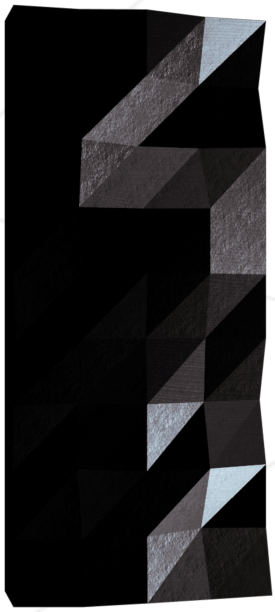

Proiect UNIX I

42sh

Staff Academy+Plus contact@academyplus.ro

Sumar: Momentul asteptat a sosit. Asezati celebrul 42sh!

Cuprins

I Preambul **2**

I.1 How to Get into Orbit . . . 2

I.2 Steps . . . 3

I.2.1 Step 0 - Rocket Design . . . 3

I.2.2 Step 1 - Launch Prep . . . 3

I.2.3 Step 2 - The Launch . . . 3

I.2.4 Step 3 - Get up to 10,000 meters . . . 4

I.2.5 Step 4 - Gravity Turn 45 degrees East until 70km Apoapsis . . . 4

I.2.6 Step 5 - Get Your Apoapsis above 70km . . . 4

I.2.7 Step 6 - Orient for On-Orbit Burn . . . 4

I.2.8 Step 7 - Burn into Orbit . . . 4

I.3 Finishing word . . . 5

II Introducere **6**

III Partea obligatoire **8**

IV Partiea optionala **9**

V Instructiuni **11**

Capitolul I

Preambul

I.1 How to Get into Orbit

Getting into orbit over Kerbin is rather simple, but it will require some knowledge and preparation. Space begins at 70,000m above the planet Kerbin. Stay above that for an entire flight around the planet and you're in orbit. The process to get into orbit follows a simple progression:

- Launch straight up to 10km
- Change your Pitch to 45° east and keep the engines on until your projected Apoapsis is above 70km
- Change your pitch to horizontal just before you reach the Apoapsis
- Burn at Apoapsis until your Periapsis is above 70km

Specifications:

- Length: 15-20 minutes
- Difficulty: Harder than a suborbital flight, easier than an orbital intercept.
- Skills needed: Seat of the pants
- For version: Every version (tested on 0.23)

I.2 Steps

I.2.1 Step 0 - Rocket Design

A liquid fueled rocket with at least two stages preferably. Anything less will either only get you suborbital or an unwieldy expensive super large fuel tank thats only good for being an orbiting billboard. The cheapest orbiter you can build with the current stock game (0.22) is:

- Unmanned: Probodobodyne OKTO2 or Manned: Command Pod Mk1 with Mk16 Parachute and TR-18A Stack Decoupler
- FL-T400 Fuel Tank
- LV-909 Liquid Fuel Engine
- TR-18A Stack Decoupler
- FL-T800 Fuel Tank
- LV-T30 Liquid Fuel Engine

For fun, use the LVT-30 engine for your first stage as it doesn't have thrust vectoring and thus will challenge you to actually fly the craft into orbit using your WASD keys keeping your navball on target. Your control module by default provides enough SAS control by itself to prevent you from going out of control for this mission. Make sure your staging sequence is the way you want it, else you can add your flight to the countless list of catastrophic mission failures.

I.2.2 Step 1 - Launch Prep

Prepare your launch.

- Set your map view with the m key to see your rocket at the launch site from space, tilted so you are looking north. You want to be able to see your apoapsis marker during your gravity turn so you can gauge when to cut your engines and coast up to it prior to burning your orbiting maneuver.
- Set your thrust to maximum by holding Shift.
- Toggle on SAS by hitting t.

I.2.3 Step 2 - The Launch

Say your countdown if you wish, and hit spacebar to launch into...well, space...without the bar (you'll make a space station with a bar at this end of the galaxy later).

I.2.4 Step 3 - Get up to 10,000 meters

Keep your rocket pointed straight up (use your navball to keep your dot on the top dot on the blue part of the navball) until you hit around 9,900 meters. Your first engine should cut out before this, just jettison it with the spacebar and burn your final engine. Throttle your second engine down to 2/3rds power since the atmosphere is weaker here and won't slow you down as much, and you will save precious fuel.

I.2.5 Step 4 - Gravity Turn 45 degrees East until 70km Apoapsis

Now for the fun part. Assuming you are pointing straight up, hit the d key to turn your ship using your navball until your dot is able to slide left and right on the 90 degree east line. Then tip on that line down toward the ground until you hit the 45 degree east mark. Hit your t key for SAS to help you keep it there, but don't rely on it. You will need to make course corrections until you get past 70km and enter space. At this point you should be going into space at a 45 degree angle from the ground, and in an easterly direction. Doing so will gain you speed and not waste the fuel you will need to get into orbit.

I.2.6 Step 5 - Get Your Apoapsis above 70km

While climbing up to 70km or 70,000 meters, it's now time to switch to your map view and control your ship from there entirely. Hopefully your navball is toggled on, else click on the collapsed tab at the bottom of the map view. You will need to keep burning your fuel until you see your apoapsis marker reach 70km. You can see how high it is by hovering your mouse over it. Once it hits 70km (I usually shoot for 75km to buy me some head room) your craft will be able to coast up to it without any further fuel. Feel free to cut your engines with the x key and save some fuel for your orbital burn.

I.2.7 Step 6 - Orient for On-Orbit Burn

As you approach apoapsis (preferably before 30 seconds prior) orient your ship to the 0 degree latitudinal mark, heading east. Again, you should be in your map view when you start your burn. Hit tab to center your view on Kerbin and zoom out so you can see your orbit as it forms.

I.2.8 Step 7 - Burn into Orbit

Once you are 10-30 seconds away from your apoapsis, begin your on-orbit burn using the Shift key to throttle up. Full throttle is best at lower altitude apoapsises since you don't want to burn past apoapsis lest you waste precious return fuel if that's your intent. If you aren't concerned, go full throttle baby at any point near apoapsis and claim a stake with the stars. You only need to burn your fuel long enough until you see a periapsis marker appear on the other side of the orbit from the apoapsis, and you see a full orbit circle, then cut your engine with the x key. Congrats, you made it into orbit. It's usually a good idea to keep burning your fuel until your new periapsis marker is above 70km. If it's below, then your orbit will cause your craft to aerobrake, eventually returning to Kerbin. If your periapsis is ever above 70km, congrats, you will orbit Kerbin forever. If

you have a manned flight, and fuel is low, then only burn until your periapsis is below 70km to ensure a safe return to Kerbin.

I.3 Finishing word

After orbiting for a while, depending on your fuel (with this design you won't have much if any at all) you can orient for a deorbit burn by burning backwards in the direction of your travel, once you are at your apoapsis point for maximum efficiency lest you be stuck in space with a manned crew forever.

Capitolul II

Introducere

Pe parcursul scolarizarii, exista momente ce marcheaza elevii pentru totdeauna. `42sh` este unul dintre aceste momente. Realizarea acestui proiect reprezinta o etapa importanta la `42`.

Este vorba sa scrieti un shell `UNIX`, cel mai stabil si complet posibil. Cunoasteti deja numeroase shell-uri fiecare are propriile caracteristici, de la banalul `sh` prezent pe toate distibutiile `UNIX` din lume, la foarte complexul si completul `zsh` pe care il utilizeaza atat de multi fara sa stie de ce. Bineinteles exista numeroase alte shell-uri, precum `bash`, `csh`, `tcsh`, `ksh`, `ash`, etc. Va este dat ca `42sh` sa fie primul vostru shell cu adevarat; se intampla des printre elevi sa vrea sa-si aleaga un shell de referinta pentru a incerca sa-i reproduca comportamentul. Este o idee buna, dar cu conditia sa va alegeti shell-ul propriu in cunostinta de cauza. Intr-adevar, cei dintre voi care vor alege `zsh` ca shell de referinta se angajeaza intr-o cautare lunga si anevoioasa, chiar daca demersul e foarte bogat in invataminte. De exemplu ei vor putea invata umilinta si munca sisifica.

Cea mai buna maniera de abordare a shell-ului de referinta consta pur si simplu in a incerca de mai multe ori si a analiza diferentele, ce sunt de multe ori subtile si uneori distorsionate. Dar nu uitati ca, in cazul in care `sh` este considerat, prea des ca fiind un shell prea “simplu”, a-l rescrie (recoda) intro maniera stabila este un rezultat mult mai interesant, decat un shell ce promite rezultate fulminante, dar care se dovedeste incapabil sa faca mai mult de redirectari si pipe-uri.

Cuvantul cheie in acest caz este “stabilitate”. Un `42sh` este neinsemnat, dar valoreaza intotdeauna mai mult ca un `42sh` ce ofera toate optiunile imaginabile, dar care nu are nicio valoare daca esueaza intr-o situatie neprevazuta. Asigurati-va deci ca livrati un `42sh` stabil; nu voi reusi niciodata sa va conving suficient in legatura cu asta.

Proiectul este compus din 2 parti descrise mai jos si este usor diferit de alte subiecte cu care sunteti obisnuiti.

- O parte **obligatorie**, de realizat in mod imperativ.
- O parte **optionala**, care nu va fi luata in considerare doar daca partea obligatorie functioneaza integral.

Ati remarcat ca partea obligatorie nu permite validarea proiectului, va fi nevoie sa alegeti si sa implementati mai multe puncte bonus pentru validare. Asta din doua motive: primul este ca partea obligatorie e foarte simpla, iar al doilea este pentru a obliga la selectarea functionalitatilor partii optionale si a va stabili propriile obiective in realizarea acestui proiect.

Un ultim amanunt, utilizarea facila a `42sh` -ului vostru va fi luata in considerare. Se presupune ca un utilizator a unui shell obisnuit ca `sh` sau `bash` sa fie capabil sa utilizeze si `42sh` vostru intr-un mod intuitiv ...

Capitolul III

Partea obligatoire

- O linie de comanda minimala.
    - Afisarea unui prompt.
    - Citirea liniei de comanda fara editarea acesteia.
    - Gestiunea corecta a spatiilor (albe) si a tabulatorului.
- Builtin-urile urmatoare, cu toate optiunile daca au:
    - `cd`
    - `echo`
    - `exit`
    - `env`
    - `setenv`
    - `unsetenv`
- Executarea de comenzi simple cu ajutorul parametrilor si gestiunea lui `PATH`.
- Gestiunea erorilor si a valorilor de retur in urma comenzilor.
- Urmatorii operatorii de redirectare: `">"`, `">>"`, `"<"` si `"|"`.
- Operatorii logici `"&&"` si `"||"`.
- Operatorul `";"`.

Capitolul IV

Partiea optionala

Aeasta parte va ajuta sa invatati partea teoretica a majoritatii punctelor. Ea este, dupa cum spuneam, la latitudinea voastra; puteti face aproape tot ce va doriti, in ordinea in care doriti. Din nou, stabilitatea va fi mult mai importanta decat cantitatea. Nu includeti optiuni ce pot pune probleme programului. Oricum, amintiti-va ca se pot obtine puncte in aceasta sectiune, doar daca partea obligatorie este completa, complet functionala si stabila.

- Semnele: `"""` (double quote), `"'"` (simple quote) si `"\"` (backslash).
- Precum si: `"*"`, `"?"`, `"[]"`, `"{}"` etc. Daca nu utilizati functia `glob(3)`, veti avea mai multe puncte.
- `"‘"` (back quotes).
- Sub shell-urile cu operatori `"()"`.
- Variabilele locale si builtin-urile `unset` si `export`.
- Istoricul comenzilor si builtin-urile `history` si `"!"` cu toate optiunile daca exista.
- Editarea unei linii asa cum se cere in `ft_sh3`.
- Descriptorii de fisiere si builtin-ul `read` cu toate optiunile sale.
- Completarea dinamica (autocomplete).
- Controlul Job si builtin-urile `job`, `fg` si `bg`, si operatorul `"&"` (foarte apreciat de baremul de corectare).
- Scrit-ul shell script (foarte apreciat de asemenea).

Acest document va va fi probabil foarte util, maiales in sectiunea 2.10.

Capitolul V

Instructiuni

- Acest proiect va fi corectat de catre oameni. Sunteti deci, liberi sa va organizati si sa va denumiti fisierele dupa cum doriti, respectand constrangerile de mai jos.
- Executabilul trebuie sa se numeasca `42sh`
- Trebuie sa livrati un Makefile cu toate regulile uzuale.
- Daca doriti sa utilizai `libft`, trebuie sa livrati sursele in directorul numit `libft` din radacina repository-ului vostru, impreuna cu un Makefile pentru compilare. Nicio versiune compilata deja de biblioteca voastra nu va fi acceptata la sustinere. `Makefile`-ul `42sh` vostru trebuie, bineinteles, sa se compileze si link-edita cu `libft`-ul vostru.
- Proiectul trebuie sa fie scris conform Normei.
- Va trebui sa te ocupi de erori intr-un mod corespunzator. In niciun caz programul nu trebuie sa se termine intr-un mod neasteptat (Eroare de segmentare etc...).
- Terminalul nu trebuie afiseze vreodata ceva de tipul `termcaps`.
- Va trebui sa livrati, in radacina repository-ului vostru, un fisier `auteur` continand login-urile personale, cate unul pe linie, astfel:

```
$>cat -e auteur
xlogin$
ylogin$
zlogin$
alogin$
$>
```

- aveti dreptul sa utilizati urmatoarele functii:
    - Toata sectiunea 2, man-urile
    - `malloc(3)`
    - `free(3)`

    - Toate functiile de biblioteca `termcaps`.

- Puteti adresati intrebarile pe forum, pe jabber, IRC, ...
- Succes tuturor!