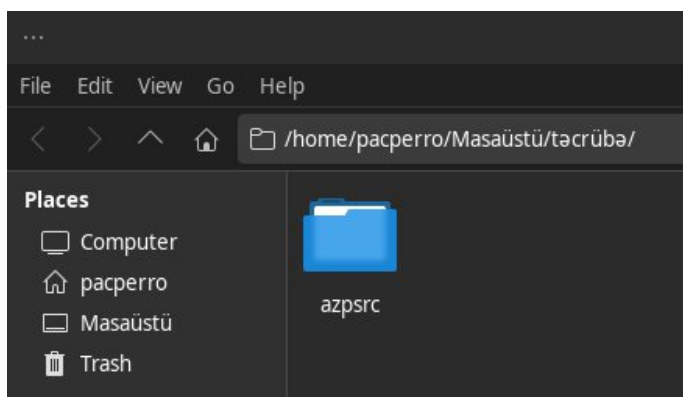
AZP Builder'də 3 paket hazırlama növü vardır. Bunlardan istifadə edərək tək-tək necə paket hazırlanacağı aşağıda qeyd olunmuşdur:

1) Normal AZP paketi hazırlama

İlk öncə, 'təcrübə' adlı bir qovluq yaradaq:

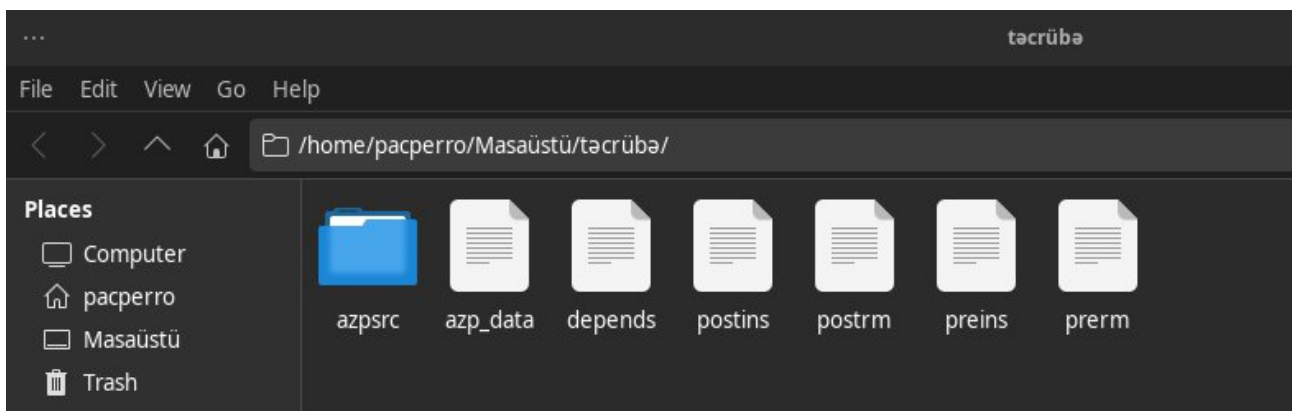


Bu qovluğun içində 'azpsrc' adlı bir qovluq daha yaradaq:



İndi isə, Masaüstündə bir terminal açaq və aşağıdakı kodları daxil edək:

```
$ azp-builder -create-data təcrübə
```



Beləliklə, qovluğun içində 'azp_data, postins, postrm, preins, prerm, depends' adlı fayllar yarandı.

Burada 'azp_data' faylından başqa digərlərini silə bilərsiniz. Ancaq bu faylların mənalərini bilmək lazımdır. Aşağıda bunların mənalrı yazılmışdır:

1) 'azp_data' = AZP paket məlumatlarını doldurmaq üçün data faylıdır

- 2) 'depends' = Proqram quraşdırılacağı zaman onun asılılıqlarını oxuyacaq və quraşdıracaq fayldır. Asılılıqlar alt-alta yazılmalıdır
- 3) 'postins' = Proqram quraşdırıldıqdan sonra icra edir. Bu kodlar hər proqramlaşdırma dili ilə yazıla bilər
- 4) 'postrm' = Proqram silindikdən sonra icra edilir
- 5) 'preins' = Proqram quraşdırılmazdan əvvəl icra edilir
- 6) 'prerm' = Proqram silinməzdən əvvəl icra edilir

İlk öncəliklə, azp_data faylını açıb doldurmağınız tələb olunur. Aşağıda bu data faylının içi göstərilmişdir:

```
...                               ~/Masaüstü/təcrübə/azp_data - Mousepad
File Edit Search View Document Help
1 name:
2 description:
3 version:
4 maintainer:
5 architecture:
6 description:
7
```

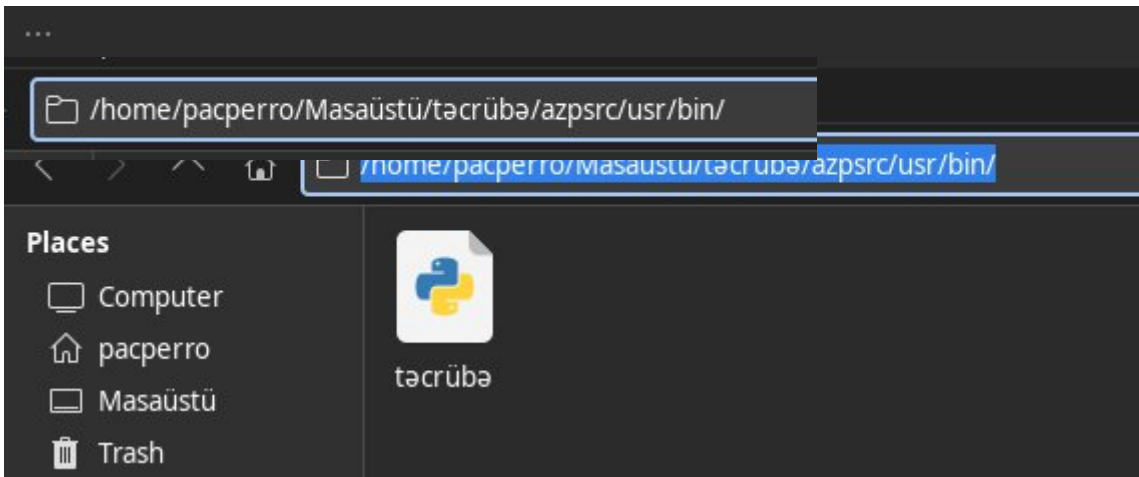
Bu faylda hər zaman 2 dəfə 'description:' yəni açıqlama qismi verilir. Əgər sizin proqram haqqında olan açıqlama mətniniz 1 sətirdən ibarətdirsə, bunu 2'ci sətirə yazmalısınız. Yox əgər proqram haqqında olan açıqlama 2 və daha artıq sətirdən ibarətdirsə o zaman 2'ci sətirdə olan açıqlama qisminə * işarəti qoyub, ən sondaki açıqlama qisminə açıqlama mətnini kopyalamağınız vacibdir.

Proqramınız, hər cür prosesorlu kompyuterdə çalışırsa o zaman 'architecture:' qismini silə bilərsiniz. Proqramınız xüsusi olaraq 32 və ya 64 bit prosesorlu kompyuterlərə uyğundursa o zaman bu qismə 64 bitdirsə 'amd64 və ya x86_64', 32 bitdirsə 'i686' yazmağınız vacibdir.

'depends, postins, postrm, preins, prerm' kimi faylları doldurmağınız məcburi deyildir, proqramınıza görə doldurub doldurmamağınız sizə qalmışdır. Sadəcə, 'depends' adlı faylda yazacağınız proqram isimləri depoda mövcud olmalıdır.

AZP paketinə proqram fayllarını yerləşdirmə

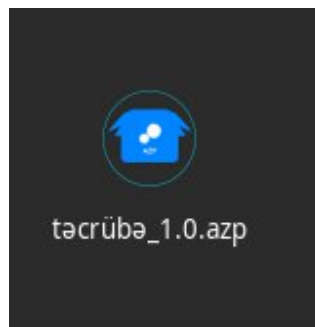
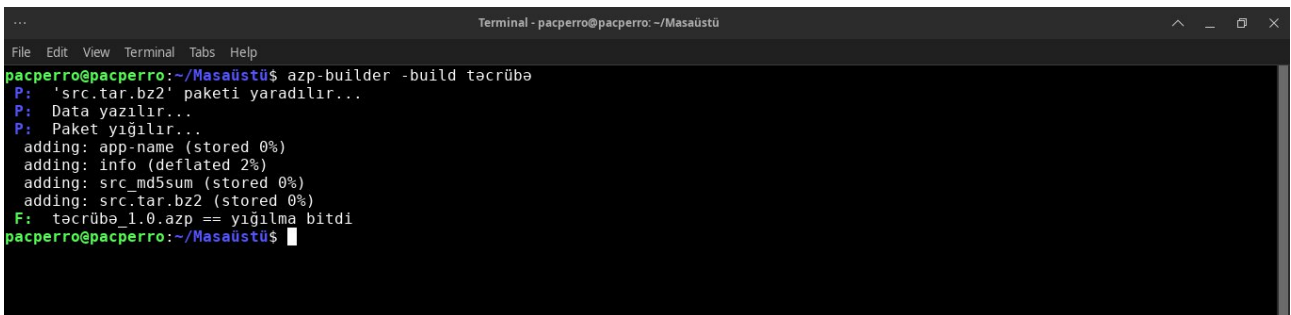
İlk öncə deyək ki, 'azpsrc' adlı əsas qovluğun içində hansı qovluqları qoysanız, fayl sisteminə o kopyalanacaq. Siz /usr/bin/ qovluğu içərisinə 'təcrübə' adlı bir python kodunuzu kopyalamaq istəsəniz, bu zaman 'azpsrc' içərisində 'usr' sonra da 'bin' qovluğu onunda içində təcrübə adlı python proqramınızı yazsanız, paket fayl sisteminə sizin proqramınız yerləşdirəcəkdir.



Uyğun
şəkildə
'azpsrc'
içərisinə
proqram
ınızın

fayllarını yerləşdirdikdən sonra terminalda aşağıdakı əməlləri icra edirsiniz:

```
$ azp-builder -build təcrübə
```



Gördüyünüz kimi, paketiniz hazırdır. Artıq bu paketi 'azp -i' kommandından istifadə edərək quraşdırma bilərsiniz.

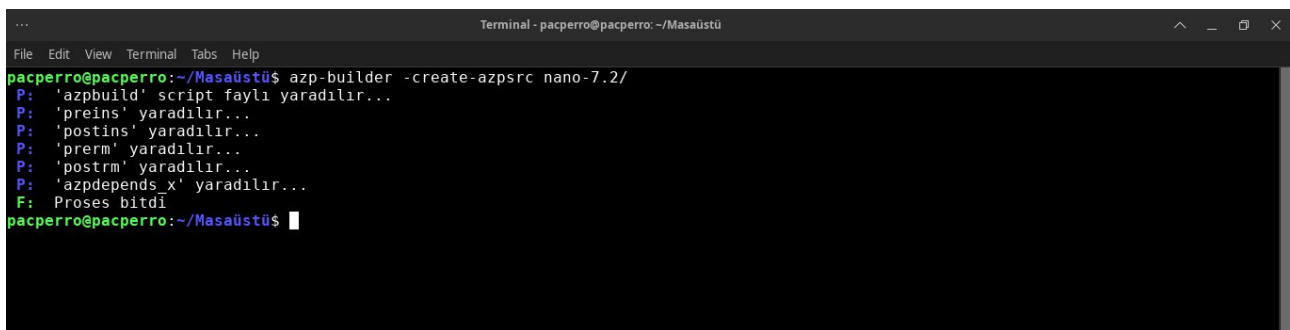
2) Mənbə kodlu AZP paketi hazırlama

Bu proses, əsasən Linux'da proqram 'compile' etməyi bacaran developerlər üçün nəzərdə tutulmuşdur.

Məsələn, əlimizdə bir 'nano-7.2.tar.xz' adlı bir paket var. Bu paket 'make' əmri vasitəsilə yığılmalıdır. Bu işi avtomatikleşdirmək və sürətləndirmək üçün AZP'nin .azpsrc adlı paket növündən və .azpsh adlı skript faylı növündən istifadə edə bilərsiniz. İlk yazıda .azpsrc ilə necə paket hazırlanacağı haqqında məlumat veriləcəkdir.

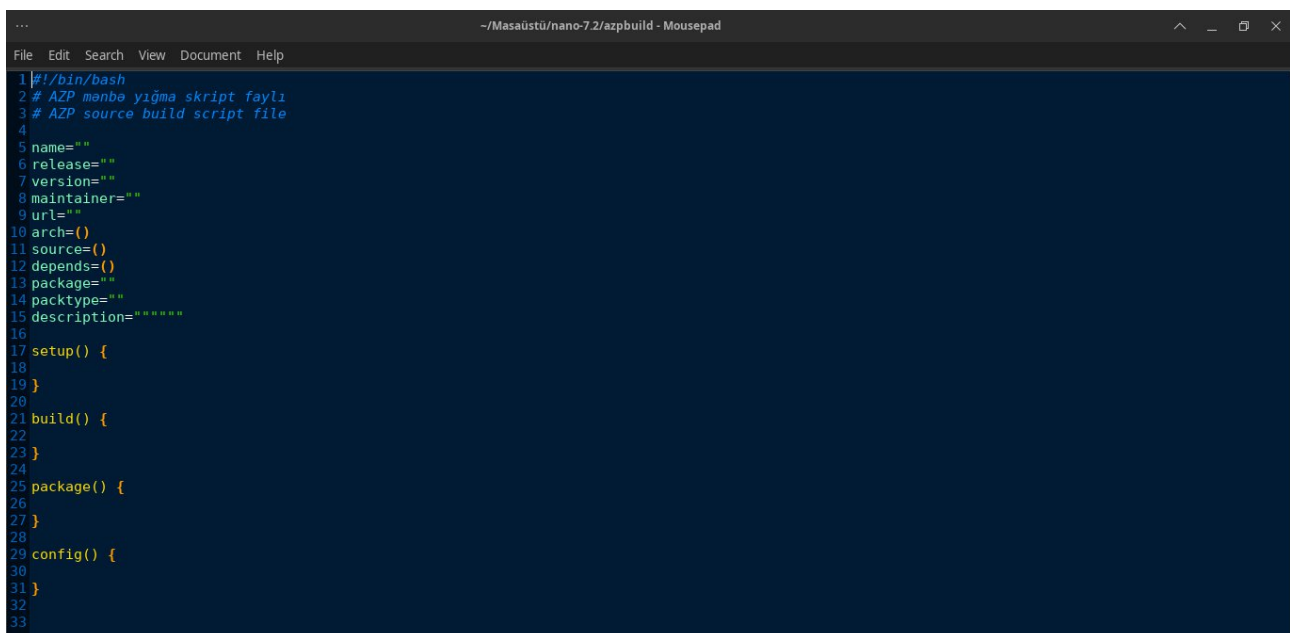
İlk öncə, paketimizin adına uyğun olaraq Masaüstündə 'nano-7.2' adlı bir qovluq yaradıb terminal açaraq aşağıdakı əmri yerinə gətiririk:

```
$ azp-builder -create-azpsrc nano-7.2
```



```
Terminal - pacperro@pacperro: ~/Masaüstü
pacperro@pacperro:~/Masaüstü$ azp-builder -create-azpsrc nano-7.2/
P: 'azpbuid' script faylı yaradılır...
P: 'preins' yaradılır...
P: 'postins' yaradılır...
P: 'prerm' yaradılır...
P: 'postrm' yaradılır...
P: 'azpdepends_x' yaradılır...
F: Proses bitdi
pacperro@pacperro:~/Masaüstü$
```

Qovluğun içərisində şəkildə gördüyünüz fayllar yaradılacaqdır. 'azpbuid' skriptindən başqa digər faylları silə bilərsiniz, bu əsas skriptdir.

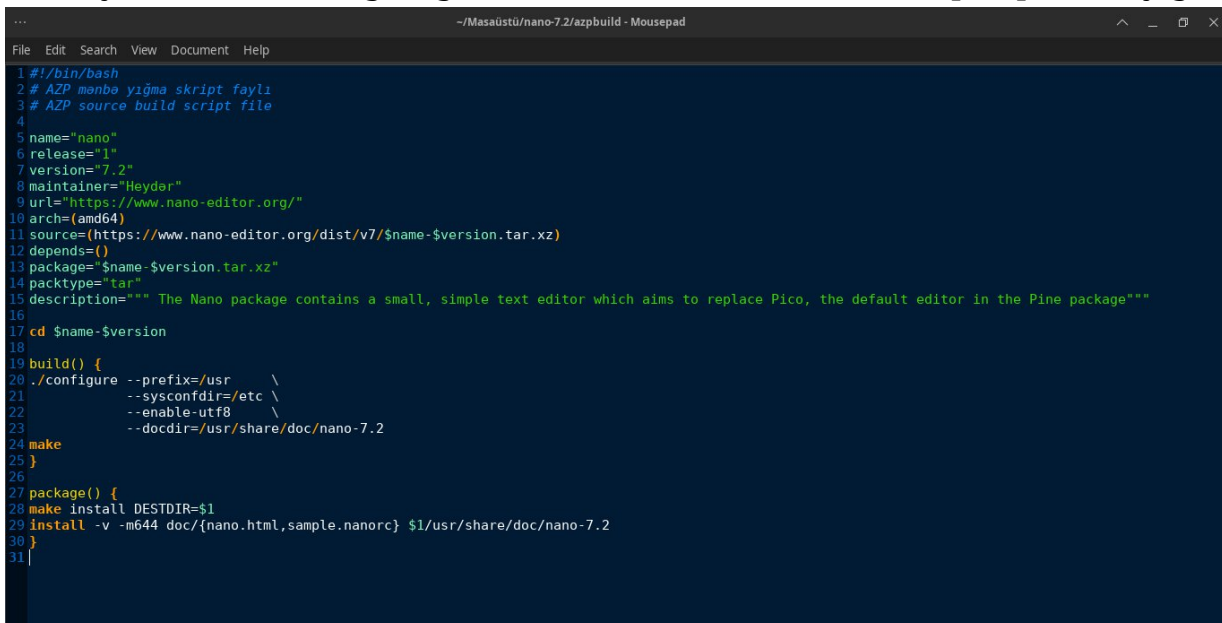


```
~/Masaüstü/nano-7.2/azpbuid - Mousepad
File Edit Search View Document Help
1#!/bin/bash
2# AZP mənbə yığma skript faylı
3# AZP source build script file
4
5name=""
6release=""
7version=""
8maintainer=""
9url=""
10arch=()
11source=()
12depends=()
13package=""
14packtype=""
15description=""
16
17setup() {
18
19}
20
21build() {
22
23}
24
25package() {
26
27}
28
29config() {
30
31}
32
33
```

Skriptin içərisi belədir. Uyğun olaraq program məlumatlarını skriptə yerləşdirin. 'package' qisminə yüklənəcək paketin adını, 'packtype' qisminə paketin tipini qeyd edəcəksiniz. Əgər paket zip tiplidirsə 'zip' olaraq, yox tar tiplidirsə o zaman 'tar' olaraq qeyd edəcəksiniz.

Skriptəki 'depends' olan hissəyə bağımlılıqlar yazılır. Bu bağımlılıqlar quraşdırılmaq üçün layihə qovluğunun içindəki 'azpdepends_x' adlı skriptdən istifadə edir. Bu skriptin 'x' olan hissəsini dəyişdirib, bağımlılığın adını yazırsınız, 'azpdepends' hissəsinə toxunulmur. Əgər birdən çox bağımlılıq varsa o zaman bu skriptdən bağımlılıq sayı qədər kopyalayıb eyni prosesi icra edirsiniz. Bağımlılıq skriptinin kodları 'azpbuild' skripti ilə eynidir, onu doldurduğunuz şəkildə bu skriptidə doldurursunuz.

Verilmiş nümunə şəkilə diqqət etdiysəniz, ordada setup() və config() funksiyalarının olmadığını görürsünüz. Bunun səbəbi, nano programını yığarkən



```
... ~/Masautti/nano-7.2/azpbuild - Mousepad
File Edit Search View Document Help
1 #!/bin/bash
2 # AZP mənbə yığma skript faylı
3 # AZP source build script file
4
5 name="nano"
6 release="1"
7 version="7.2"
8 maintainer="Heyder"
9 url="https://www.nano-editor.org/"
10 arch=(amd64)
11 source=(https://www.nano-editor.org/dist/v7/$name-$version.tar.xz)
12 depends=()
13 package="$name-$version.tar.xz"
14 packtype="tar"
15 description=""" The Nano package contains a small, simple text editor which aims to replace Pico, the default editor in the Pine package""
16
17 cd $name-$version
18
19 build() {
20 ./configure --prefix=/usr \
21 --sysconfdir=/etc \
22 --enable-utf8 \
23 --docdir=/usr/share/doc/nano-7.2
24 make
25 }
26
27 package() {
28 make install DESTDIR=$1
29 install -v -m644 doc/{nano.html,sample.nanorc} $1/usr/share/doc/nano-7.2
30 }
31
```

bu funksiyalara ehtiyac olmamasıdır. Normalda 'azpbuild' skriptində bu funksiyalarda gəlir, lazım deyilsə onları silə bilərsiniz. Amma build() və package() funksiyaları mütləq olmalıdır, onları silmək olmaz. Skriptdəki 'depends' hissəsinə də əgər program asılılıqlara ehtiyac duymursa boş qoya bilərsiniz.

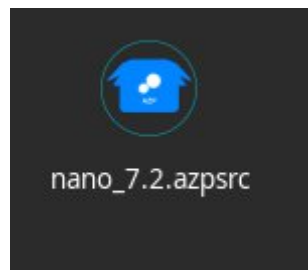
Məlumat: Skriptdə 'cd \$name-\$version' əmrini gördünüz. Bu əmr, AZP paketi parçaladıqdan sonra kodların yığıldığı qovluğun içinə girmək üçündür. Məsələn; 'nano-7.2.tar.gz' paketi parçalananda, 'nano-7.2' adlı qovluq yaranır və bütün kodlarda onun içində olduğu üçün skript onun içinə girməlidir amma AZP avtomatik bu işi etmir. Bunun səbəbi, AZP paketi parçalayanda kodların yığıldığı qovluğun paketin adı ilə bəzən eyni olmamasıdır. AZP bunu təyin edə bilməz, buna görə içinə girilib proses aparılacaq qovluğun adını siz skriptə qeyd etməlisiniz.

Siz program yığarkən mənbə paketini layihə qovluğunun içinə qoya bilərsiniz. Qoymasanızda, skriptdə qeyd etdiyiniz paketin yükləmə linkinə görə paket yüklənəcəkdir. Qoysanız, paketin yüklənməsinə ehtiyac olmayacaqdır.

Bütün işləri bitirdikdən sonra aşağıdakı əmri icra edərək paketi yığaq:

```
$ azp-builder -build-azpsrc nano-7.2
```

```
Terminal - pacperro@pacperro: ~/Masaüstü
pacperro@pacperro:~/Masaüstü$ azp-builder -build-azpsrc nano-7.2
P: Paket yığılır...
  adding: azpbuilt (deflated 41%)
F: nano_7.2.azpsrc == yığılma bitdi
pacperro@pacperro:~/Masaüstü$
```



Artıq mənbə kodu paketi hazırdır. Paketi quraşdırmaq üçün:

```
$ sudo azp -azpsrc-build
```

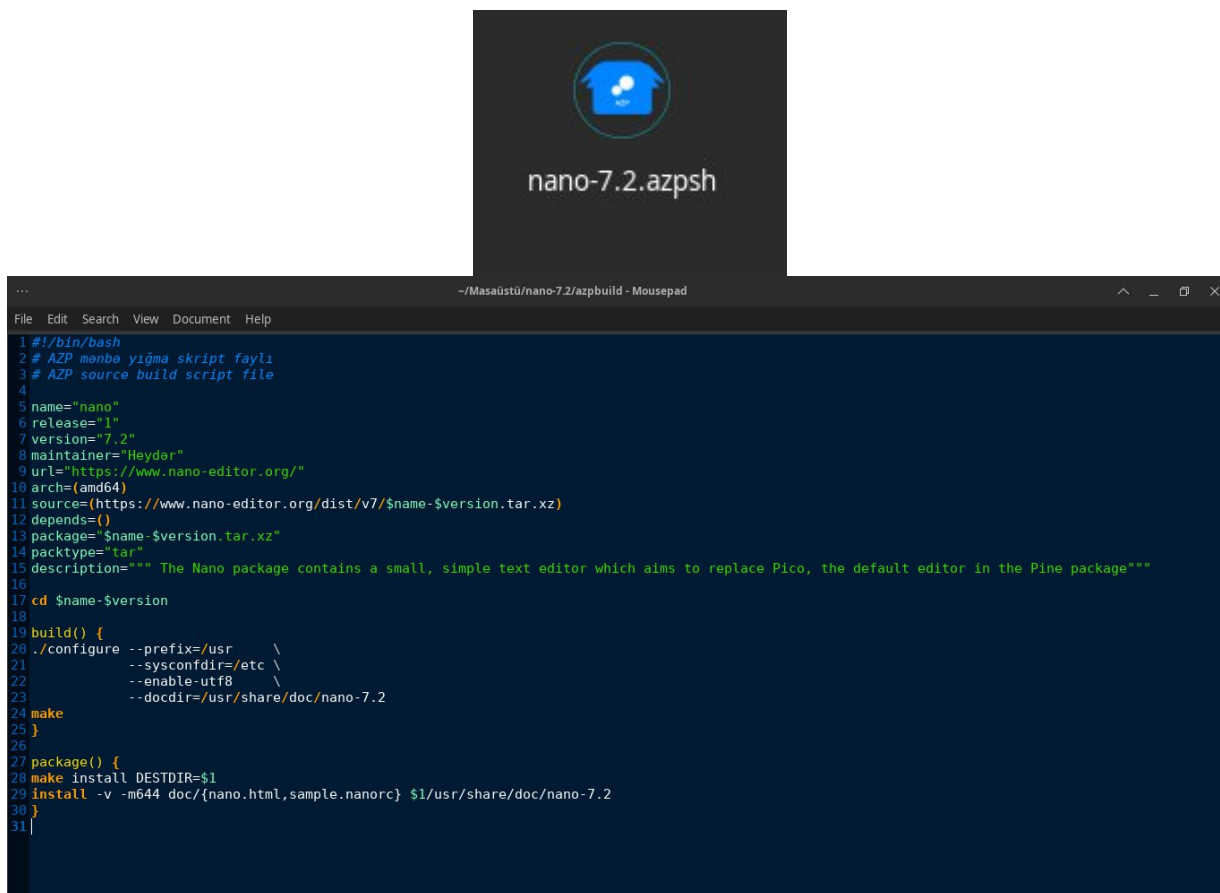
3) Mənbə kodlu AZP skripti hazırlama

Bu funksiyanın .azpsrc paketindən fərqli olaraq ən yaxşı üstünlüyü paket yox, skript halında olmasıdır. .azpsh skripti, .azpsrc'nin istifadə etdiyi 'azpbuilt' skriptindən fərqlənmir. Yuxarıda .azpsrc ilə yığmaq istədiyimiz paketi indi .azpsh skripti istifadə edərək yığmağa çalışacağıq.

Aşağıdakı əmrdən istifadə edərək 'nano-7.2' adlı bir .azpsh skripti yaradaq:

```
$ azp-builder -create-azpsh nano-7.2
```

```
Terminal - pacperro@pacperro: ~/Masaüstü
pacperro@pacperro:~/Masaüstü$ azp-builder -create-azpsh nano-7.2
P: nano-7.2.azpsh == skript faylı yaradılır...
F: Proses bitdi
pacperro@pacperro:~/Masaüstü$
```

Əlimizdə bir 'nano-7.2.azpsh' adlı skript var. Bu skript, 'azpbuild' skripti ilə eynidir. O, necə doldurulursa bu skriptdə eynən doldurulmalıdır.

Əlavə hər hansı bir prosesə ehtiyac yoxdur, skript hazırdır. Lakin AZP'də .azpsh skriptlərini birbaşa quraşdırılmır, avtomatik olaraq .azp paketinə çevirlərək istifadəçiyə verir.

```
$ sudo azp -azpsh-build nano-7.2.azpsh
```

Amma avtomatik quraşdırılmasını istəyirsiniz, o zaman aşağıdakı əmri icra edin

```
$ sudo azp -azpsh-build nano-7.2.azpsh -ins
```

4) .azpweb data faylı yaratma

.azpweb data faylı, fərqli bir mənbədə yerləşən proqramın qısaca yüklənib quraşdırılmasını təmin edir. Məsələn bir, 'neofetch' proqramının bu data faylı ilə qısaca yüklənib quraşdırılmasını təmin etmək istəyirik. Bu data faylını hazırlamaq üçün terminal açıb aşağıdakı kodları daxil edirsiniz:

```
$ azp-builder -create-azpweb neofetch
```

```
turan@turan-hp-probook-450-g1: ~  
Fayl Düzəlt Görünüş Axtar Terminal Yardım  
turan@turan-hp-probook-450-g1:~$ azp-builder -create-azpweb neofetch  
P: neofetch.azpweb == web faylı yaradılır...  
F: Proses bitdi  
turan@turan-hp-probook-450-g1:~$
```

Daha sonra bu data faylını açırıq:

```
~/neofetch.azpweb - Mousepad  
File Edit Search View Document Help  
1 <?xml version="1.0" encoding="UTF-8"?>  
2 <!-- AZP veb yükləmə -->  
3 <!-- AZP web download -->  
4 <azp-web>  
5   <name></name>  
6   <description></description>  
7   <version></version>  
8   <package></package>  
9   <size></size>  
10  <source></source>  
11  <architecture></architecture>  
12 </azp-web>  
13  
14
```

Daha sonra, burada gördüyünüz hissələrə uyğun məlumatları doldurmalısınız. `package` bölümünə bu proqramın tam paket adını daxil etməlisiniz. `size` bölümünə də proqramın ölçüsünü yazmalısınız. Ölçünü yazarkən, əgər proqram ölçüsü “KB” ilədirsə “K”, “MB” ilədirsə “M”, “GB” ilədirsə “G” yazmalısınız (şərt deyil). `source` bölümünə də paketin yüklənəcək linkini yazacaqsınız.

Məsələn:

```
~/neofetch.azpweb - Mousepad  
File Edit Search View Document Help  
1 <?xml version="1.0" encoding="UTF-8"?>  
2 <!-- AZP veb yükləmə -->  
3 <!-- AZP web download -->  
4 <azp-web>  
5   <name>neofetch</name>  
6   <description>Shows Linux System Information with Distribution Logo</description>  
7   <version>7.1.0-3</version>  
8   <package>neofetch_7.1.0-3-3.azp</package>  
9   <size>92K</size>  
10  <source>https://azp-paket-sistemi.github.io/azp-depo/n/neofetch/neofetch_7.1.0-3-3.azp</source>  
11  <architecture>amd64</architecture>  
12 </azp-web>  
13  
14
```

Daha sonra bu faylı qeyd edirsiniz. Məlumatları düzgün daxil etdiyinizdən əmin olun. Proses bura qədərdir və sadədir. Daha sonra bu data faylını aşağıdakı əmr ilə istifadə edə bilərik:

```
$ sudo azp -azpweb neofetch.azpweb
```

```
turan@turan-hp-probook-450-g1: ~  
Fayl Düzəlt Görünüş Axtar Terminal Yardım  
turan@turan-hp-probook-450-g1:~$ sudo azp -azpweb neofetch.azpweb  
[sudo] password for turan:  
P: .azpweb faylı oxunur...  
-----  
I: Ad: neofetch  
I: Açıqlama: Shows Linux System Information with Distribution Logo  
I: Versiya: 7.1.0-3-3  
I: Paket: neofetch_7.1.0-3-3.azp  
I: Ölçü: 92K  
I: Mənbə: https://azp-paket-sistemi.github.io/azp-depo/n/neofetch/neofetch_7.1.0-3-3.azp  
I: Arxitektura: amd64  
I: Paket Novu: .azp  
-----  
Bu paketin quraşdırılmasını istəyirsiniz ? [h/y]: h  
P: Paket yüklənir...  
neofetch_7.1.0-3-3.azp 100%[=====] 89,18K 212KB/s in 0,4s  
P: Quraşdırılacaq paket: neofetch_7.1.0-3-3.azp  
-----  
P: Fayllar çıxarılır...  
P: 'src_md5sum' təhlükəsizliyi yoxlanılır...  
P: Fayllar fayl sisteminə kopyalanır...  
F: Quraşdırma bitmişdir  
turan@turan-hp-probook-450-g1:~$
```

Və, nəticədə neofetch proqramı quraşdırılmış olur

5) Bir AZP Paketi sökmə

Əgər hər hansı bir AZP Paketini redaktə etmək üçün sökmək istəyirsiniz, aşağıdakı əmərdən istifadə edə bilərsiniz:

```
$ sudo azp -package-open paket.azp
```

```
turan@turan: ~/Masaüstü  
Fayl Düzənlə Görünüş Axtar Terminal Kömək  
turan@turan:~/Masaüstü$ azp-builder -package-open paket_1.0.azp  
P: Paket çıxarılır...  
P: 'src' mənbə paketi çıxarılır...  
P: Köhnə fayllar silinir...  
F: Paket açıldı  
turan@turan:~/Masaüstü$
```



Gördüyünüz kimi, paket açıldı. Paket içərisinə daxil olaraq artıq bu paket üzərində düzəlişlər edib paketi yığaraq yenidən istifadə edə bilərsiniz