Introduction to Operating Systems
Computer Architecture and Technology Department

Faculty of Computer Science

UPV/EHU

The Wizard of OS: Final report

Development of a text adventure game to learn the use of a terminal.

Mikel Aristu
David Cuenca
Leire Estanga
Zdravko Todorov

eman ta zabal zazu

Universidad del País Vasco
Euskal Herriko Unibertsitatea

May 17, 2021

Contents

1. Introduction

This final report talks about the project we have done in the subject of Introduction to Operating Systems. This project has been developed in C programming language, which is a language that provides constructs that map efficiently to typical machine instructions.

The main goal was to put into practice the theoretical knowledge acquired during the course, learning how the Linux Operating System works and using some system calls provided by the OS. So in order to put into practice all that acquired knowledge, we needed to create a text-based adventure game in which a player moves through some directories with the aim of reaching somewhere. This way, the player will learn about some Linux OS basic commands.

So we got to work. Firstly, we collected some ideas to decide what our game was going to be about. We had some different ideas, but in the end, we decided to create a game based on *The Wizard Of Oz*, without many doubts, calling it *The Wizard Of OS*. All the text of the game is available in Appendix A, as well as all the command's specification and verification documents in Appendix B.
The file tree can be consulted at any moment in Appendix C.

1.1 Questions

What audience is the game aimed at?

This game has its aim in people who want to learn how a terminal works. Also for those who already have some experience. It will help to understand different commands, their functions, and how to travel through directories, as there are a lot of them in the game. It is not really recommended for kids, but more for teenagers and older people, because the game has some explicit words. Disclaimer: Is not indented to hurt anyone's feelings by the used words, they are for fictitious purpose only.

What do you expect people who play will learn?

We expect that people will learn different Linux terminal commands and how to use them, how to view different files and edit them via the terminal, or to get to different directories, in which working directory are we at the moment... In order to learn all this, we use a story and make it more entertaining.

How is the game played?

The games design is based on text and interaction with the player. When you press the Enter button, a line prints, until a command needs to be used or the player must give an answer. If a wrong answer is given or a command is used incorrectly, a warning appears. After three errors the player will lose the game and she will have to start again. There are different colours used in the design so the directories can be recognised easily. The principal path is the yellow one (referencing *The Wizard Of Oz* thematic) but there are some secondary missions in the blue path. The journey is not too long, but has different obstacles on it. The game is won when the player finishes the complete journey, arriving to Emerald City and killing The Most Evil Bad Witch.

How hard was to design and program the game? What problems did you had?

Being honest, the path of the whole game development, took us through a lot of different problems, errors, etc., forcing us to learn a lot of new things, in order to face those obstacles. The main problems that challenged us, we had them when programming the commands, but researching and thinking about the errors/warnings we had, we achieved to solve them. Another problem we had, was an incompatibility between compilers' versions, so that we needed to redesign the script created to auto-compile the whole game with its commands,library as well as all the signals.

What did you learn while creating the game?

We really can not mention all the knowledge we acquired, as we have learned a lot. But as a teaser, we now manage how to use a variety of system calls, we know the principles of the usage of pipes to ”connect” two commands, we know exactly what is behind the most used Linux OS terminal commands, as we have implemented many of them, we now know how a correct error catching has to be done in order to avoid system crashes, apart from all the new things we have learned about the Linux Operating System and the C programming language. All this new knowledge will be really useful in our path of becoming computer engineers/scientists. We now feel really confident reading some code written in C language as well as making some operations in the Linux terminal.

1.2 Game plot

The user is dropped into the world of the Terminal along with her files, from the graphical shell, without having too much idea and accompanied by his dog Toto.

A good witch appears and will give her a little introduction: she will teach her how to use the first commands (`cd, ls`), she will give her the patent leather shoes to continue advancing in the adventure and she will explain that to return to his parents, the player will have to follow the yellow path to reach the ADMIN, that is the one who will help him to get out of this world. (Actually the power to get out of the world will be achieved thanks to the patent leather shoes that this good witch has given her). She will also introduce her to the `man` command, in case at any point she does not know how to use `X` command. She also will demand the player to take the dog with her, since *"strange things are happening lately"*, so she will bring the dog with her with the command `mv`. Apart from that, the `help` command will be available at any moment to check all the available commands.

Here, the journey to ADMIN begins! (Note: in APPENDIX C, a directory tree map is given for a full understanding of the paths).

At first, the player will be dropped in the village, where all mentioned before will happen. The Most Evil Bad Witch will be introduced here too.

Then, the player will only be able to access the Grove (`cd grove`), where she will meet the Scarecrow. It will teach her how to introduce her name into the Terminal world by using the `stee` command. It will ask her where she is at that precise moment with the command `pwd`, it will explain that it wants to have a brain and it will ask her to get to the ADMIN looking for its brain. In return, it will give her the following clue to move forward, since it knows all the paths in the area.

The player will now have two alternative paths. These will be the Forest Entrance (following the yellow path) and the optional path (the blue one) to the Haunted House.

If she chooses to follow the blue path, arriving at the Haunted House, the usual `ls` and `cd` commands will be used to inspect the house. There are some rooms to be discovered inside, such as the Living Room, here, an option will be given to the player to be able to tie Toto using the link command. Also, some other characters will appear in the house, like the ghost Casper. In the Basement, a text file is found, which has to be opened using `cat`. The bedroom contains a password needed later to enter the Emerald City.

She will continue through the forest entrance, where she will find the Tinman. This guy is a super intelligent man but has no heart, so with some options that are given, the player must convince Tinman to come with her. There are three options, one makes Tinman not to come with her and with the other two, she will get his help. It could be really helpful to have him with you. Each option has different obstacles to overcome, one is to help him to move by using oil using `touch` and the other is by demonstrating your intelligence using the `grep` command

and count words, once they are overcome, the player must continue its journey, going to the Trees or into the Forest.

In the Trees, we will find some food and help to beat the most Evil Bad Witch. Otherwise, if we go into the Forest we will find the Lion. This lion is coward, so he wants back his bravery, because the "kings of the jungle" are the lions so that it wouldn not be accepted by his family. For that, it asks help from the player, so it will continue the journey with them. To do that, the player will use the `cp` and `cat` command, she "cats" a text, she read the motivational text and then she copies it with `cp` in the lion so it can be brave enough to continue the journey.

They will continue on the yellow path until they arrive to the Emerald City. This is the place where the ADMIN is. To enter here a password is needed, which is given in the Haunted house, so if we want to continue with the journey we must get this password.

After entering the Emerald City, our next stop is the Prairie, where we will find The Most Evil Bad Witch. We must defeat her using the `kill` command so she does not bother us anymore and save the world OS from her. Finally we will meet with ADMIN in the castle, who is going to give us our desires and we can end the game.

2. APPENDIXES

2.1 APPENDIX A

2.1.1 CHAPTER 1: Village

Text:

-Good Witch: Hello to this strange world called Terminal! You have just arrived from your real world to this one, because the Most Evil Bad Witch wants to get those fantastic shoes to destroy your home, your fantastic and loved world. You must continue the YELLOW path to arrive at the ADMIN to be able to destroy the Most Evil Bad Witch, and when you get that, you will be able to get your most admired desire, to be able to get back to your admired world. To start your trip, you can use the command `cd` in order to move around different places and also you can use `ls` to watch which places are available. The use of `cd` is `cd LOCATION`, where `LOCATION` is a place adjacent to the player's current location. You and your Dog will have a long journey to the ADMIN, remember if you want the Dog to be always with you you have to use the command move to each place you wanna go. If you need to see which commands are available at any moment you can use the help command.

-The Most Evil Bad Witch: If you write any command in an incorrect way, I will destroy your appreciated world. Take a lot of care writing those commands, MUAJAJAJA!

-Good Witch: Remember the terminal is dark and full of terrors.

2.1.2 CHAPTER 2: Grove

Text:

-Scarecrow: Hi! Who are you? What 's your name? You can use the command `stee` to edit the file that will learn your name.

-Scarecrow: Hello [Name of the player], I'm Scarecrow. I suppose that you are trying to get to the ADMIN as you are in the YELLOW path. My dream is to get a brain, I know that ADMIN could get me one, so I will help you. But first we will have to know where we are, I don't have any idea, because of my little problem called no brain, could you help me? The good witch one day told me something about `pwd` I am not sure what it is about.

-Scarecrow: Where are we now? (Use command `pwd`)

-Scarecrow: If we are in the grove now, the next place to go could be the haunted house or the Forest entrance.

2.1.3 CHAPTER 2.1: Haunted house *(Secondary)*

Text:

-Scarecrow: This place doesn't seem really friendly, it's quite scary.

-Dog: wof wof wof wof wofff wof wof wof wooooof wof wof!

-Scarecrow: Yes let's take a look, we can use commands known before to get it.

2.1.3.1 Bedroom

Text:

-Scarecrow: This seems like a quite big bed, let's take a nap!

-Dog: wof wof woof wof wof wof!

-Scarecrow: No, don't fall into temptations, we must continue with our journey. Wait! What is this?

-Player: It seems like something suspicious, let's get a look.

-Scarecrow: We don't want The Most Evil Bad Witch to appear, so be careful please.

-Player: These files are encrypted so we must use something to see this text. Uhm

[Password is obtained using `getpass` command]

-Player: What is this?

-Scarecrow: I have no idea, just let's continue and try to get out from here.

2.1.3.2 Kitchen

Text:

-Scarecrow: this kitchen is a disaster, someone must have destroyed it.

2.1.3.3 Living Room

Text:

-Scarecrow: Look! This is a dog's leash, we can use to tie it! In that way, we don't need the `mv` command anymore! The owner of the house before must have been a lover of animals.

-Dog: woof!

[The dog is linked with `link`]

2.1.3.4 Bathroom

Text:

-Casper: BOO!

-Scarecrow: AAAAAAAAAHHHHHHH! It's a ghost!

-Casper: AAAAAAAAHHHHHH! It's a Scarecrow!

-Scarecrow: OK, that was scary, and it's supposed that I'm the one who has to scare.

-Casper: Yes, but you are supposed to scare crows, no ghosts duh.

-Player: Who are you? What are you doing here?

-Casper: I'm the real and original owner of this beautiful house.

-Scarecrow: Quite destroyed right now...

-Casper: True. Oh, what a cute puppy! I love Dogs, maybe you can find something really useful in the living room.

-Scarecrow: Thank you!

2.1.3.5 Basement

Text:

-Dog: Wof, wof.

-Player: This place it's really dark, I can't see anything.

-Scarecrow: Here's the interruptor for the light.

-Player: So, what are you waiting for to switch on the lights?

-Player: Now it's better.

-Scarecrow: Look, there's something there.

-Player: True, we can use ls to look at what it is.

[`ls` is used to find the text]

basement_text:

Hello adventurers:

This is a long journey, and even that, you decided to continue. It's not going to be easy, and there are going to be several obstacles to overcome. But be brave, don't give up and you'll get it. It's not easy to get to the ADMIN, it's even harder to enter to the Emerald city, so in order to help you I want to remember, you little adventurer, that the game you are playing it's called The Wizard of OS and the number 73 it's always really special for everyone. Programming languages don't really use capital letters and separation it's not used in attributes. So make friends and free this world from the bad witch.

Your little secret admirer.

Have a nice trip!

This text will be a clue for guessing the password to enter into the emerald city.

2.1.4 CHAPTER 3: Forest entrance

Text:

-Scarecrow: What the hell is that!

-Tinman: I am Tin Man, I was created by some people that used me as an experiment. Actually, in their words, I'm a piece of tin that has no use and no importance.

-Scarecrow: Oh, that must have hurted.

-Tinman: Not really, nothing can hurt me, I'm made of tin and i have no heart, so some people say that I'm indestructible.

-Dog: *Tries to bit he Tinman*

-Tinman: Hey buddy, you can't hurt me but that's not gentle.

-Player: We are trying to get to the ADMIN to ask our desires, you could come with us to get your heart.

-Tinman: Is it that even possible?... I have calculated the percentage of possibilities of achieving that aim and it's less than 0.0001%. The Most Evil Bad Witch will find us before getting there. I think that having a heart will just make me weaker and easier to beat. I told you I'm almost indestructible like this.

[options to choose what the player would say, the player must convince Tinman to come with her]

OPTIONS:

a) You're right, The Most Evil Bad Witch is horrible, but being a human is worse. You are weak and it's easier to kill us. But we can feel, love, trust and become better people so we achieve our objectives and enjoy life, you are indestructible but not immortal, you'll die one day, why not spice up your life a bit?

b) That's a way of seeing it but if you think about it, you could have friends if you get a heart and that's the happiest thing humans have in life. Not loving someone can be tough and quite sad, we can help you to achieve that.

c) You are completely wrong, if you don't have a heart, you can't have a soul and you'll never have a dream or something that inspires you. So you can be something more, you can have loved people not just information that you store, it can help you to be better and learn more about the world you live in. I'm from another world and as you are my new dear friend I will be happy to teach you about my life in that world called Earth. More colourful and cheerful than the terminal.

IF CHOOSING OPTION a)

-Tinman: It makes sense but are you sure that if I start a journey with you I will get a heart?
-Scarecrow: No but we need your help!!
-Tinman: Then you are going to help me if you want me to come with you. As you see, it's not easy for me to move, so I need oil, you can use one command to help me. You can have some in your inventory.
[The command `touch` is used to create oil]

-Tinman: Thank you! Now it's easier for me to move, and we can go to our next stop. We can have some break to eat in the trees, next to here or we could go into the Forest, which has a lot of horrors.

IF CHOOSING OPTION b)

-Tinman: I don't need friends nor help and I'm not interested in helping anyone. Less in loving someone, love is what makes you weak, I'm sorry, well no, you don't care me, I'm not coming with you. So as you have learned, you can go to your next stop. You can have a break to eat in the trees, next to here or we could go into the Forest, which has a lot of horrors.

IF CHOOSING OPTION c)

-Tinman: that sounds interesting, what could you teach me about your world that could be interesting for me?
-Player: We also have Dogs but they are not in pixels, and curveines exist.
-Tinman: what are pixels and curves?
-Player: If you come with us I'll explain it to you.
-Tinman: OK, I'm coming with you but you must prove that you are able to be as intelligent as me. As you see I have this marvellous book called Silvia Plath I need you to find and copy some information about it. What you have to find is daddy, and count how many times the word is repeated.
[`grep` command is used to count the number of occurences of a word]

POEM TO COUNT REPEATED WORDS:
[word to find: daddy, found 5 times, excluding the title]

---

daddy
by Silvia Plath

you do not do, you do not do
any more, black shoe
in which i have lived like a foot
for thirty years, poor and white,
barely daring to breathe or achoo.

daddy, i have had to kill you.
you died before i had time
marble-heavy, a bag full of god,
ghastly statue with one gray toe
big as a frisco seal

and a head in the freakish atlantic
where it pours bean green over blue
in the waters off the beautiful nauset.
i used to pray to recover you.
ach, du.

in the german tongue, in the polish town
scraped flat by the roller
of wars, wars, wars.
but the name of the town is common.
my polack friend

says there are a dozen or two.
so i never could tell where you
put your foot, your root,
i never could talk to you.
the tongue stuck in my jaw.

it stuck in a barb wire snare.
ich, ich, ich, ich,
i could hardly speak.
i thought every german was you.
and the language obscene

an engine, an engine,
chuffing me off like a jew.
a jew to dachau, auschwitz, belsen.

i began to talk like a jew.
i think i may well be a jew.

the snows of the tyrol, the clear beer of vienna
are not very pure or true.
with my gypsy ancestress and my weird luck
and my taroc pack and my taroc pack
i may be a bit of a jew.

i have always been scared of you,
with your luftwaffe, your gobbledygoo.
and your neat mustache
and your aryan eye, bright blue.
panzer-man, panzer-man, o you

not god but a swastika
so black no sky could squeak through.
every woman adores a fascist,
the boot in the face, the brute
brute heart of a brute like you.

you stand at the blackboard, daddy,
in the picture i have of you,
a cleft in your chin instead of your foot
but no less a devil for that, no not
any less the black man who

bit my pretty red heart in two.
i was ten when they buried you.
at twenty i tried to die
and get back, back, back to you.
i thought even the bones would do.

but they pulled me out of the sack,
and they stuck me together with glue.
and then i knew what to do.
i made a model of you,
a man in black with a meinkampf look

and a love of the rack and the screw.
and i said i do, i do.
so daddy, i'm finally through.
the black telephone's off at the root,
the voices just can't worm through.

if i've killed one man, i've killed two
the vampire who said he was you
and drank my blood for a year,
seven years, if you want to know.
daddy, you can lie back now.

there's a stake in your fat black heart
and the villagers never liked you.
they are dancing and stamping on you.
they always knew it was you.
daddy, pappy, you bastard, i'm through.

-Tinman: Perfect! what you have just found it's very useful for me. now we can go to our next stop. We can have some break to eat in the trees, next to here or we could go into the Forest, which has a lot of horrors.

2.1.5 CHAPTER 3.1: Trees *(Secondary)*

Text:

-Tree1: What do you want? What did you come for? You are disturbing, please leave and let us take the sun.

-Scarecrow: We are sorry tree, we wanted some food, we are sturbing.

-Tree2: As my friend said **DO NOT DISTURB US!** We are sunbathing. Please, be silent.

-Scarecrow: Look there something written there. We can use some command to read it.

-Tinman: I think that it's similar to the main word but I don't remember how to spell it. It says something about kill, let's try with that!

-Tree2: Shhhhh

-Player: It's better if we stay silent, let's look for the text you found.

[The `man` command is used to read the manual of a command, this could help to understand how to kill the witch]

-Scarecrow: This could help us defeat The Most Evil Bad Witch.

-Player: ok perfect but I'm starving, I wanna eat what we can do?

-Tree1: Pleeeeeeease, stay quiet I beg you, if you want to eat just take an apple from me, just stay silent. You can use touch to eat.

[The touch command is used to create an apple]

-Player: Thank you mister Tree!

-Tree2: Yes, yes, now leave! Enter the forest and never come back.

2.1.6 CHAPTER 4: Forest

Text:

-Dog: Wof wof woof wof wof wof!

-Lion: Aaaaaah! There's people here. Leave, go, leave me, don't hurt me please.

-Player: Why would we hurt you?

-Lion: I don't know, you look scary.

-Scarecrow: But you are a Lion you must be the one scaring us, aren't you?

-Lion: Of course, but I'm the Coward Lion, I would love to be braver but it doesn't matter how hard I try it's impossible for me. I'm never gonna be brave.

-Scarecrow: You could get back your bravery with the ADMIN, but we need you, you must help us defeat The Most Evil Bad Witch!!!

-Tinman: The Lion is too cowardly, but if we try we can become brave enough to help us beat The Most Evil Bad Witch. There is a text called braver, you need to use the right command to print it.

[The `cat` command is used]

-Lion: That's empowering, if only I could internalize that, everything could be easier and I could help you. There must be a Lion file, maybe you need to create it, but if you figure out which command to use I may help you defeat The most evil bad witch.

[The `cp` command is used]

-Lion: you don't get a sense of how much that could help me. Next stop Emerald city.

2.1.7 CHAPTER 5: Emerald City

Text:

-Guardian: Hello, this is the entrance to Emerald city. What do you want?

-Scarecrow: We want to visit the ADMIN.

-Guardian: Do you have an appointment?

-Player: No? How can we get one as fast as possible for today?

-Guardian: Sorry but he is really busy. He doesn't have time to see you for a year.

-Scarecrow: That's a lot of time!

-Lion: But there's got to be another way to visit him.

-Player: We can try to climb the wall.

-Lion: Don't you see it? That's too high.

-Guardian: And full of weapons and traps.

-Player: OK. What we have to do is to visit the city, we want to see the ADMIN. We don't wanna be camping outside here for a year.

-Guardian: You need a password for that. A really hard password to guess. You have 3 tries, if you don't get it, you'll have to start back again.

[The password is written, the password is `wizardofos73`]

(if they don't write the password right)

-Guardian: Maybe you didn't do your homework well and you really need to start your path again because you haven't discovered some places yet.

-Guardian: Now you can start your visit to the Emerald City, I recommend you to start with the prairie.

2.1.8 CHAPTER 6: Prairie

Text:

-Tinman: Player, that was really awesome! Now we can try to find the ADMIN. Let's begin.

-Player: Yes! So now we can start finding the ADMIN.

-Scarecrow: So the password was that, I thought we were going to wait for real. You really have a brilliant brain. I hope I can get one.

-The Most Evil Bad Witch: Welcome to my territory! JAJAJAJAJAJA! These all are my lands. You didn't arrive at the best place.

-Dog: wof wooof, wof wof wooof wof wof.

-Lion: Scarecrow please help me, i'm scared.

-Player: You... your name describes you perfectly..- you're the worst! You're a complete torment for everyone who meets you. So that's why we will defeat you.

-The Most Evil Bad Witch: Are you sure? You're a child who doesn't know anything about this black world, you know nothing, even though you're looking for the ADMIN to help you, you'll never arrive there, you're nothing.

-Lion: That's not a true, Player helped me!

-The Most Evil Bad Witch: Ouh you little poor, are you sure that Player helped you because it wanted to or because it couldn't advance in its mission any more if it didn't help you?

-Player: Do not listen to her, she's trying to manipulate us.

-Lion: Are you sure? But she has a point, you don't like me, you quite force me to come.

-Scarecrow: Nobody forced you to come here, we want you to be brave and ADMIN is going to help us.

-The Most Evil Bad Witch: ADMIN helps no one. I asked her a lot of times for help, to free me but I still continue here and I can't achieve my dream by being here, conquer all worlds!

-Player: That's because you are evil, and your dream will destroy thousands of other dreams.

-Tinman: Guys, do you remember what the trees told us? We must do that. if we want to achieve our dreams, we need to do it.

-Player: OK. so let's do this.

-The Most Evil Bad Witch: NOOOOOOOOOOOOOOOOOOOOO! you horrible piece of shit!

[The `kill` command is used]

-Lion: Yes! no more The Most Evil Bad Witch, no more oppression, now we are free! and we can achieve what we want.

-Scarecrow: Wiiiiiii! Let's go to the Castle! We can meet the ADMIN there.

2.1.9 CHAPTER 7: Castle

Text:

-ADMIN: Hello my beloved followers. I was waiting for you.

-Player: For real? But the guardian told us that you were busy and we have to wait a long time to see you.

-ADMIN: I knew for years that you were coming, I had an appointment already prepared for you, even if you didn't know about it. I'm supreme. I know everything and I can get things for everyone. I'm anywhere and nowhere at the same time. But The Most Evil Bad Witch asked for something a long time ago that made it impossible to destroy her, so it was impossible for me to KILL her. Thank you for doing it. This world is going to really appreciate it.

-ADMIN: Now let's go to the important thing. Scarecrow, you want a brain. Am I right?

-Scarecrow: Yes.

-ADMIN: Of course I am. FLIT FLAAT FLOO! There has to be some difference now. You already are owning a brain in your head.

-Scarecrow: Do I have to notice some change right now?

-ADMIN: Maybe not right now but yes in some time. OK, next!

-ADMIN: Tinman it's your turn, you wanted a heart?

-Tinman: Yes, I want to feel, not just to be a robot. I want to love but what I really want is to have friends.

-ADMIN: Perfect, FLARAT FLUU! Now you have a heart if I hit you like this do you feel it?

-Tinman: Ausch! Yes, I also can feel some things, I will learn to distinguish them with the time.

-ADMIN: Perfect whose time it's now?

-Lion: That 's me!

-ADMIN: I see, You want to be braver, but I see that someone really encouraged you in this new journey.

-Lion: Friends always help but I'm still really scared by a lot of things. Please I want to be a real lion and as brave as we must be. I want to reconvert into the king of the jungle!

-ADMIN: That's already done. You just need friends. But I'll still do it. FLUSH FLUSH! Perfect it's not going to be momentaneous, but this can help you to be as brave as you want to be.

-ADMIN: And now last but not least, Player. You aren't from this world so it's not really easy for me to know your biggest desire. Freedom from your parents, a new phone, a new computer, a program becoming rich?

-Player: The only thing I want is to leave this place with my Dog, to be able to come back to my home page on my computer.

-ADMIN: Oh, I see! That's not hard, the only thing you need to do is to use the EXIT command. Just use it and you will close the terminal world but don't forget to say goodbye to everyone here!

-Player: GOODBYE! And have a better live in this dark world and with no terrors anymore!

[Use `exit` command and finish game]

2.1.10 Witch phrases

2.1.10.1 1st warning:

-The Most Evil Bad Witch: As I suspect you are not good enough for this world. This is the first warning, be careful because Your little lovely Dog would die if you make another mistake.

2.1.10.2 2nd warning:

-The Most Evil Bad Witch: It's time, you can start saying goodbye to your Dog, because it doesn't exist anymore in this world, we advertise to you the Terminal is full of terrors. In the 3rd warning you will end up without legs making it impossible for you to continue with the game. MUAJAJAJA, so do you know what to do? Please make another mistake, I will be able to take control of all the worlds.

2.1.10.3 3rd warning:

-The Most Evil Bad Witch: Are you prepared? No more legs for you, no more adventure for you. It 's the end.
-Good witch: I'm sorry this happened, Use command and start again, help us defeat this witch.
-The Most Evil Bad Witch: Come if you dare, I'm just going to tell you that's impossible.

2.2 APPENDIX B

2.2.1 Specification documents

2.2.1.1 `cat` command

```
NAME
    cat - read a file and write it in the standard output.

SYNOPSIS
    cat [file_name]

DESCRIPTION
    Read an existing file by typing the file name as argument.

    We have learned the file descriptors and files pointers and how they are
        managed. We can put the file descriptors into practice using the write
        system call and the fclose function. Moreover, we have practised with
        pointers using the fopen function.

SEE ALSO
    fopen() - A function that opens a file whose name is the string pointed to by
        path and associates a stream with it.
    write() - A system call that writes something in the specified file descriptor.
    fclose() - A function that closes a file using its file descriptor.

REPORTING BUGS
    No bugs reported so far.

AUTHOR
      Written by Mikel Aristu.
```

2.2.1.2 `cd` command

```
NAME
    cd - to move around nearby directories

SYNOPSIS
    cd [DIRECTORY NAME]

DESCRIPTION
    This command allows the user to move around directories by typing cd
       DIRECTORY, where DIRECTORY is a place adjacent to the users current
       directory.
    The command  cd  DIRECTORY  in a terminal changes the users working directory
       to DIRECTORY. It can also be used to go to the previous directory by typing
       .. as the second argument, cd .. and to print the current directory by
       typing ., cd .  .

    This command must be implemented in the shell and not executed as an external
       command because else, the chdir system call affects only to the child
       process and not to the parent shell, so since we want to execute in our
       shell, we will need to implement it there.

SEE ALSO
    pwd() - The command implemented by the team. It will be used in order to write
       the current directory, as we have said above, by typing cd . .
    chdir(path) - A system call used to change the current working directory of
       the calling process to the directory specified in path.
    opendir()

REPORTING BUGS
    No bugs reported so far.

AUTHOR
    Written by Mikel Aristu.
```

2.2.1.3 cp command

```
NAME
    cp - copy a file

SYNOPSIS
    cp [file_name] [new_filename]

```

```
DESCRIPTION
    This command copies an existing file to another one. The content of the file
        is saved in a buffer and then the content of the buffer is written on the
        new file. If the file we want to copy on exists, then it will be replaced
        by the file from which we are copying.

SEE ALSO
    open() - A system call that opens and possibly creates a file or device.
    write() - A system call that writes something in the specified file descriptor.

REPORTING BUGS
    No bugs reported so far.

AUTHOR
    Written by Mikel Aristu.
```

2.2.1.4 getpass command

```
NAME
    getpass - to decrypt the password

SYNOPSIS
    getpass

DESCRIPTION
    It decrypts our password. The decryption is only simulated by the process of
        reading the files in a specific directory in a different way. We can
        achieve this executing tr command first and then the ls command. So the
        command will change a letter by another one and then ls will write the
        files in the current directory. This is made by creating a pipe that will
        execute tr command and then using what we get here as input of ls command
        to view the corresponding password.

    We have learned the management of processes using fork() command and also how
        to use pipes, in this case, the output of tr command is used as input of ls
        command. It can also be considered as a temporary connection between
        commands. This is because the connection would not be forever.

SEE ALSO
    execlp() - This C library function replaces the current process image with a
        new process image.
    pipe - creates a pipe, an unidirectional data channel that can be used for
        interprocess communication. The array pipefd is used to return two file
        descriptors referring to the ends of the pipe. pipefd[0] refers to the read
        end of the pipe. pipefd[1] refers to the write end of the pipe. Data
        written to the write end of the pipe is buffered by the kernel until it is
        read from the read end of the pipe.
    dup2 - dup2() makes newfd be the copy of oldfd, closing newfd first if
        necessary.
    fflush - For output streams, fflush() forces a write of all user-space
        buffered data for the given output or update stream via the streams
        underlying write function. For input streams, fflush() discards any
        buffered data that has been fetched from the underlying file, but has not
        been consumed by the application.

REPORTING BUGS
    No errors detected so far.

AUTHOR
    Written by Mikel Aristu.
```

2.2.1.5 grep command

```
NAME
    grep - show the occurrences of a string in lines and the number of them.

SYNOPSIS
    grep [STRING]... [FILE]...

DESCRIPTION
    Displays the entire lines where the given string occurs as well as the number
       of occurrences.

    This command works using open(), read(), write() and close() system calls.

    The grep command goes through each line of the file, analysing all the
       containing characters, in order to find if there is a match between an
       array of characters found and the one given as argument. As the entire file
       has to be read, a do while loop is executed for the task, inside containing
       another one for storing the entire line in an array. Then a nested loop is
       used to find any matching between the two char arrays for then, printing
       the entire file that has the matching string as well as the number of
       occurrences.
```

```

SEE ALSO
    open(), read(), write(), close()

REPORTING BUGS
    No errors detected so far.

AUTHOR
    Written by Zdravko Todorov.
```

2.2.1.6 `help` command

```
NAME
    help    display brief description about all available commands.

SYNOPSIS
    help

DESCRIPTION
    Displays all of the available commands in the shell and gives a brief
        description about each of them.

    This command works using write() system call.

    The help command basically, after checking that the argument number is
        correct, uses the write() system call to display all the commands and
        information about them.

SEE ALSO
    The command man has a similar, but more detailed output for each of the
        commands.

REPORTING BUGS
    No errors detected so far.

AUTHOR
    Written by Zdravko Todorov.
```

2.2.1.7 `ls` command

```
NAME
    ls    show the elements in the current directory.

SYNOPSIS
    ls [OPTION]...

DESCRIPTION
    List  information about the FILES and DIRECTORIES in the current directory.

    Flags of execution:
        -a
            do not ignore entries starting with .
        -l
            list all the entries of the directory, and their properties:
                name
                size
                UID
                permissions

    This command works using opendir(), readdir(), write() and stat() system calls.

    Since directories are no more than files that contain pairs of i-nodes and
        names in order to identify all the files and directories inside them, ls
        makes use of opendir() to open a certain directory, in a DIR struct. Then,
        readdir() reads the contents in the directory, line by line, like a char
        array and, making use of write() system call, the programs writes  those
        elements in stdout. The stat() system call is used to obtain certain
        information about the file, in the option -l. This extra info is written in
        the stdout using the printf() library function.

SEE ALSO
    In order to simplify the functioning of this command, printf() is used for
        formatted printing of the elements of stat file.

REPORTING BUGS
    No errors detected so far.

AUTHOR
    Written by Zdravko Todorov.
```

2.2.1.8 man command

```

NAME
    man - show the manual for the given command

SYNOPSIS
    man [command_name]

DESCRIPTION
    Prints the specification document for the requested command.

    The given command will have a .txt file with its related specification
        document. So, basically, the man command takes the file related to the
        given command and prints it directly in the terminal. Note: The
        specification documents have been previously converted into .txt files in
        order to gain the ability to read them from the code.

SEE ALSO
    help - to see all the available commands in the game.

REPORTING BUGS
    No bugs detected so far.

AUTHOR
    Written by Zdravko Todorov.
```

2.2.1.9 `mv` command

```
NAME
    mv - move a selected file to a selected location.

SYNOPSIS
    mv [file_name] [new_location]

DESCRIPTION
    Move an existing file to a new location by typing the file and the location as
        arguments.

    We have learned the pointers to different files and how they are located in
        different working directories. Moreover, we have seen the functions link
        and unlink. The link() function is used to assign a new directory to a
        file, in other words, link() creates a new hard link to an existing file.
        On the other hand, with the function unlink() we can delete a file. So
```

```
           actually what we are doing is a copy of a file in another directory and
           then delete the file from which we have done such copy.

SEE ALSO
       link() - This function creates a new link (also known as a hard link) to an
           existing file.
       unlink() - This function deletes a name from the file system. If that name was
           the last link to a file and no processes have the file opened, the file is
           deleted and the space it was using is made available for reuse.

REPORTING BUGS
       No bugs reported so far.

AUTHOR
       Written by Mikel Aristu.
```

2.2.1.10 pwd command

```
NAME
       pwd - Look in which location is the player at the moment.

SYNOPSIS
       pwd

DESCRIPTION
       This command will tell you in which place is the player at the moment.

       Add here how this code is related to what we have learned in class.

       The getcwd library function will return the current working directory in case
           of no errors detected. Otherwise it will return a NULL value so the player
           will be advised.

SEE ALSO
       getcwd() - This function copies an absolute pathname of the current working
           directory
       free() - This function frees the memory space pointed to.

REPORTING BUGS
       No bugs reported so far.

AUTHOR
```

```
    Written by Mikel Aristu.
```

2.2.1.11 stee command

```
NAME
    stee - It asks your name in order to save it.

SYNOPSIS
    stee

DESCRIPTION
    This command will ask your name and then it will save it.

    We have learned the file pointers and file descriptors and how they are managed. We can put the file pointers into practice using the fopen() function. Moreover, we have practised with file descriptors using the write system call.

SEE ALSO
    fopen() - A function that opens a file whose name is the string pointed to by path and associates a stream with it.
    write - A system call that writes something in the specified file descriptor.
    printf() - A function that closes a file using its file descriptor.
    scanf() - The scanf() function reads input from the standard input stream stdin.
REPORTING BUGS
    No bugs reported so far.

AUTHOR
    Written by Mikel Aristu.
```

2.2.1.12 touch command

```
NAME
    touch - create a new file

SYNOPSIS
    touch [new_file_name]

DESCRIPTION
```

```
    Create a new file by writing the new file name as argument.

    We have learned the way the files are created. In this case, we are using the
        system call open() which returns a file descriptor that we will use to read
        or write the file, of course, if we have such permissions.

SEE ALSO
    open() - A system call that creates a new open file description, an entry in
        the system table of open files. This entry records the file offset and the
        file status flags. A file descriptor is a reference to one of these
        entries; this reference is unaffected if pathname is subsequently removed
        or modified to refer to a different file. The new open file description is
        initially not shared with any other process, but sharing may arise via
        fork(2).

REPORTING BUGS
    No bugs detected so far.

AUTHOR
    Written by Mikel Aristu.
```

2.2.2 Verification documents

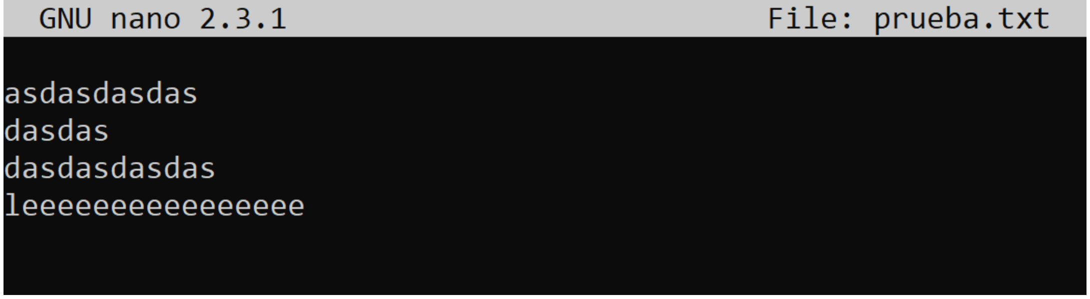
2.2.2.1 `cat` command

Program or utility to verify: cat command
Author(s) of the design: Mikel Aristu, David Cuenca
Author(s) of the code: Mikel Aristu, David Cuenca
Author(s) of the verification document: Mikel Aristu, Zdravko Todorov
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

1st verification case

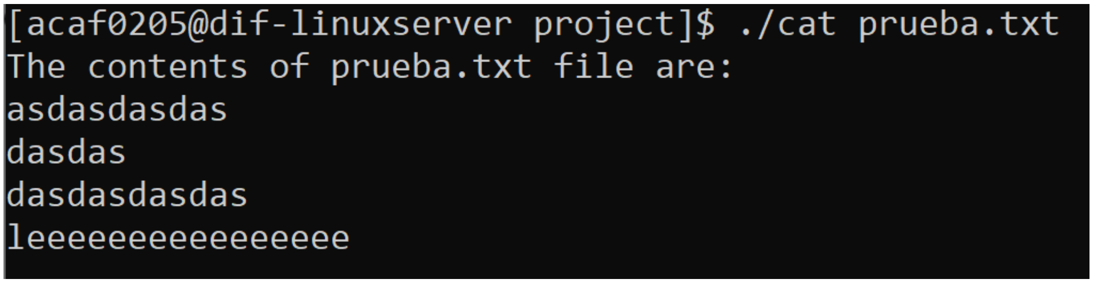
Objective: To check if the `cat` command works well. *Note: An alternative file is used for testing purposes only, as the game files are really long.*
Input: `cat prueba.txt`
Previewed output: Write in the terminal the contents of prueba.txt

The data of prueba.txt:



```
  GNU nano 2.3.1                                File: prueba.txt

asdasdasdas
dasdas
dasdasdasdas
leeeeeeeeeeeeeeee
```

Verification output:



```
[acaf0205@dif-linuxserver project]$ ./cat prueba.txt
The contents of prueba.txt file are:
asdasdasdas
dasdas
dasdasdasdas
leeeeeeeeeeeeeeee
```

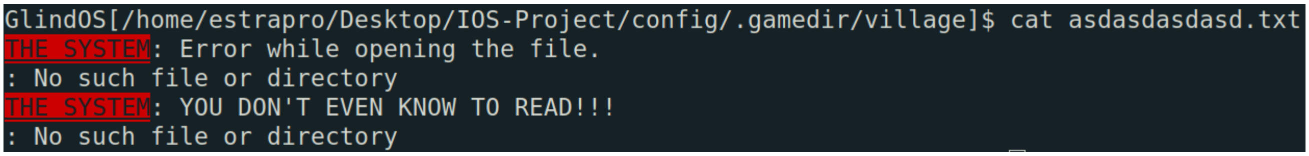
Explanation: The command is executed correctly and it has no execution errors, so that the contents of the file are shown.

2nd verification case

Objective: To check that the cat command will write an error message on the terminal if input file does not exist. *Note: An alternative file is used for testing purposes only, as the game files are really long.*
Input: `cat asdasdasdasd.txt` (asdasdasdasd.txt does not exist)
Previewed output:

```
THE SYSTEM: Error while opening the file.
:  No such file or directory
THE SYSTEM: YOU DON'T EVEN KNOW TO READ!!!
:  No such file or directory
```

Verification output:



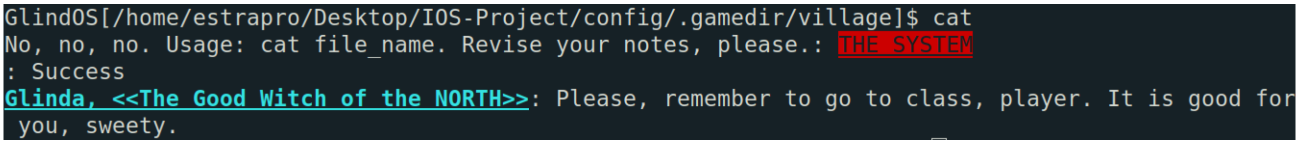
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cat asdasdasdasd.txt
THE SYSTEM: Error while opening the file.
: No such file or directory
THE SYSTEM: YOU DON'T EVEN KNOW TO READ!!!
: No such file or directory
```

Explanation: The asdasdasd.txt file doesn't exist, so it can't be opened in order to read the contents.

<u>**3rd verification case**</u>

Objective: To check that if the command is written in an incorrect way, it explains how to use it.
Input: `cat`
Previewed output: `No, no, no. Usage: cat file_name. Revise your notes, please.:`
`THE SYSTEM : Success`
`Glinda, <<The Good Witch of the NORTH>>: Please, remember to go to class, player. It is good for you, sweety.`

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cat
No, no, no. Usage: cat file_name. Revise your notes, please.: THE SYSTEM
: Success
Glinda, <<The Good Witch of the NORTH>>: Please, remember to go to class, player. It is good for
 you, sweety.
```

Explanation: When trying to execute the cat command with no input file, it will tell the player how to use the command.

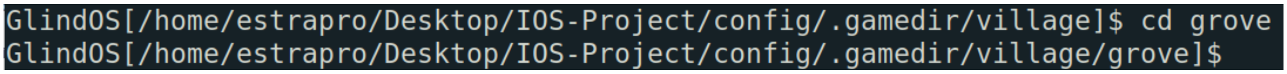
2.2.2.2 `cd` command

Program or utility to verify: cd command
Author(s) of the design: Mikel Aristu, David Cuenca
Author(s) of the code: Mikel Aristu, David Cuenca
Author(s) of the verification document: Mikel Aristu, Zdravko Todorov
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

<u>**1st verification case**</u>

Objective: To check if the `cd` command works well. We will change from *village* directory to *grove*.
Input: `cd grove`
Previewed output: The current directory will be changed to grove.

Verification output:



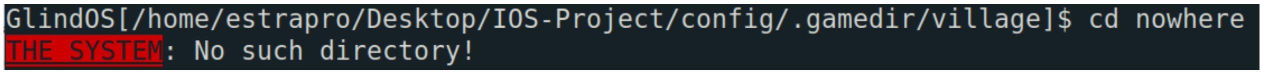
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cd grove
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove]$
```

Explanation: The command is executed correctly and it has no execution errors, so that the current working directory will be changed. We can see the change in the whole directory.

2nd verification case

Objective: To check if we put an invented directory, it advises about that.
Input: `cd nowhere`
Previewed output: `THE SYSTEM: No such directory!`

Verification output:



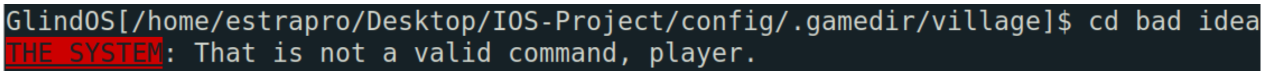
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cd nowhere
THE SYSTEM: No such directory!
```

Explanation: The command is executed correctly but the directory is not found in the current working directory. The error message will appear informing the player about the issue.

3rd verification case

Objective: To check what happens if the command is not used properly. In this case more than one argument is inserted
Input: `cd bad idea`
Previewed output: `THE SYSTEM: That is not a valid command, player.`

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cd bad idea
THE SYSTEM: That is not a valid command, player.
```

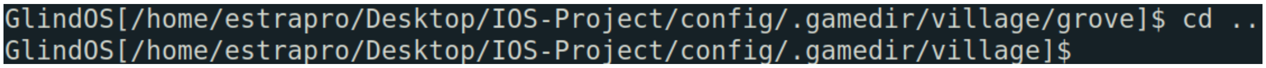
Explanation: When trying to execute the cd command incorrectly, that's it, inserting more then the correct number of arguments, an error is thrown, telling the player that the command is not valid, as it is not used correctly.

4th verification case

Objective: To check how `cd ..` can be used to return to the parent directory, that's it, to go to the previous directory in the hierarchical level.
Input: `cd ..`

Previewed output: It will go from *grove* to *village*.

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove]$ cd ..
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$
```

Explanation: The command work perfectly as expected, it returns from grove directory to his parent directory, village.

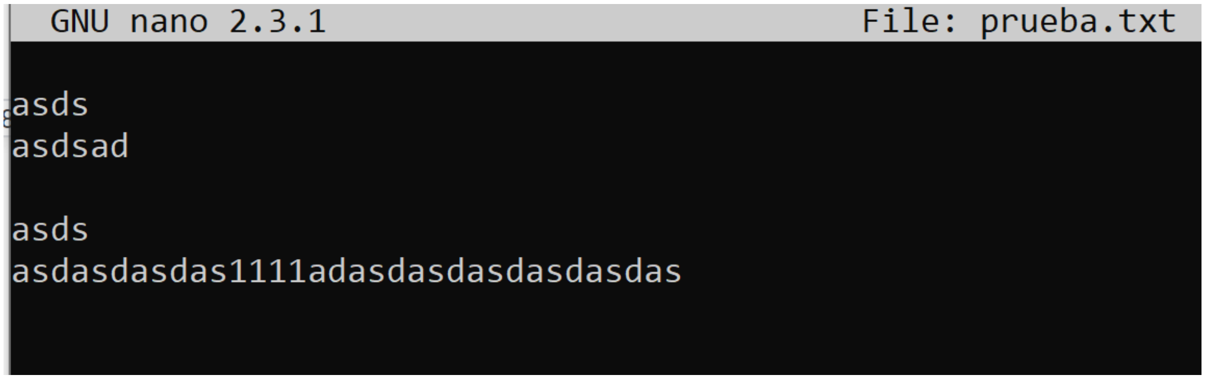
2.2.2.3 `cp` command

Program or utility to verify: `cp` command
Author(s) of the design: Mikel Aristu
Author(s) of the code: Mikel Aristu
Author(s) of the verification document: Mikel Aristu , Zdravko Todorov
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

<u>**1st verification case**</u>

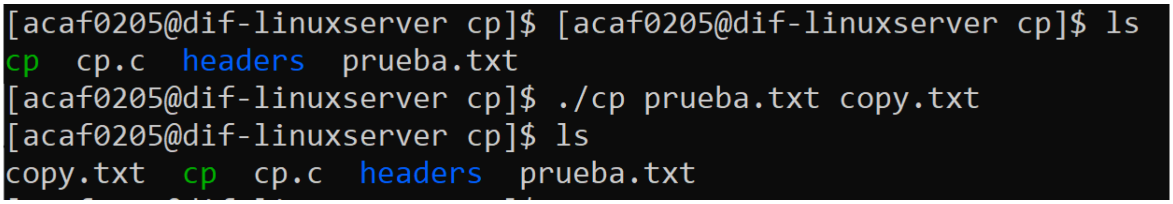
Objective: To check if the `cp` command works well. *Note: An alternative file is used for testing purposes only, as the game files are really long.*
Input: `cp prueba.txt copy.txt`
Previewed output: Copy the file prueba.txt in copy.txt

The data of prueba.txt:



```
  GNU nano 2.3.1                         File: prueba.txt

asds
asdsad

asds
asdasdasdas1111adasdasdasdasdasdas
```

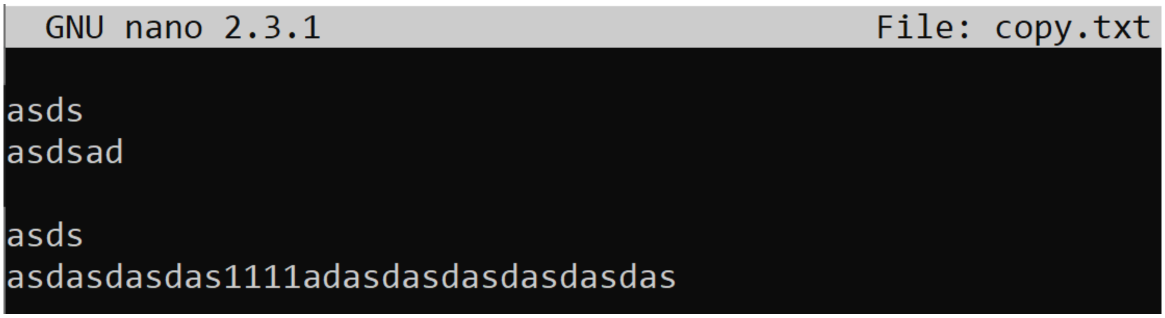
Verification output:



```
[acaf0205@dif-linuxserver cp]$ [acaf0205@dif-linuxserver cp]$ ls
cp  cp.c  headers  prueba.txt
[acaf0205@dif-linuxserver cp]$ ./cp prueba.txt copy.txt
[acaf0205@dif-linuxserver cp]$ ls
copy.txt  cp  cp.c  headers  prueba.txt
```

Explanation: The command is executed correctly and it has no execution errors, so that the contents of the file prueba.txt are copied in copy.txt.

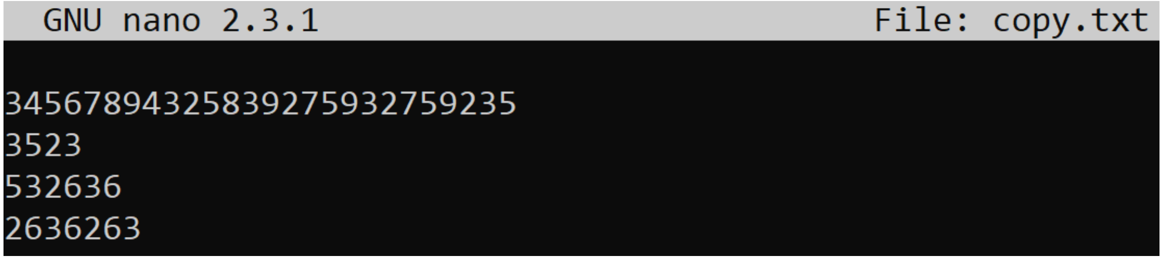
The content of copy.txt:



```
  GNU nano 2.3.1                        File: copy.txt

asds
asdsad

asds
asdasdasdas1111adasdasdasdasdasdas
```

2nd verification case

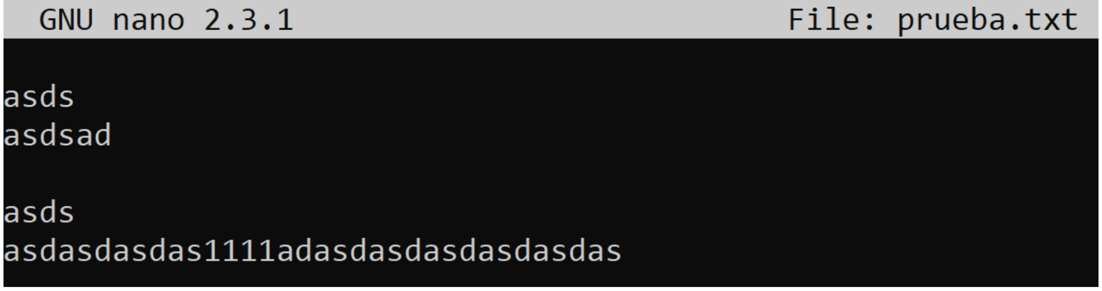
Objective: To check that the copy command works well if already exists the file in which we want to copy. This file is the same that in the 1st verification case but we have changed its content. The new content is the following: *Note: An alternative file is used for testing purposes only, as the game files are really long.*



```
  GNU nano 2.3.1                        File: copy.txt

34567894325839275932759235
3523
532636
2636263
```

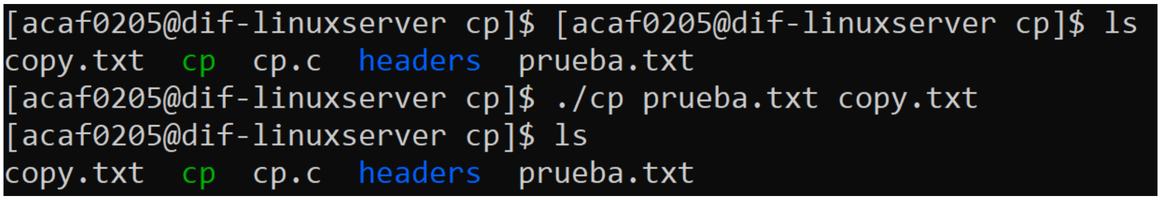
Input: `./cp prueba.txt copy.txt` (the file copy.txt already exists)
Previewed output: The data from prueba.txt will be copied in copy.txt, replacing all its content.
Verification output:

The content of prueba.txt:



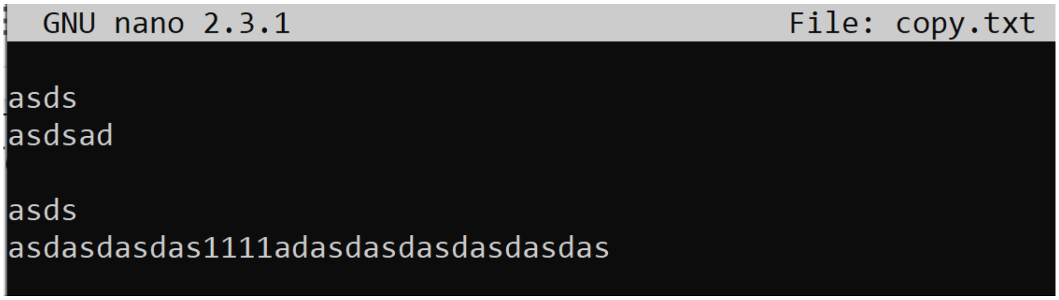
```
 GNU nano 2.3.1                    File: prueba.txt

asds
asdsad

asds
asdasdasdas1111adasdasdasdasdasdas
```

We execute the command:



```
[acaf0205@dif-linuxserver cp]$ [acaf0205@dif-linuxserver cp]$ ls
copy.txt  cp  cp.c  headers  prueba.txt
[acaf0205@dif-linuxserver cp]$ ./cp prueba.txt copy.txt
[acaf0205@dif-linuxserver cp]$ ls
copy.txt  cp  cp.c  headers  prueba.txt
```

And the content of prueba.txt is copied and replaced in copy.txt:

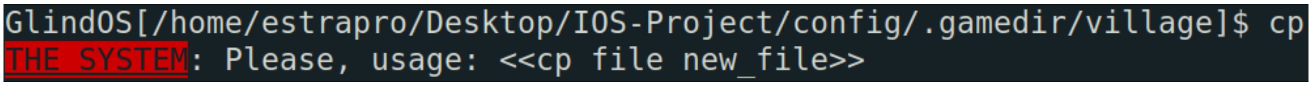


```
 GNU nano 2.3.1                    File: copy.txt

asds
asdsad

asds
asdasdasdas1111adasdasdasdasdasdas
```

Explanation: The copy.txt file already exists, so its content is replaced by the content of prueba.txt.

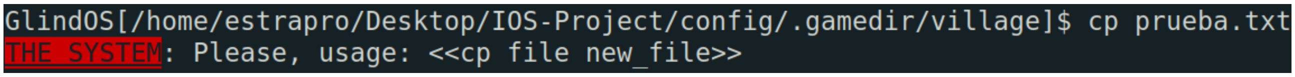
<u>3rd verification case</u>

Objective: To check that if the command is written in an incorrect way, it explains how to use it.
Input: `cp`
Previewed output: `THE SYSTEM: Please, usage: <<cp file new_file>>`
Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cp
THE SYSTEM: Please, usage: <<cp file new_file>>
```

Explanation: When trying to execute the `cp` command with no input files, it will tell the player how to use the command. The same happens if we try to execute the command with a file that we want to copy from but with no file in which we want to copy the content, as we can see:



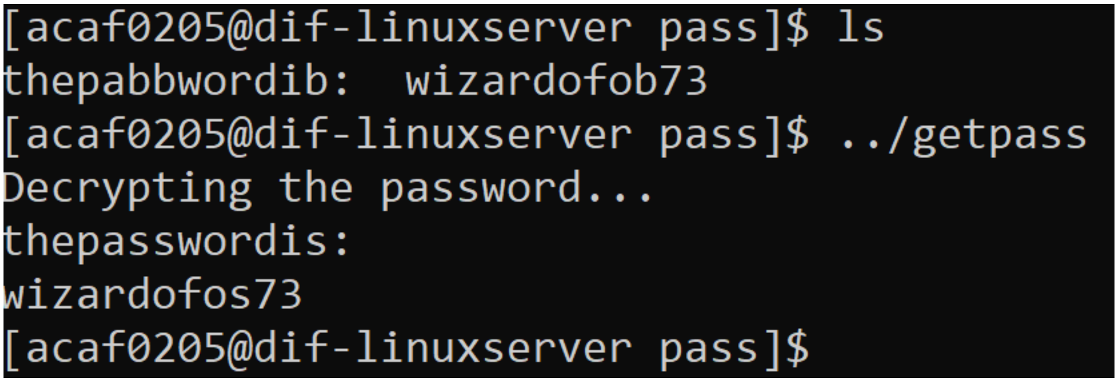
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ cp prueba.txt
THE SYSTEM: Please, usage: <<cp file new_file>>
```

2.2.2.4 `getpass` command

Program or utility to verify: `getpass` command
Author(s) of the design: Mikel Aristu
Author(s) of the code: Mikel Aristu, Leire Estanga
Author(s) of the verification document: Leire Estanga
Author(s) of the verification: Mikel Aristu

1st verification case
Objective: To check if the `getpass` command works well.
Input: `getpass`
Previewed output: The correct password to access the emerald city.

Verification output:



```
[acaf0205@dif-linuxserver pass]$ ls
thepabbwordib:  wizardofob73
[acaf0205@dif-linuxserver pass]$ ../getpass
Decrypting the password...
thepasswordis:
wizardofos73
[acaf0205@dif-linuxserver pass]$
```

Explanation: The command is executed correctly and it has no execution errors, so that the password is decrypted correctly. At first, we have two different files with random letters and then, executing the command we are able to see that the filenames have changed telling us that the password is `wizardofos73`.

2.2.2.5 `grep` command

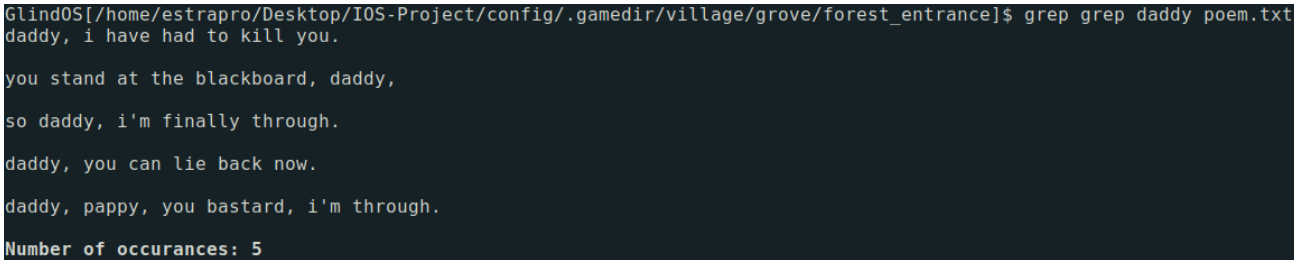
Program or utility to verify: grep command
Author(s) of the design: Zdravko Tororov, Leire Estanga
Author(s) of the code: Zdravko Tororov
Author(s) of the verification document: Leire Estanga
Author(s) of the verification: Zdravko Tororov

<u>**1st verification case**</u>

Objective: To check if the `grep` command works well.
Input: `grep daddy poem.txt`
Previewed output:

```
daddy, i have had to kill you.
you stand at the blackboard, daddy,
so daddy, i’m finally through.
daddy, you can lie back now.
daddy, pappy, you bastard, i’m through.

Number of occurances:  5
```

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove/forest_entrance]$ grep grep daddy poem.txt
daddy, i have had to kill you.

you stand at the blackboard, daddy,

so daddy, i'm finally through.

daddy, you can lie back now.

daddy, pappy, you bastard, i'm through.

Number of occurances: 5
```

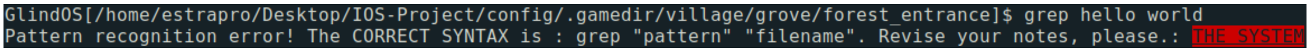
Explanation: The command is executed correctly and has no execution errors, so the lines are printed correctly, and exactly the ones that have the *daddy* word.

<u>**2nd verification case**</u>

Objective: To check what happens if the command is not used properly.
Input: `grep`
Previewed output:

```
Pattern recognition error!  The CORRECT SYNTAX is :  grep "pattern" "filename".
Revise your notes, please.
```

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove/forest_entrance]$ grep hello world
Pattern recognition error! The CORRECT SYNTAX is : grep "pattern" "filename". Revise your notes, please.: THE SYSTEM
```

Explanation: When trying to execute the `grep` command incorrectly, then the shell will teach how to use the command.

2.2.2.6 `help` command

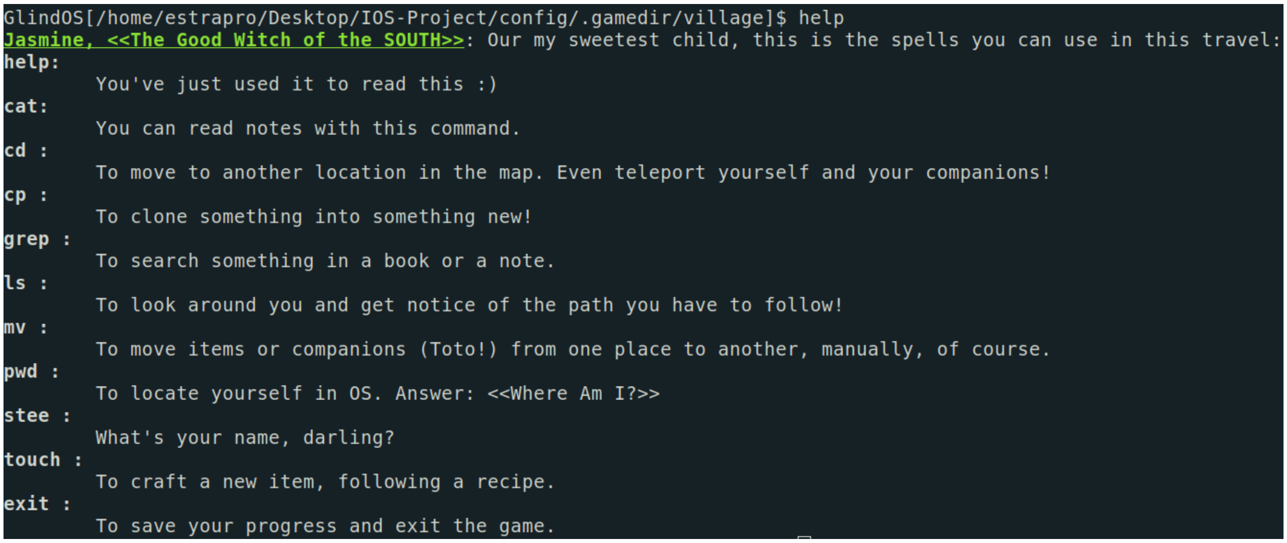
Program or utility to verify: `help` command
Author(s) of the design: Zdravko Tororov, Leire Estanga
Author(s) of the code: Zdravko Tororov, David Cuenca
Author(s) of the verification document: Zdravko Tororov
Author(s) of the verification: Zdravko Tororov

1st verification case

Objective: To check if the `help` command works well. This command can be written at any moment
Input: `help`
Previewed output:

```
Jasmine, <<The Good Witch of the SOUTH>>: Our my sweetest child, this is the spells you can use in this travel:

help :
    You have just used it to read this :)
cat :
    You can read notes with this command.
cd :
    To move to another location in the map. Even teleport yourself and your companions!
cp :
    To clone something into something new!
grep :
    To search something in a book or a note.
ls :
    To look around you and get notice of the path you have to follow!
mv :
    To move items or companions (Toto!) from one place to another, manually, of course.
pwd :
    To locate yourself in OS. Answer: <<Where Am I?>>
stee :
    What is your name, darling?
touch :
    To craft a new item, following a recipe.
exit :
    To save your progress and exit the game.
```

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ help
Jasmine, <<The Good Witch of the SOUTH>>: Our my sweetest child, this is the spells you can use in this travel:
help:
        You've just used it to read this :)
cat:
        You can read notes with this command.
cd :
        To move to another location in the map. Even teleport yourself and your companions!
cp :
        To clone something into something new!
grep :
        To search something in a book or a note.
ls :
        To look around you and get notice of the path you have to follow!
mv :
        To move items or companions (Toto!) from one place to another, manually, of course.
pwd :
        To locate yourself in OS. Answer: <<Where Am I?>>
stee :
        What's your name, darling?
touch :
        To craft a new item, following a recipe.
exit :
        To save your progress and exit the game.
```

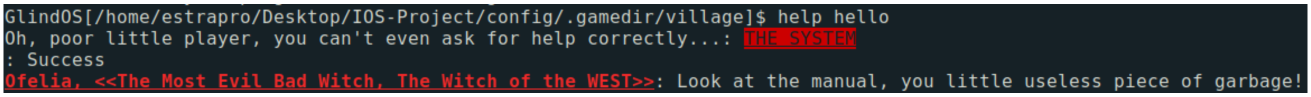
Explanation: As the syntax is correct, the command works as expected.

2nd verification case

Objective: To display an error message due to the incorrect syntax.
Input: `help hello`
Previewed output:

```
Oh, poor little player, you can't even ask for help correctly...:  THE SYSTEM
Ofelia, <<The Most Evil Bad Witch, The Witch of the WEST>>: Look at the manual,
you little useless piece of garbage!
```

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ help hello
Oh, poor little player, you can't even ask for help correctly...: THE SYSTEM
: Success
Ofelia, <<The Most Evil Bad Witch, The Witch of the WEST>>: Look at the manual, you little useless piece of garbage!
```

Explanation: The syntax is incorrect, because the number of argument is not 1, so an error message is displayed in the standard error output.

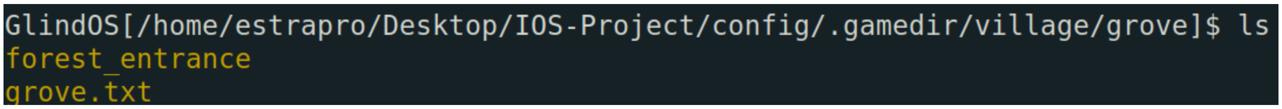
2.2.2.7 `ls` command

Program or utility to verify: `ls` command
Author(s) of the design: Zdravko Todorov, David Cuenca
Author(s) of the code: Zdravko Todorov, David Cuenca
Author(s) of the verification document: Leire Estanga
Author(s) of the verification: Zdravko Todorov

1st verification case

Objective: To check if the `ls` command works well. We will introduce simply `ls` to show the yellow path to follow. This will be at any directory.
Input: `ls`
Previewed output: It will list the content of the directory. In this case, the file containing the text for that chapter and the next directory in the yellow path to be followed.

Verification output:



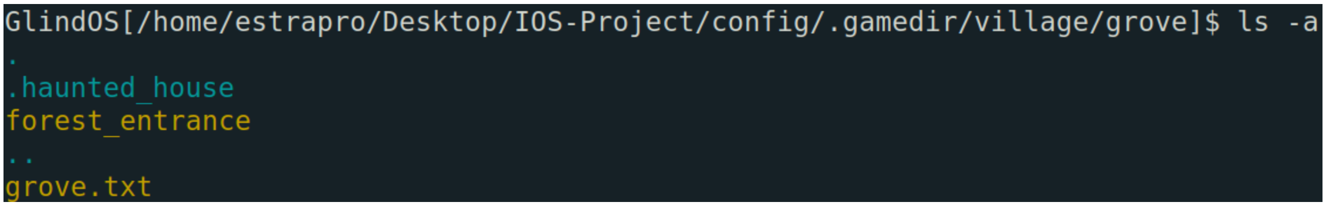
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove]$ ls
forest_entrance
grove.txt
```

Explanation: Using only `ls` shows uniquely the yellow path, so no more information is given. The command works as expected.

2nd verification case

Objective: To check if the `ls -a` command works well. We will introduce `ls -a` to show both the yellow path and the blue path to follow. That's it, the main path to follow (the yellow one) will be displayed, as well as the alternative one (the blue). This will be at any directory.
Input: `ls -a`
Previewed output: It will list the content of the directory. In this case, the file containing the text for that chapter and both, the next directory in the yellow path and the blue one, to be followed. This is obtained differing the ordinary directories from the hidden ones, making them blue.

Verification output:



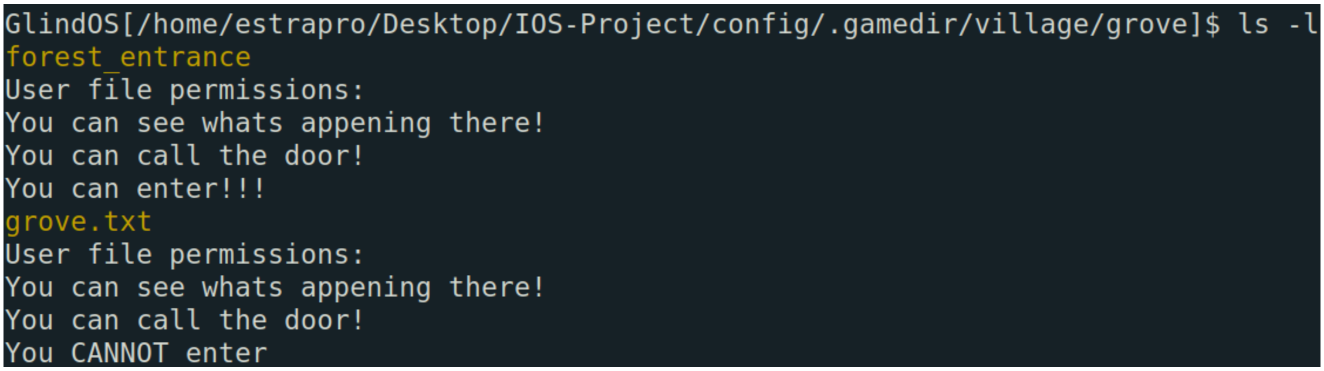
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove]$ ls -a
.
.haunted_house
forest_entrance
..
grove.txt
```

Explanation: Using `ls -a` shows perfectly all the content of the directory. The command works as expected.

3rd verification case

Objective: To check if the `ls -l` command works well. We will introduce `ls -l` to show a more detailed information about the yellow path to follow. This will be at any directory.
Input: `ls -l`
Previewed output: It will list the yellow path to be followed, along the detailed information about each next directory, more precisely, the permissions. We can see that we CAN enter to the directories, but we CANNOT enter to the files, as it is obvious.

Verification output:



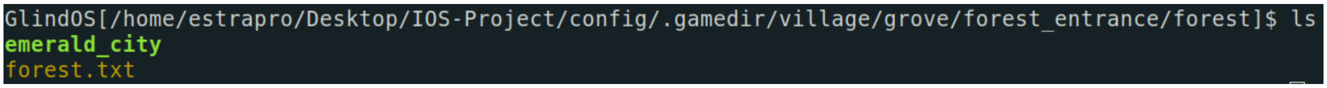
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove]$ ls -l
forest_entrance
User file permissions:
You can see whats appening there!
You can call the door!
You can enter!!!
grove.txt
User file permissions:
You can see whats appening there!
You can call the door!
You CANNOT enter
```

Explanation: Using `ls -l` shows the permissions for each directory and file. The command works as expected.

4th verification case

Objective: To check if using `ls` command in the forest directory will have the expected results. This will ONLY happen in the forest directory, as it is the entry for Emerald City.
Input: `ls`
Previewed output: The content of the directory will be listed as usual, but as we have Emerald City in front of us (our main leading objective), it will appear in green colour.

Verification output:



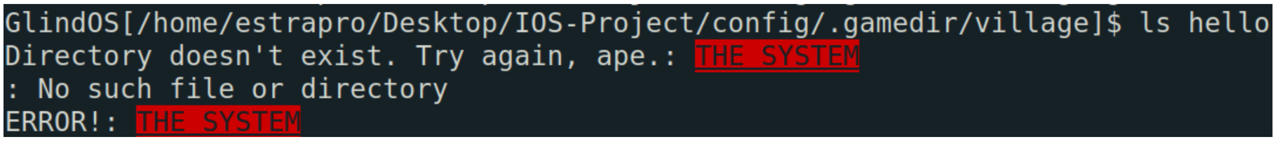
```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village/grove/forest_entrance/forest]$ ls
emerald_city
forest.txt
```

Explanation: We can see that `emerald_city` appears tinted in green colour. The command works as expected.

5th verification case

Objective: To check if the `ls` command works well. We will introduce a random word after ls, so it should give us an error.
Input: `ls hello`
Previewed output: `Directory doesn’t exist. Try again, ape.: THE SYSTEM : No such file or directory ERROR!: THE SYSTEM`

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ ls hello
Directory doesn't exist. Try again, ape.: THE SYSTEM
: No such file or directory
ERROR!: THE SYSTEM
```

Explanation: As it should, an error is thrown when we introduce the command incorrectly. The command works as expected.

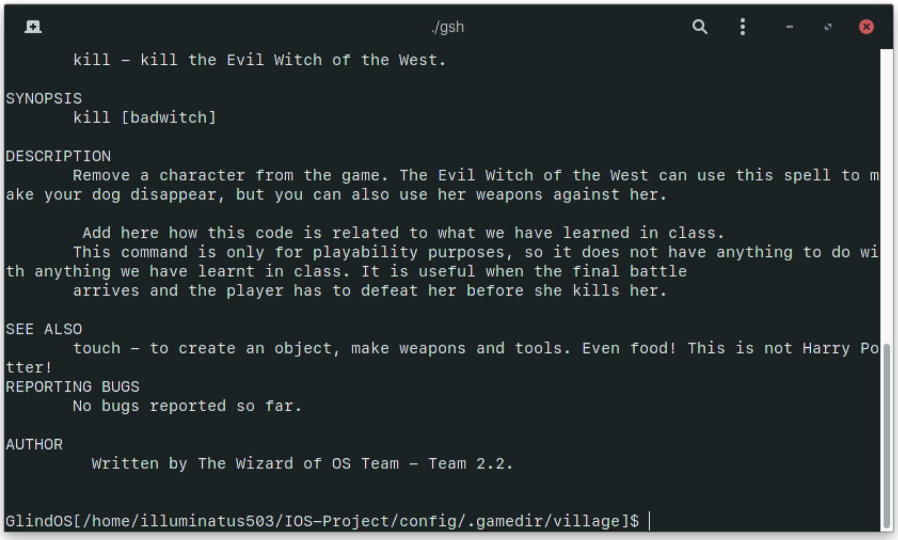
2.2.2.8 `man` command

Program or utility to verify: `man` command
Author(s) of the design: David Cuenca
Author(s) of the code: David Cuenca
Author(s) of the verification document: Leire Estanga
Author(s) of the verification: David Cuenca

1st verification case

Objective: To check if the `man` command works well.
Input: `man kill`
Previewed output: It will show the manual of the `kill` command.

Verification output:



```
      kill - kill the Evil Witch of the West.

SYNOPSIS
      kill [badwitch]

DESCRIPTION
      Remove a character from the game. The Evil Witch of the West can use this spell to m
ake your dog disappear, but you can also use her weapons against her.

       Add here how this code is related to what we have learned in class.
      This command is only for playability purposes, so it does not have anything to do wi
th anything we have learnt in class. It is useful when the final battle
      arrives and the player has to defeat her before she kills her.

SEE ALSO
      touch - to create an object, make weapons and tools. Even food! This is not Harry Po
tter!
REPORTING BUGS
      No bugs reported so far.

AUTHOR
        Written by The Wizard of OS Team - Team 2.2.


GlindOS[/home/illuminatus503/IOS-Project/config/.gamedir/village]$
```

Explanation: The command work correctly and doesn't have any execution error.

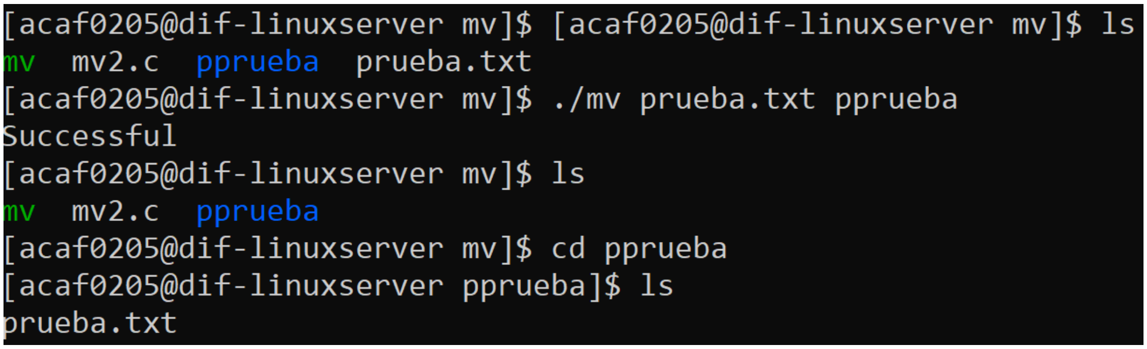
2.2.2.9 `mv` command

Program or utility to verify: `mv` command
Author(s) of the design: Mikel Aristu
Author(s) of the code: Mikel Aristu
Author(s) of the verification document: Mikel Aristu, Zdravko Todorov
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

<u>1st verification case</u>

Objective: To check if the `mv` command works well. *Note: An alternative file is used for testing purposes only, as the game files are really long.*
Input: `mv prueba.txt pprueba`
Previewed output: The file `prueba.txt` will move to the directory called `pprueba`.

Verification output:



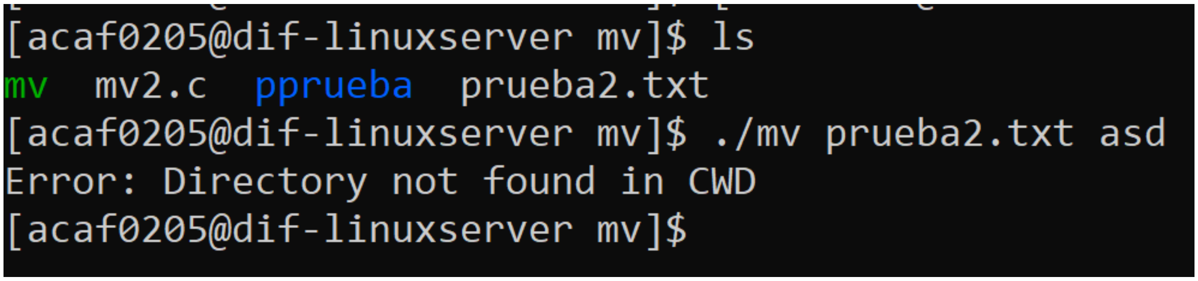
```
[acaf0205@dif-linuxserver mv]$ [acaf0205@dif-linuxserver mv]$ ls
mv  mv2.c  pprueba  prueba.txt
[acaf0205@dif-linuxserver mv]$ ./mv prueba.txt pprueba
Successful
[acaf0205@dif-linuxserver mv]$ ls
mv  mv2.c  pprueba
[acaf0205@dif-linuxserver mv]$ cd pprueba
[acaf0205@dif-linuxserver pprueba]$ ls
prueba.txt
```

Explanation: The command is executed correctly and it has no execution errors, so that the file is moved to the selected directory. In the first ls command we can see that the file prueba.txt is located in a directory called mv, then we execute the mv command and we can see that in the directory is not the file prueba.txt, this is because it has been moved to the prueba directory, as we have asked. After that, we go into the pprueba directory and the file is there, so we can conclude that the command works properly.

2nd verification case

Objective: To check if we put an invented directory, it advises about that. *Note: An alternative file is used for testing purposes only, as the game files are really long.*
Input: `mv prueba2.txt asd`
Previewed output: `Error: Directory not found in CWD.`

Verification output:



```
[acaf0205@dif-linuxserver mv]$ ls
mv   mv2.c   pprueba   prueba2.txt
[acaf0205@dif-linuxserver mv]$ ./mv prueba2.txt asd
Error: Directory not found in CWD
[acaf0205@dif-linuxserver mv]$
```

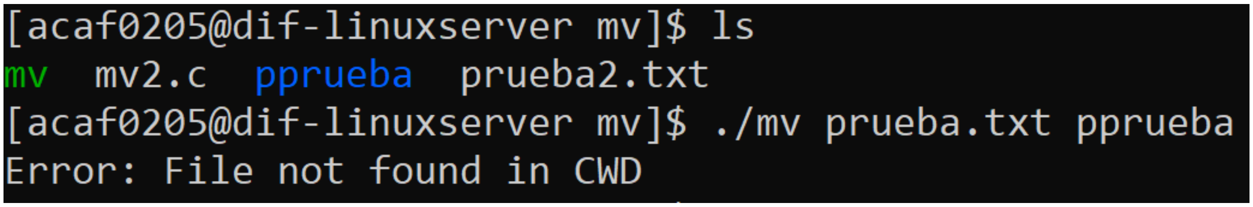
Explanation: The command is executed correctly but the directory is not found in the current working directory. The error message will appear and no file will be moved.

3rd verification case

Objective: To check what happens if the input file doesn't exist.
Input: `mv prueba.txt pprueba`

Previewed output: `Error: File not found in CWD.`

Verification output:

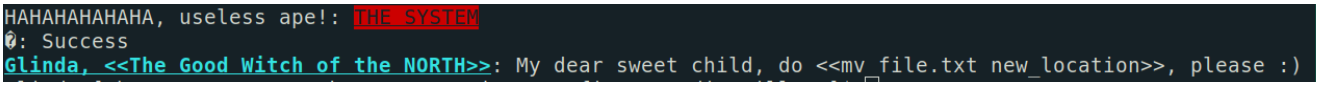


```
[acaf0205@dif-linuxserver mv]$ ls
mv  mv2.c  pprueba  prueba2.txt
[acaf0205@dif-linuxserver mv]$ ./mv prueba.txt pprueba
Error: File not found in CWD
```

Explanation: When trying to execute the mv command with a file that is not in the current working directory, an error message will appear and no file will be moved.

<u>**4th verification case**</u>

Objective: To check what happens if the command is not used properly.
Input: `mv`
Previewed output:
`HAHAHAHAHAHA, useless ape!:` <u>`THE SYSTEM`</u>
<u>`Glinda, <<The Good Witch of the NORTH>>:`</u> `My dear sweet child, do <<mv file.txt new_location>>, please :)`
Verification output:



```
HAHAHAHAHAHA, useless ape!: THE SYSTEM
�: Success
Glinda, <<The Good Witch of the NORTH>>: My dear sweet child, do <<mv_file.txt new_location>>, please :)
```

Explanation: When trying to execute the mv command incorrectly, then the shell will teach how to use the command.

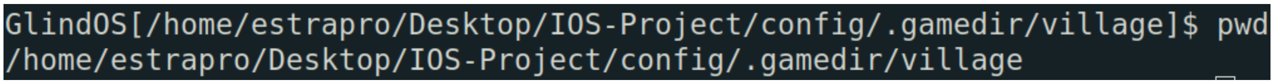
2.2.2.10 `pwd` command

Program or utility to verify: `pwd` command
Author(s) of the design: David Cuenca
Author(s) of the code: David Cuenca
Author(s) of the verification document: Mikel Aristu, Zdravko Todorov
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

<u>**1st verification case**</u>

Objective: To check if the `pwd` command works well.
Input: `pwd`
Previewed output: Write in the terminal the current working directory.

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ pwd
/home/estrapro/Desktop/IOS-Project/config/.gamedir/village
```

Explanation: The command is executed correctly and it has no execution errors, so that the shell says you in which working directory are you at the moment.

2.2.2.11 `stee` command

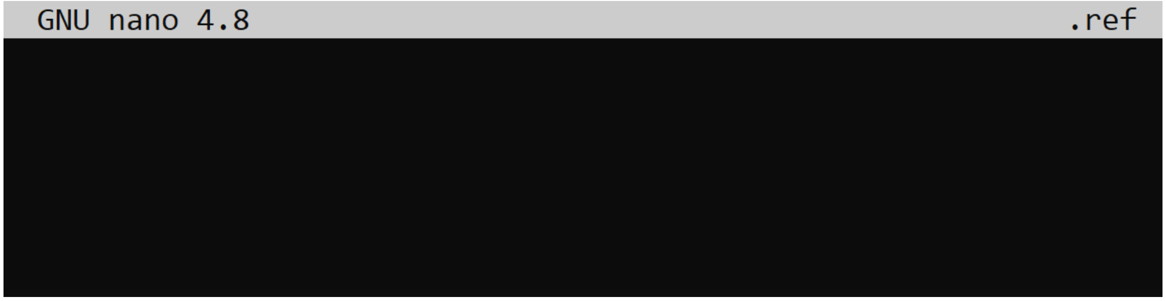
Program or utility to verify: `stee` command
Author(s) of the design: Mikel Aristu, Leire Estanga
Author(s) of the code: Mikel Aristu
Author(s) of the verification document: Mikel Aristu
Author(s) of the verification: Mikel Aristu

<u>1st verification case</u>

Objective: To check if the `stee` command works well.
Input: `stee`
Previewed output: Saves the name of the player in `config/.player/.ref` file.

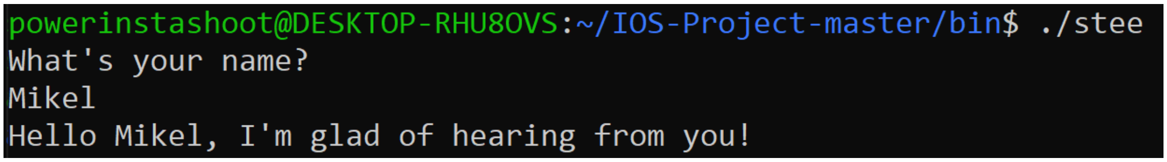
Verification output:

The content of `.ref` file:



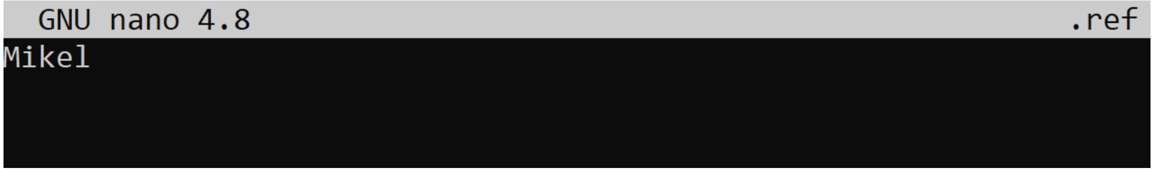
```
  GNU nano 4.8                                                    .ref
```

The `stee` command is executed:



```
powerinstashoot@DESKTOP-RHU8OVS:~/IOS-Project-master/bin$ ./stee
What's your name?
Mikel
Hello Mikel, I'm glad of hearing from you!
```

The `.ref` file's content is changed:



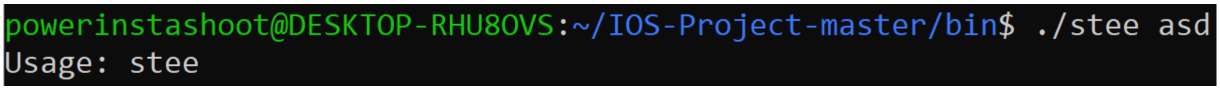
```
  GNU nano 4.8                                                  .ref
Mikel
```

Explanation: The command is executed correctly and it has no execution errors, so that it will ask the player's name, the player will write it and the name will be saved on the corresponding file.

2nd verification case

Objective: To check that if the command is written in an incorrect way, it explains how to use it.
Input: `./stee asd`
Previewed output: Error message: `Usage:  stee`

Verification output:



```
powerinstashoot@DESKTOP-RHU8OVS:~/IOS-Project-master/bin$ ./stee asd
Usage: stee
```

Explanation: When trying to execute the `stee` command with one or many parameters, it will tell the player how to use the command, only with one argument.

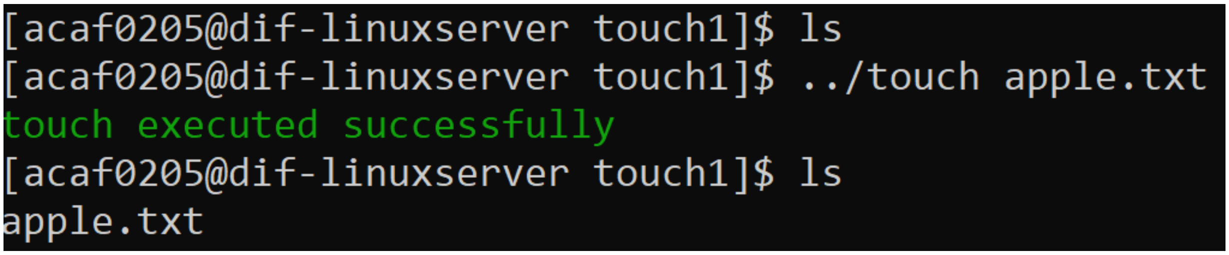
2.2.2.12 `touch` command

Program or utility to verify: `touch` command
Author(s) of the design: Mikel Aristu, David Cuenca
Author(s) of the code: Mikel Aristu
Author(s) of the verification document: Mikel Aristu
Author(s) of the verification: Mikel Aristu, Zdravko Todorov

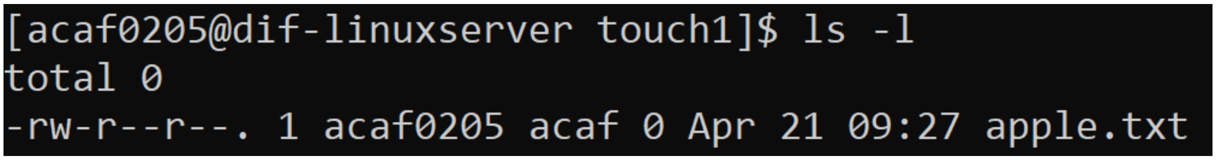
1st verification case

Objective: To check if the `touch` command works well.
Input: `touch apple`
Previewed output: The file `apple.txt` will be created with permissions of reading and writing for the owner and only of reading for groups and others.

Verification output:



```
[acaf0205@dif-linuxserver touch1]$ ls
[acaf0205@dif-linuxserver touch1]$ ../touch apple.txt
touch executed successfully
[acaf0205@dif-linuxserver touch1]$ ls
apple.txt
```



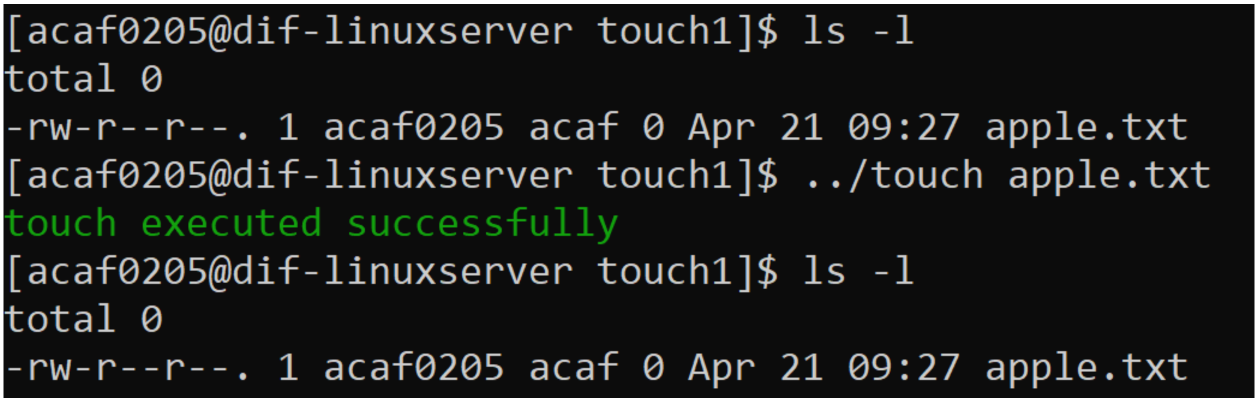
```
[acaf0205@dif-linuxserver touch1]$ ls -l
total 0
-rw-r--r--. 1 acaf0205 acaf 0 Apr 21 09:27 apple.txt
```

Explanation: The command is executed correctly and it has no execution errors, so that the file is created in the current working directory. At first, we are in the directory called touch1 and we can see that there is nothing. Then, we execute the touch command in order to create a directory named apple and we can see that the apple directory is created. The created file has the expected permissions.

2nd verification case

Objective: To check if the command is executed with selecting an existing file.
Input: `touch apple`
Previewed output: It does not need to do anything.

Verification output:



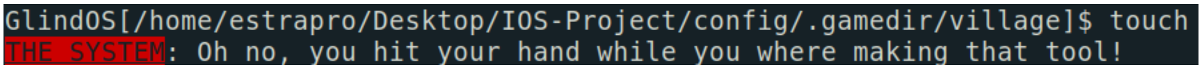
```
[acaf0205@dif-linuxserver touch1]$ ls -l
total 0
-rw-r--r--. 1 acaf0205 acaf 0 Apr 21 09:27 apple.txt
[acaf0205@dif-linuxserver touch1]$ ../touch apple.txt
touch executed successfully
[acaf0205@dif-linuxserver touch1]$ ls -l
total 0
-rw-r--r--. 1 acaf0205 acaf 0 Apr 21 09:27 apple.txt
```

Explanation: The command is executed correctly but the file already exists. As we can see, the file doesn't change even if we try to create the same file.

3rd verification case

Objective: To check if we try to execute the `touch` command incorrectly, it will appear an orientative message.
Input: `touch`
Previewed output:
`THE SYSTEM: Oh no, you hit your hand while you where making that tool!`

Verification output:



```
GlindOS[/home/estrapro/Desktop/IOS-Project/config/.gamedir/village]$ touch
THE SYSTEM: Oh no, you hit your hand while you where making that tool!
```

Explanation: The command is not used correctly, so with this message, the user should know something is not correct.

2.3 APPENDIX C

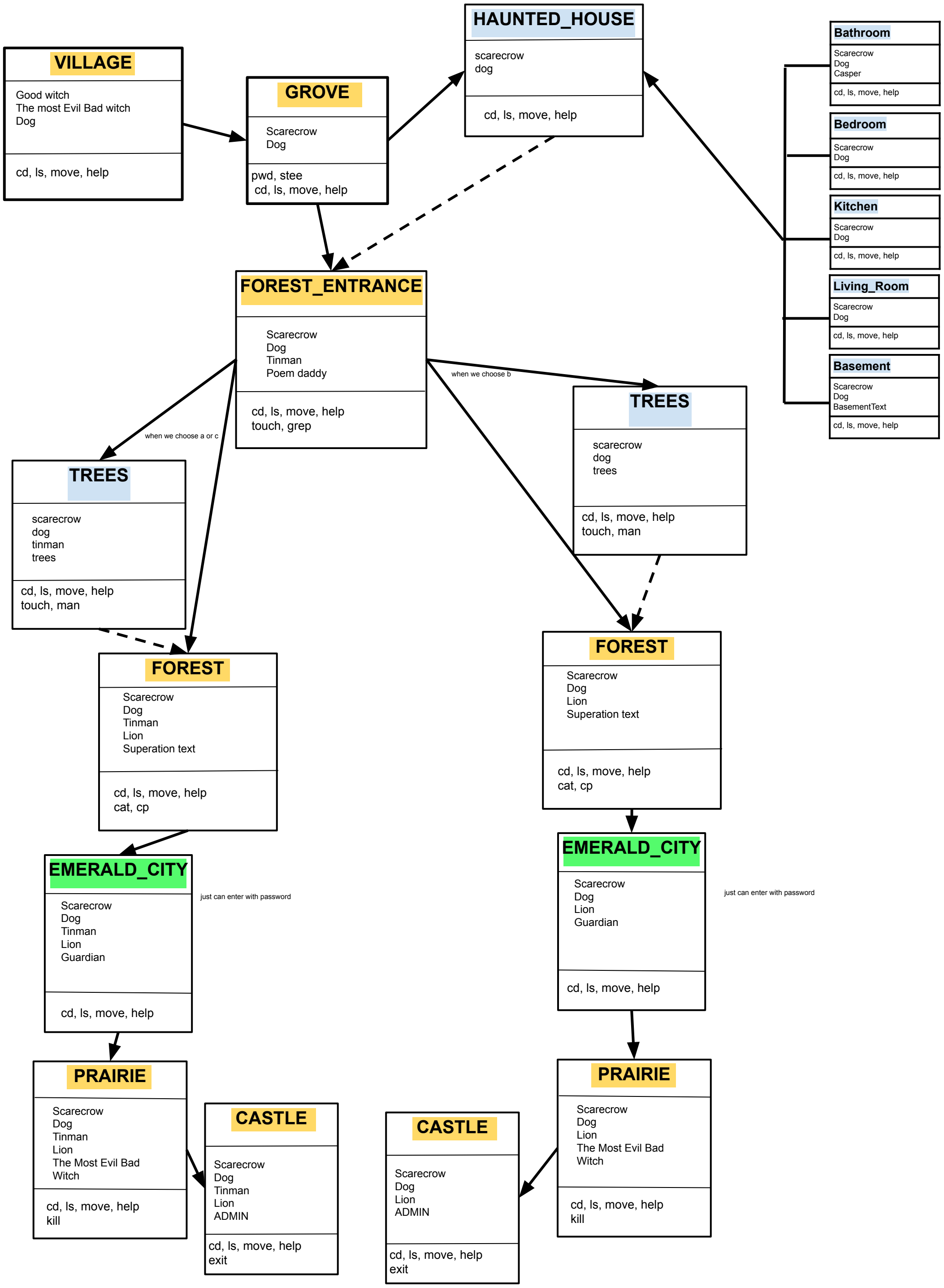


Figure 2.1: File Tree

3. Conclusions

On the one hand, talking about the subject's theoretical perspective, we have improved our C language skills learning and using some system calls, that are some powerful tools provided by the Operating Systems.

On the other hand, we have completed the objective of the development of the project, that was to put into practice the theoretical knowledge acquired in the subject and to develop our ability to program using the programming interface offered by the Linux OS.

To sum up, we can say that we have learned a lot of different concepts and notions, but also to teamwork, despite all the obstacles that the pandemic has placed on us.

4. Bibliography

[1] Colour text in C

[2] `ls` command

[3] How to use `execvp()`

[4] How to use `execlp()`

[5] Poem used in the game Daddy

[6] Pipes