

HAL 9000 System



Introducció a les pràctiques de Sistemes Operatius

Un sistema operatiu està format, bàsicament, per 4 mòduls o subsistemes:

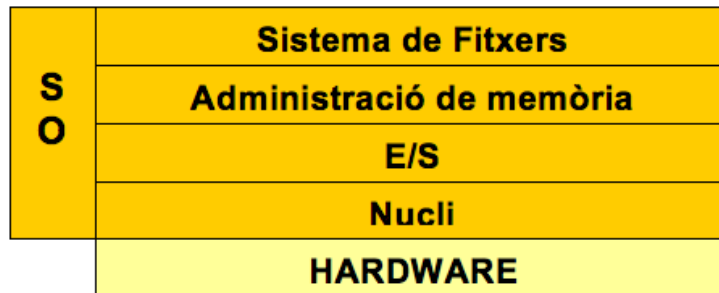


Figura 1. Arquitectura per capes d'un Sistema Operatiu.

NUCLI. És l'única capa que pot inhibir interrupcions. Les seves funcions bàsiques són la representació dels processos en execució, el control de la concurrència de processos, la gestió i control de les interrupcions, i dotar al sistema de mecanismes de comunicació entre processos, de sincronització i exclusió mútua.

SISTEMA E/S. És l'única capa que pot executar instruccions del tipus entrada i sortida. La seva funció és la comunicació amb els perifèrics. Per tant, les capes superiors, per a dialogar amb el *hardware*, ho han fer a través d'aquesta capa.

GESTIÓ DE MEMÒRIA. Conjuntament amb el *hardware*, crea la possibilitat de disposar de memòria virtual. A més a més, aquesta capa és la responsable de garantir (també conjuntament amb el *hardware*) la protecció de les dades a memòria i també la seva compartició.

SISTEMA DE FITXERS. Dota al sistema d'una visió estructurada de les dades emmagatzemades al disc.

Les aplicacions d'usuari són programes creats per l'usuari. Quan aquests programes necessiten alguna funcionalitat del sistema operatiu, realitzen el que s'anomena una "crida al sistema", que consisteix en fer crides a funcions que ofereixen les diferents capes. És a dir, si l'aplicació vol mostrar un caràcter per pantalla, haurà de cridar a una funció de la capa d'E/S per a aconseguir-ho.

El contingut de la pràctica de l'assignatura de Sistemes Operatius està orientat a l'aprenentatge d'una arquitectura distribuïda mitjançant la diferents mecanismes de comunicació (*sockets* principalment), posant de relleu la problemàtica de la concurrència de processos i la multitasca. L'objectiu principal és establir connectivitat entre els diferents equips i permetre la transmissió d'informació entre els diversos nodes d'un sistema distribuït. A més, cal la utilització de mecanismes del nucli del

sistema operatiu (memòria compartida, mètodes d'exclusió mútua, sincronització i creació de processos) per a poder resoldre la pràctica.

NOTA 0. Les quatre pràctiques del curs 2004-2005 rebien els noms de *Arda*, *Arda++*, *Harkonnen* i *IluvatarSon*. Tot era en honor a Frank Herbert i la seva famosa novel·la *Dune*. Força alumnes van recordar l'origen i van descobrir d'on provenia el nom.

NOTA 1. Les quatre pràctiques del curs 2005-2006 rebien els noms de *Oedipus Rex*, *Sphinx*, *Antigona* i *Teiresias*. Com totes les pràctiques de SO, no hi ha res a l'atzar. Els noms provenien de la tragèdia d'Èdip de Sòfocles i dotaven de sentit a les pràctiques. S'iniciava el curs amb una pràctica 1 senzilla, Èdip Rei. Èdip viu feliç com a rei mentre ignora que ha matat el seu pare i s'ha casat amb la seva mare. La ignorància el fa feliç. Els alumnes de SO viuen feliços perquè la pràctica 1 és fàcil i no saben que els espera. La pràctica 2 és l'Sphinx. Èdip ha de superar la prova de l'Sphinx ja que, sinó, aquesta el matarà. Òbviament, la pràctica 2 del curs 2005-2006 era la més difícil de totes i si no la superaves no aprovaves l'assignatura. La pràctica 3 es deia Antigona, qui ajuda a Èdip un cop aquest cau en desgràcia i en la desesperació. Antigona era la salvació dels alumnes que volien anar a l'examen de juny i necessitaven una pràctica llarga lliurada. Us estalvio els detalls de Teiresias. Tot i que molts alumnes van seguir els noms i les referències, cap d'ells va acabar de copsar el sentit de la tragèdia...

NOTA 2. Les pràctiques del curs 2006-2007 es van centrar en l'obra mestra de Dante Alighieri, "La Divina Commedia". Les pràctiques es deien "Inferno", "Purgatorio" i "Paradiso". Suposo que no calen gaire comentaris per entendre què passa a la pràctica 1. La segona pràctica es suavitzava però la necessites si vols presentar-te a examen. Finalment, la tercera és relativament curta i senzilla, un paradís comptant amb tot el que has fet anteriorment.

NOTA 3. Les pràctiques del curs 2007-2008 es van centrar en l'obra mestra dels germans Wachowski, "Matrix", on els humans (alumnes) han de lluitar contra les màquines per a sobreviure (aprovar l'assignatura). Les pràctiques es deien "Matrix", "El Ferroviario" i "Keymaker". Matrix era el servidor central on s'havien de connectar els clients, com *Nebuchadnezzar*, per a poder intercanviar arxius i comunicar-se mitjançant un xat distribuït. També podien connectar-se amb altres dimonis, com *Oraculo* o *Link*, que els donaven consells. *El Ferroviario* era una pràctica de simulació de càrrega de processos i administració de memòria. Si coneixeu el paper de *El Ferroviario* a la pel·lícula *Matrix* veureu que la relació és directa. Finalment, la codificació de fitxers en EXT2. Vam considerar que un bon nom per descodificar era el personatge de *Matrix* anomenat *Keymaker*.

NOTA 4. Les pràctiques del curs 2008-2009 es van centrar en l'obra mestra de Isaac Asimov, "La fundació". Les pràctiques es deien: "Trantor", el planeta central de la galàxia, tenint, per tant, la mateixa magnitud que la pràctica 1 de Sistemes Operatius; "Trevize, the loader", nom del conseller de la primera fundació, amb una intuïció insòlita, però amb perilloses intencions; i "Pelorat, the file reader", nom del professor d'història que ajudà a Trevize a trobar el planeta Terra, la llar on descansar després d'aprovar les tres pràctiques de Sistemes Operatius.

NOTA 5. Les pràctiques del curs 2009-2010 es van centrar en el grup australià de *hard rock* AC/DC, com a homenatge a la marxa del que va ser professor de pràctiques de l'assignatura els tres darrers anys, Hugo Meza. La primera pràctica, sota el nom de *Highway to shell* (petita modificació del nom de la cançó més popular del grup), conduïa als alumnes per tres fases. La primera, *Welcome to the Shell*, donava la benvinguda a l'infern als alumnes amb la implementació d'un intèrpret de comandes *shell*, sota el nom de Malcolm (cantant del grup). A la segona fase, per tal de fugir de les tenebres en què es trobaven, els alumnes es van veure pactant amb el diable (*Dealing with the Devil*), on havien de seguir tot un protocol per a comunicar-se amb el dimoni Angus (guitarrista del grup). Finalment, quan creien que tot havia passat, es topaven amb *Ballbreaker* (disc que van gravar al 1995), nom que no requereix gaires explicacions. La segona pràctica del curs 2009-2010 rebia el nom de *Back in Black*, nom del disc llençat al 1980, on els alumnes havien d'identificar el format d'un volum donat i extreure'n informació.

NOTA 6. Les pràctiques del curs 2010-11 es van centrar en les obres del polèmic *Donatien Alphonse François de Sade*, més conegut com "el Marquès de Sade". Així, la pràctica es titulava "*Les 120 journées de Sodome*". Aquesta estava dividida en tres fases: "*Le Château de Silling*" era la primera d'elles, on els alumnes gestionaven el comportament del client de l'aplicatiu, el qual rebia el nom del personatge de la novel·la "Blangis". Posteriorment els alumnes s'endinsaven al castell en una segona fase anomenada "*Les quatre madames*", on Blangis es connectava a un dimoni anomenat "Thérèse". Finalment, la cosa es desmadrava a la tercera fase, on els alumnes havien de crear el servidor "*Libertinage*", que, com el seu nom indica, és on començava el "festival".

NOTA 7. Les pràctiques del curs 2011-12 es van centrar en la mundialment coneguda sèrie televisiva del Simp(so)ns. Aquesta pràctica estava dividida en cinc fases: "La shell de Homer", "el bar de Mou", "Clancy Wiggum", "Living in Springfield" i "Living in Springfield++", en el seu conjunt aquestes fases formaven un sistema que permetia executar comandes de forma local, algunes pròpies en un servidor remot i l'activació de diversos serveis que mostraven frases mítiques de la sèrie, tot això seguint una arquitectura client - servidor.

NOTA 8. La pràctica del curs 2012-13 es va anomenar LsBox. El motiu va ser força simple: es tractava de dissenyar i implementar un sistema molt similar (de fet simplificat) del conegut Dropbox. L'únic detall curiós era el logotip de la pràctica que portava un mapa d'Austràlia encobert doncs era un petit homenatge a un monitor de SO que deixava aquestes tasques.

NOTA 9. La pràctica del curs 2013-14 es va anomenar LsHangIn. El motiu era que era una versió simplificada del conegut Google Hangouts: es tractava de dissenyar i implementar un sistema de xats amb múltiples sales per a usuaris. A més, cadascuna de les Fases tenien relació amb la pel·lícula *Resacón en las Vegas*.

NOTA 10. La pràctica del curs 2014-15 es va anomenar Gekko. Gekko és un empresari i principal protagonista de la pel·lícula *Wall Street*, relació directa amb la Borsa i l'objectiu de la pràctica. Però, a més, hi havia diversos homenatges més: *TumblingDice*, el generador de fluctuacions, és

també una cançó dels Rolling Stones i Dozer, l'operador de Borsa, és un dels personatges de Matrix.

NOTA 11. La pràctica del curs 2015-16 es va anomenar LsTransfer, the force awakens. Era un clar homenatge al retorn de la saga d'Starwars. A més el servidor es deia Naboo (planeta que surt en diferents episodis d'Starwars) i els clients Gungan (habitants d'aquest planeta).

NOTA 12. La pràctica del curs 2016-17 es va anomenar LsTinder, may the Love be with you. Era per la clara similitud a la xarxa social a la qual la pràctica feia referència i el lema un altre homenatge a la saga Starwars. A més, els diferents processos es deien Rick i Morty en referència a una sèrie americana de televisió d'animació per adults.

NOTA 13. La pràctica del curs 2017-18 es va anomenar LsEat, may the Food be with you. Primerament, el nom en referència a la moda del Just Eat. El lema un altre homenatge a la saga Starwars, en la seva imminent estrena de l'Episodi VIII, The Last Jedi. A més, els diferents processos tenien noms d'homenatge a StarTrek. Picard i Data personatges i Enterprise, la nau.

NOTA 14. La pràctica del curs 2018-19 es va anomenar Cosgrove System, Stairway to heaven. Primerament, Cosgrove és el cognom de la família protagonista d'una pel·lícula mítica de Peter Jackson: Braindead, considerada un clàssic de les pel·lícules gore de l'època. El lema, Stairway to Heaven, no era res relacionat amb els observatoris de la pràctica sinó un homenatge a la famosa cançó de Led Zeppelin. A més, els diferents processos tenien noms dels protagonistes de la pel·lícula Braindead: McGruder, McTavish, Paquita i Lionel.

NOTA 15. La pràctica del curs 2019-20 es va anomenar Cypher System era un homenatge de nou a Matrix, doncs s'està gravant pel 2022 Matrix IV, un mite en el món de la ciència ficció. L'script de la portada si l'executàveu feia les lletres verdes caient per la pantalla de Matrix. Trinity és la protagonista principal de Matrix i els exemples sempre són noms de personatges de la saga. No hi ha res a l'atzar.

NOTA 16. La pràctica del curs 2020-21 es va anomenar Overlook System. Això era un homenatge a la mítica pel·lícula de El Resplandor (the Shining) d'Stanley Kubrick on el 2020 feia 40 anys de la seva estrena. L'Hotel s'anomenava Overlook i la imatge de la portada de la pràctica era la d'un dels seus mític passadissos. Els diferents processos del sistema a dissenyar eren Jack, Wendy, Danny i Lloyd que són noms de personatges de la saga. No hi ha res a l'atzar.

NOTA 17. La pràctica del curs 2021-22 es va anomenar Arrakis System. Això era un homenatge a la mítica pel·lícula de Dune de Denis Villeneuve del 2021, però pensant amb l'original de David Lynch de 1984. Arrakis és el planeta on es desenvolupa la gran part de la pel·lícula. Hi havia els Atreides (els bons), els Harkonnen (els dolents) i els Fremen (la raça nativa del planeta Arrakis) que són elements importants i claus a la mítica pel·lícula.

NOTA 18. La pràctica del curs 2022-23 es va anomenar Eä System. Això era un homenatge a la sèrie El Senyor dels Anells: Els anells de poder (originalment en anglès, The Lord of the Rings: The Rings of Power). És una sèrie de televisió basada en la novel·la El Senyor dels Anells i els seus apèndixs de J. R. R. Tolkien. Transcorre milers d'anys abans d'El hòbbit i El Senyor dels Anells de Tolkien a la Segona Era de la Terra Mitjana. Està produïda per Amazon Studios.

HAL 9000 System

En els inicis de la revolució digital de la música, les empreses Napster i Spotify van canviar la forma en què les persones escoltaven música. Napster va ser pionera permetent la compartició de fitxers MP3 entre usuaris a través d'Internet, mentre que Spotify va introduir el concepte de l'streaming de música legal.

Veient l'èxit empresarial d'Spotify, diferents competidors volen també entrar al mercat, tot proposant millors solucions i arquitectures diferents per tal que els usuaris puguin gaudir del mateix servei sense, per exemple, haver de pagar subscripcions o haver de compartir comptes amb amics, amigues, parelles o, fins i tot, ex-parelles.

Amb la gran reputació que tenen els alumnes de Sistemes Operatius, les empreses ens han contactat per a poder desenvolupar una primera versió del futur competidor de Spotify... SO Team al rescat!

El sistema a implementar ha estat anomenat HAL 9000 per l'empresa que ens ha contractat. El seu principal enfocament és la facilitat d'ús de cara a l'usuari, juntament amb la rapidesa i fiabilitat del sistema per a així poder captar fàcilment nous clients. És per això que caldrà implementar una arquitectura i infraestructura tecnològica, on caldrà configurar, dissenyar i implementar un sistema de comunicacions per a poder gestionar música. Partim d'un conjunt de dispositius que tenen instal·lat un sistema operatiu Linux i la xarxa TCP/IP.

L'ecosistema a dissenyar es basa en disposar de diferents servidors físics, on cadascun tindrà música disponible per tal que els usuaris la puguin descarregar. Aquests servidors presenten ubicacions físiques desconegudes i només podrem accedir a ells mitjançant una IP i un port. Els usuaris es connectaran a aquests servidors i els podran fer peticions com ara descarregar cançons, llistes de reproducció, o veure tota la música disponible al servidor.

Caldrà doncs que dissenyem i implementem de nou tot el software de l'ecosistema segons les especificacions de disseny, requisits i restriccions que ens faran arribar. Això sí, ara amb robustesa, eficiència i escalabilitat. Per facilitar-nos el treball, el disseny ens vindrà donat per fases incrementals. En cadascuna d'elles s'especifiquen les seves funcionalitats, les seves limitacions i es dona un conjunt d'exemples d'execució i *testing*.



Figura 1. Logo de Napster, un programa pioner en l'intercanvi de fitxers

Descripció General

Una primera aproximació funcional de l'arquitectura de processos del sistema seria la que es presenta la Figura 2 tot i que, evidentment, l'arquitectura de disseny interna serà molt més complexa.

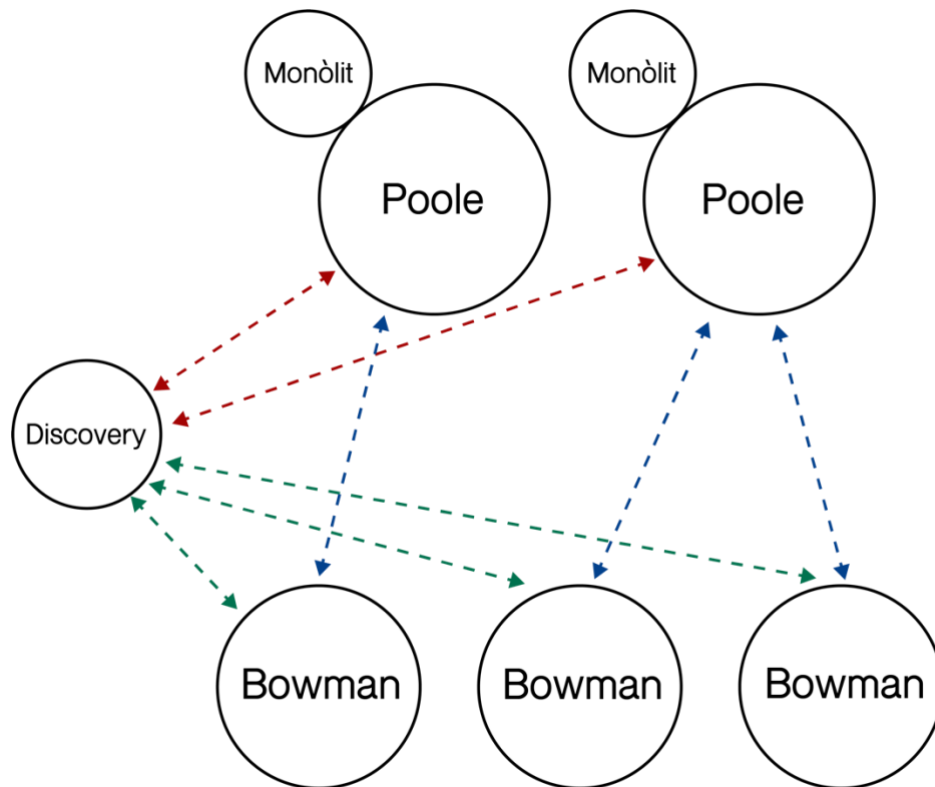


Figura 2. HAL 9000 System: esquema funcional.

El **processos Poole** actuaran de servidors que emmagatzemaran la música disponible per a poder ser descarregada. Els **processos Bowman** seran els diferents usuaris de l'aplicació, els quals es voldran connectar per a veure la música disponible i poder obtenir les cançons o llistes de reproducció que desitgin. Per a evitar la possible saturació d'algun dels servidors, s'ha proposat una millora de l'arquitectura amb un tercer procés, el **procés Discovery**, el qual actuarà com a un balancejador de càrrega dels diferents servidors **Poole**. D'aquesta manera, quan un usuari vulgui connectar-se al sistema **HAL 9000**, aquest es connectarà al servidor **Discovery** i, posteriorment i de manera transparent per a l'usuari, efectuarà connexió amb el servidor **Poole** amb menys usuaris connectats en aquell moment (el balancejador de càrrega decidirà a quin servidor de música es connectarà l'usuari).

En resum, l'objectiu principal del sistema és que els diferents processos Bowman puguin comunicar-se amb un servidor de música (Poole), on puguin fer peticions i descarregar fitxers MP3. Més endavant es detallarà aquest funcionament.

Per poder dissenyar i implementar la pràctica es recomana que es faci una **lectura completa de l'enunciat**. Per facilitar la seva realització s'han planificat 4 fases:

1. A la **fase 1** es crearà el procés Bowman i el procés Poole. Del procés Bowman, es processarà el fitxer de configuració i es realitzarà tota la implementació del *shell* amb les comandes esteses corresponents. Del procés Poole, només caldrà processar el fitxer de configuració.
2. A la **fase 2** s'implementarà el procés Discovery, juntament amb les diferents connexions entre servidors i el balancejador de càrrega. També es faran les primeres trames entre processos.
3. A la **fase 3** es realitzarà tota la comunicació entre els diferents processos, incloent la descàrrega de música, playlists... i les desconnexions de clients.
4. A la **fase 4** s'implementarà un cinquè procés, el Monòlit, el qual s'encarregarà de computar diferents estadístiques dels servidors de música, i mantenir aquestes estadístiques persistents entre execucions.

Fase 1: Just warming up

En la Fase 1 es començarà el disseny i programació de Bowman i de Poole. En ambdós processos, primer de tot caldrà processar el fitxer de configuració, el nom del qual es rep per paràmetre.

A més a més, el procés Bowman funcionarà en modus línia de comandes i tindrà programades un conjunt reduït de funcions (Veure Taula 1).

Fitxer de configuració - Bowman

Aquest fitxer de text tindrà diversos camps:

- Nom de l'usuari que executa.
- Nom de la carpeta on es troben els arxius de l'usuari.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on es troba el servidor Discovery.

Exemple de fitxer de text de configuració a la Figura 4:

```
Floyd
/floyd
192.168.50.51
8010
```

Figura 3. Fitxer de configuració d'un procés Bowman.

Fitxer de configuració - Poole

Aquest fitxer de text tindrà diversos camps:

- Nom del servidor de música.
- Nom de la carpeta on es troben els arxius del servidor.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on es troba el servidor Discovery.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on s'obrirà el servidor Poole.

Exemple de fitxer de text de configuració a la Figura 4:

```
Smyslov
/smyslov
192.168.50.51
8011
192.168.50.51
8015
```

Figura 4. Fitxer de configuració d'un procés Poole.

Seguidament caldrà dissenyar i implementar tota la lògica de programa que interpreta totes les comandes, les quals permeten el funcionament de Bowman. Aquestes comandes són *case insensitive*.

COMANDA	DESCRIPCIÓ
CONNECT	Connecta l'usuari al sistema. Primer es connectarà al balancejador de càrrega (Discovery) i, automàticament i de manera transparent per l'usuari, a un servidor Poole.
LOGOUT	Desconnecta l'usuari del sistema
LIST SONGS	Llista totes les cançons disponibles del servidor Poole on està connectat l'usuari.
LIST PLAYLISTS	Llista totes les llistes de reproducció disponibles del servidor Poole on està connectat l'usuari.
DOWNLOAD <SONG/PLAYLIST>	Descarrega una cançó o una llista d'aquestes.
CHECK DOWNLOADS	Consulta l'estat de les descàrregues en curs, en forma de barres de progrés
CLEAR DOWNLOADS	Neteja les descàrregues ja acabades de la llista de descàrregues.

Taula 1. Llistat de comandes esteses.

```
$matagalls:> Bowman config.dat

Floyd user initialized
$ DOWNLOAD MACARENA.MP3
Cannot download, you are not connected to HAL 9000
$ CONNECT PLEASE
Unknown command
$ CONNECT
Floyd connected to HAL 9000 system, welcome music lover!
$ CHECK DOWNLOADS
You have no ongoing or finished downloads
$ LIST SONGS
There are 6 songs available for download:
1. Macarena.mp3
2. Walk_of_life.mp3
3. Levels.mp3
4. Less_is_more.mp3
5. Stand_up.mp3
6. Isla_nostalgia.mp3
$ DOWNLOAD Isla_nostalgia.mp3
Download started!
$ something else
ERROR: Please input a valid command.
$ LIST PLAYLISTS
There are 2 lists available for download:
1. Pim_pam_trucu_trucu
   a. Levels.mp3
   b. Stand_up.mp3
   c. Macarena.mp3
2. Copeo_pre_costa
   a. Macarena.mp3
   b. Isla_nostalgia.mp3
$ DOWNLOAD Pim_pam_trucu_trucu
Download started!
$ CHECK DOWNLOADS
Isla_nostalgia.mp3                96% |===== % |
Pim_pam_trucu_trucu - Levels.mp3  61% |===== % |
Pim_pam_trucu_trucu - Stand_up.mp3 45% |===== % |
Pim_pam_trucu_trucu - Macarena.mp3 13% |== % |
$ LOGOUT
Thanks for using HAL 9000, see you soon, music lover!

$matagalls:>
```

Figura 5. Execució d'un procés Bowman.

Es demana:

- Que dissenyeu i programeu Bowman.c (nom del programa que farà servir els clients).
- Que programeu Poole.c (nom del programa que faran servir els servidors de música).
- Bowman primer processa el fitxer de configuració i guarda la seva informació en una estructura adequada.
- Bowman haurà de reconèixer (però no implementar en aquesta fase) les comandes pròpies descrites anteriorment.
- Poole haurà de processar el fitxer de configuració i guarda la seva informació en una estructura adequada.

Consideracions Fase 1:

- Un cop finalitzada l'execució dels processos Bowman i Poole s'ha d'**alliberar** tot tipus de memòria dinàmica significativa si n'hi hagués.
- El nom dels usuaris no pot contenir '&'. En cas que en continguin, s'hauran d'eliminar. Això és degut al protocol de comunicació del sistema, el qual està a l'Annex. Aquest serà *case sensitive*.
- Les comandes pròpies són *case insensitive*.
- Podem considerar que el **format del fitxer de configuració és correcte**.
- **No** es poden utilitzar les funcions *printf*, *scanf*, *gets*, *puts*, etc. Només es podrà interactuar amb la pantalla i amb fitxers amb les funcions *read* (lectura) i *write* (escriptura). Sí que es permet fer ús de la funció *sprintf* i similars.
- **No** es poden utilitzar les funcions *system*, *popen* ni cap variant.
- Cal garantir l'estabilitat de l'aplicació i el seu correcte funcionament. En cap cas es poden produir bucles infinits, *core dumped*, esperes actives, *warnings* al compilar, etc. També cal controlar tots aquells aspectes susceptibles de donar algun error i, en cas que es produeixin, **informar degudament a l'usuari** i, si és possible, seguir amb el funcionament normal de l'aplicació.
- Des d'ara i fins al final de la pràctica cal considerar que el procés Bowman pot finalitzar la seva execució utilitzant la comanda adequada o prement CTRL+C. Cal tenir-ho en compte i procurar perquè tot el sistema quedi el més **estable** possible en cas que es produeixi.
- És obligatori la utilització d'un *makefile* per generar l'executable.
- No és suficient que l'arxiu que pugeu al pou tingui extensió .tar, sinó que s'ha de poder desempaquetar amb la comanda tar. Qualsevol pràctica o *checkpoint* que no es pugui "descomprimir" d'aquesta manera **no serà corregida**.
- Recordeu que el codi ha d'estar degudament modulats, és a dir: no es pot ubicar tot en un sol fitxer .c, i és obligatori que hi hagi un fitxer *makefile*. Tingueu en compte que si en intentar corregir una pràctica aquesta no compila amb la comanda *make* (pel motiu que sigui), l'entrega serà qualificada com a no apta (2).

Fase 2: Feeling the connection

En aquesta segona fase caldrà implementar les connexions dels Bowman amb els servidors Poole. Com s'ha explicat breument en apartats interiors, un tercer procés (Discovery) actuarà de balancejador de càrrega per tal que els clients sempre es connectin al servidor menys saturat en aquell moment.

Com que els diversos processos Bowman, Poole i Discovery és evident que poden estar en servidors físics diferents, aquesta connectivitat caldrà dissenyar-la mitjançant **sockets**. Reviseu el **protocol de comunicació** als Annexos.

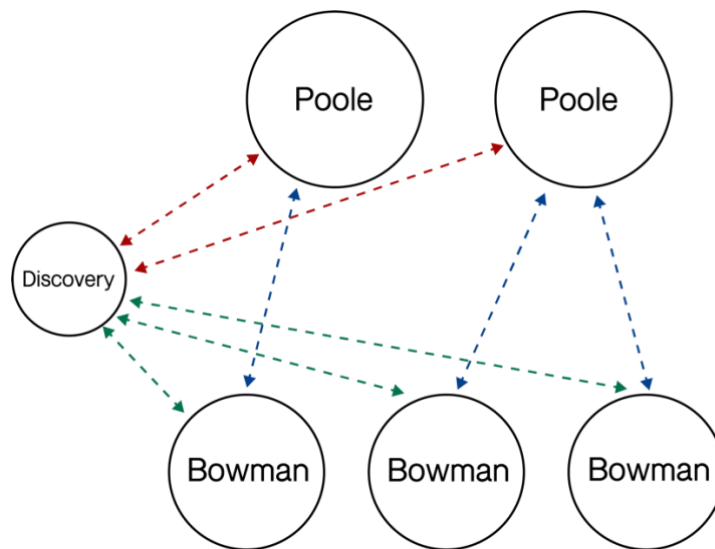


Figura 6. Esquema a dissenyar i implementar.

Els servidors Poole es connectaran a Discovery per a indicar que estan llestos per a rebre connexions de clients. Aquesta connexió serà momentània i només es farà al principi de la connexió dels processos Poole.

De la mateixa manera, els processos Bowman es connectaran també a Discovery per a notificar que volen entrar al sistema. Discovery respondrà aquesta connexió amb un nou servidor on connectar-se (el procés Poole ja registrat al sistema amb menys usuaris connectats), i el procés Bowman realitzarà la nova connexió (de manera automàtica). Només aquesta última connexió serà estable durant tota la seva execució ja que, com veurem, els dos processos intercanviaran trames durant l'execució del sistema.

El servidor Discovery estarà sempre en funcionament. Aquest no serà mai tancat ni rebrà interrupcions que puguin tallar la seva execució. D'altra banda, les possibles desconexions de Bowman i Poole sí que caldrà gestionar-les, però no en aquesta fase.

Dit això, caldrà implementar el següent:

1. El procés Discovery, el qual processarà un fitxer de configuració i començarà a rebre connexions de processos Poole i Bowman.
2. La connectivitat de Poole amb Discovery (no estable).
3. La connectivitat de Bowman amb Discovery (no estable).
4. La connectivitat de Bowman amb Poole (estable), juntament amb les comandes per a llistar els fitxers disponibles als servidors Poole.

Fitxer de configuració - Discovery

Aquest fitxer de text tindrà diversos camps:

- La IP i port (cadascun en una línia que acaba amb un '\n') d'on escoltarà connexions de processos Poole.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on escoltarà connexions de processos Bowman.

Exemple de fitxer de text de configuració a la Figura 4:

```
192.168.50.51
8010
192.168.50.51
8011
```

Figura 7. Fitxer de configuració d'un procés Discovery.

Connectivitat del sistema

A continuació s'exposa un exemple de connexions amb un entorn amb 1 procés Discovery (sempre hi haurà només 1, i sempre estarà encès), 2 processos Poole, i 3 processos Bowman.

1. Discovery comença la seva execució i escolta peticions de processos Poole i Bowman.
2. Els dos processos Poole es connecten al sistema, fent peticions a Discovery.
3. Es connecta un procés Bowman a Discovery. Aquest li respon amb la direcció IP i port (veure annex de protocol de trames) de qualsevol dels dos servidors Poole (els dos tenen 0 usuaris connectats). Discovery actualitza una llista d'usuaris connectats, tot apuntant que un client està connectat al Poole_1.
4. Es connecta un segon procés Bowman a Discovery. Aquest li respon amb la direcció IP i port del servidor Poole que encara no té cap usuari connectat. Discovery actualitza una llista d'usuaris connectats, tot apuntant que un client està connectat al Poole_2.

5. Es connecta un tercer procés Bowman a Discovery. Aquest li respon amb la direcció IP i port de qualsevol dels dos servidors Poole, ja que els dos tenen un client connectat cadascun. Discovery actualitza una llista d'usuaris connectats, tot apuntant que un client està connectat al Poole_2.

Una vegada es connecti Bowman a Poole, aquest podrà començar a llençar peticions. En aquesta fase cal implementar les comandes de:

1. LIST SONGS
2. LIST PLAYLISTS
3. LOGOUT

LOGOUT

És important tenir en compte que, de cara al logout, Bowman haurà d'avisar a Discovery, que és el que manté sempre la llista actualitzada d'usuaris i a quins servidors Poole estan connectats. De la mateixa manera, en aquesta fase caldrà gestionar quan el procés Bowman surt del sistema amb CTRL+C.

Dit això, podeu gestionar com vulgueu la desconexió de Bowman per a que Discovery actualitzi la seva llista:

1. Bowman avisa a Poole, el qual avisa a Discovery. Discovery, llavors, esborra a Bowman de la llista i envia un ACK a Poole. Poole envia un ACK a Bowman, el qual es desconnecta.
2. Bowman avisa a Discovery i a Poole, i espera la confirmació dels dos per a desconnectar-se.

Una vegada Bowman es desconnecti, torneu a tenir dues opcions d'implementació:

1. Acabar el procés
2. Tornar a posar a disposició la Shell, per si es vol tornar a connectar

Tanmateix, si un Bowman fa logout sense estar connectat, aquest acabarà la seva execució.

Les decisions de disseny del logout caldrà explicar-les de manera adient a la memòria de la pràctica, tot fent un diagrama de la comunicació implementada.

En aquesta fase, no caldrà gestionar la desconexió dels processos Poole. Podeu considerar que Poole i Discovery seran servidors que estaran sempre connectats, i no cauran durant l'execució. Per veure un exemple senzill de comportament podeu fixar-vos en la figura següent:

```
$matagalls:> Poole config.dat
Reading configuration file
Connecting Smyslov Server to the system..
Connected to HAL 9000 System, ready to listen to Bowmans petitions

Waiting for connections...

New user connected: Floyd.

New request - Floyd requires the list of songs.
Sending song list to Floyd

New user connected: Elena.

New request - Floyd wants to download Isla_Nostalgia.mp3
Sending Isla_Nostalgia.mp3 to Floyd

New request - Elena requires the list of playlists.
Sending playlist list to Elena

New request - Elena wants to download the playlist Copeo_pre_costa
Sending Copeo_pre_costa to Elena. A total of 2 songs will be sent.
...
```

Figura 8. Exemple execució Poole

Consideracions Fase 2:

- Les de la Fase 1 segueixen essent vàlides.
- En aquesta Fase NO s'ha d'implementar la transferència de fitxers (DOWNLOAD SONG I DOWNLOAD PLAYLIST) però sí LIST SONGS, LIST PLAYLIST i LOGOUT.
- Caldrà gestionar les caigudes dels processos Bowman, i mantenir l'estabilitat del sistema.
- Es pot considerar que Poole i Discovery no podran caure de cara a aquesta fase.

Fase 3: Pirates of the Caribbean

En aquesta fase s'acabarà d'implementar la connectivitat entre els diferents processos del sistema HAL 9000. Aquesta fase és l'encarregada de gestionar l'enviament de trames i fitxers entre els processos Bowman i Poole.

També caldrà implementar les comandes de CHECK DOWNLOADS i CLEAR DOWNLOADS, per tal de poder tenir un seguiment de les descàrregues en curs i completades des dels processos Bowman. També caldrà gestionar les caigudes controlades dels servidors Poole, per Control + C.

Recordeu que a l'Annex es troba el protocol de comunicació a implementar.

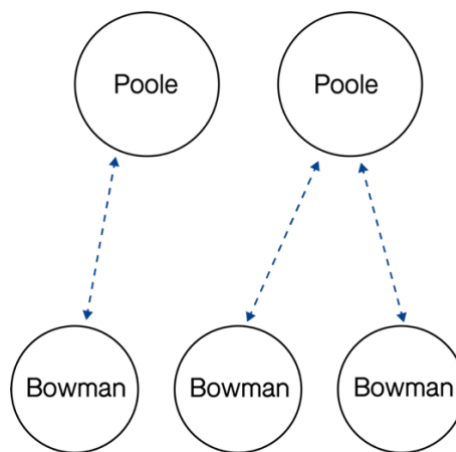


Figura 9. Comunicació a implementar a la fase 3, incloent totes les trames d'enviament de fitxers

Recordem alguns detalls ja explicats amb anterioritat que són rellevants:

- Amb la comanda `DOWNLOAD SONG_NAME`, Bowman demana a Poole que li envii una cançó que té Poole. Caldrà gestionar que l'usuari demani una cançó que no existeix.
- Amb la comanda `DOWNLOAD PLAYLIST_NAME`, Bowman demana a Poole que li envii TOTES les cançons d'una certa llista de reproducció que té Poole. Caldrà gestionar que l'usuari demani una playlist que no existeix.
- Amb la comanda `CHECK DOWNLOADS`, Bowman podrà veure les seves descàrregues en curs, juntament amb una barra de progrés indicant per on van, o si ja estan completades.
- Amb la comanda `CLEAR DOWNLOADS`, Bowman podrà netejar de la llista de descàrregues les cançons ja acabades de descarregar.
- En cas de rebre un fitxer amb el mateix nom d'un rebut anteriorment, aquest es sobreescrirà.

<pre> \$matagalls:> Bowman config.dat Floyd user initialized \$ CONNECT Floyd connected to HAL 9000 system, welcome music lover! \$ LIST SONGS There are 6 songs available for download: 1. Macarena.mp3 2. Levels.mp3 3. Stand_up.mp3 4. Isla_nostalgia.mp3 \$ LIST PLAYLISTS There are 2 lists available for download: 1. Pim_pam_trucu_trucu a. Levels.mp3 b. Stand_up.mp3 2. Copeo_pre_costa a. Macarena.mp3 b. Isla_nostalgia.mp3 \$ DOWNLOAD Isla_Nostalgia.mp3 Download started! \$ DOWNLOAD Pim_pam_trucu_trucu Download started! \$ CHECK DOWNLOADS Isla_nostalgia.mp3 100% ===== Pim_pam_trucu_trucu - Levels.mp3 61% ===== Pim_pam_trucu_trucu - Stand_up.mp3 45% ===== \$ CLEAR DOWNLOADS Pim_pam_trucu_trucu - Levels.mp3 73% ===== Pim_pam_trucu_trucu - Stand_up.mp3 65% ===== ... </pre>	<pre> \$montserrat:> Poole config1.dat Reading configuration file Connecting Smyslov Server to the system.. Connected to HAL 9000 System, ready to listen to Bowmans petitions Waiting for connections... New user connected: Floyd. ... New request - Floyd requires the list of songs. Sending song list to Floyd New request - Floyd requires the list of playlists. Sending playlist list to Floyd New request - Floyd wants to download Isla_Nostalgia.mp3 Sending Isla_Nostalgia.mp3 to Floyd New request - Floyd wants to download the playlist Pim_pam_trucu_trucu. Sending Pim_pam_trucu_trucu to Floyd. A total of 2 songs will be sent. ... </pre>
--	--

Figura 10. Exemple de comunicació entre un procés Bowman i un Poole

Consideracions Fase 3:

- Les de les fases anteriors segueixen essent vàlides.
- **El correcte enviament de fitxers s'haurà de verificar amb l'eina MD5SUM.**
- No es permès usar codi programat per fer l'MD5SUM. Cal executar la comanda md5sum que proporciona el mateix *bash* del sistema operatiu (md5sum).
- Quan un servidor Poole es desconnecti, el sistema haurà de detectar-ho i actualitzar degudament el balancejador de càrrega.

Opcionals Fase 3:

- Davant una caiguda d'un servidor Poole, reconnectar un a un tots els Bowman que hi havia connectats al servidor, tot preguntant a Discovery on s'han de connectar.
- Gestionar les caigudes de Poole i Bowman per interrupcions no gestionables (SIGKILL).

Fase 4: Teaching statistics to my children

Després d'implementar la tota la comunicació del sistema, se'ns ha demanat que el sistema HAL 9000 pugui calcular certes estadístiques del sistema. Per això implementarem un quart procés, el procés Monòlit.

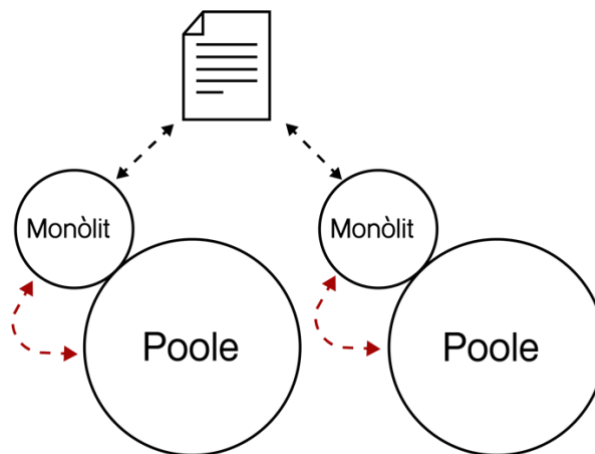


Figura 11. Implementació dels processos Monòlit, encarregats de computar estadístiques amb la informació que reben dels servidors Poole

Els processos Monòlit hauran de ser creats al principi de l'execució de Poole. És a dir, els programes Poole.c hauran de crear un subprocés (Monòlit) i s'hauran de comunicar amb ell per a poder enviar-li les dades pertinents per a fer les estadístiques.

La comunicació entre Poole i Monòlit NO ES PODRÀ FER AMB SOCKETS NI CUES DE MISSATGES. Caldrà que penseu una alternativa de comunicació.

Concretament, caldrà sempre tenir una llista de totes les cançons descarregades, juntament amb el nombre total de descàrregues per cançó.

Consideracions Fase 4:

- Les de les Fases anteriors segueixen essent vàlides.
- No es poden emprar sockets ni cues de missatges per a comunicar Poole i Monòlit.
- El fitxer a actualitzar serà únic, i s'anomenarà "stats.txt".
- Cal garantir l'exclusió mútua en l'escriptura del fitxer.
- No es pot realitzar la comunicació fent ús de memòria secundària.

Requeriments de lliurament i planificació

Aquesta pràctica disposarà de diferents **punts de control o checkpoints** per poder fer-ne un seguiment acurat i garantir la consolidació de cadascuna de les fases de manera incremental. Concretament se seguirà el següent calendari de dates límit:

CONCEPTE	DEADLINE
Fase 1	23/10/2023
Fase 2	20/11/2023
Fase 3	13/12/2023
Lliurament Final 1 (sobre 10)	08/01/2024
Lliurament Final 2 (sobre 10)	02/02/2024
Lliurament Final 3 (sobre 8)	20/05/2024
Lliurament Final 4 (sobre 6)	21/06/2024

Els *checkpoints* no són obligatoris, tot i ser altament recomanables per poder garantir la robustesa de la pràctica. Aquests *checkpoints* només serviran per incrementar linealment la qualificació de la pràctica. En cap cas penalitzaran la seva nota.

També és recomanable **validar el disseny global de la pràctica** amb els monitors de pràctiques abans d'iniciar les fases 2, 3 i 4. Així podreu garantir que la implementació no patirà d'inconsistències insalvables per fases posteriors. Per això cal aprofitar els horaris de dubtes i **portar els dissenys** impresos a validar. Cal recordar que una pràctica funcioni no és cap garantia que sigui vàlida, doncs pot ser que tot el disseny no compleixi els requisits de l'enunciat. Per això és molt important tenir garanties sobre el disseny a implementar.

Els lliuraments es realitzaran a l'eStudy en un pou amb un fitxer que inclogui el codi font (fitxers .c, .h i el makefile) de la pràctica, funcionant completament sobre Montserrat, així com els fitxers de dades i configuració utilitzats. El fitxer haurà de ser obligatòriament en format .tar. El podeu generar amb la comanda següent:

```
tar cf Gx_Fn.tar *.c *.h makefile *.ext_fitxers
```

on Fn és el número de Fase a lliurar. Per exemple, el grup 12 enviaria G12_F1.tar per al lliurament del *checkpoint* de la Fase 1.

En els lliuraments finals també caldrà dipositar-la **memòria en format PDF**. La memòria ha de constar, obligatòriament, dels punts que s'indiquen a continuació en l'Annex corresponent.

Annex I: Contingut de la memòria

La memòria ha de ser una documentació externa correctament maquetada. Ha de contenir els següents apartats:

1. **Portada**
2. **Índex**
3. **Disseny:** explicació de com s'ha dissenyat i estructurat la pràctica. Es pot explicar per fases o de la pràctica en global. Ha d'incloure de manera clara i entenedora:
 - a. Diagrames que expliquin processos s'han creat, les diferents comunicacions entre processos, etc.
 - b. Estructures de dades usades i la seva justificació.
 - c. Recursos del sistema utilitzats (*signals, sockets, semàfors, pipes*, etc.) amb la seva justificació.
 - d. Opcionals implementats.
4. **Problemes observats i com s'han solucionat.**
5. **Estimació temporal:** temps dedicat en el desenvolupament total de la pràctica per a cadascun dels estudiants. Les categories a tenir en compte són:
 - a. Investigació: temps dedicat a buscar eines i/o a aprendre components a implementar (fora del temps dedica a les sessions).
 - b. Disseny: temps dedicat a dissenyar la pràctica
 - c. Implementació: temps de codificació
 - d. *Testing*: temps dedicat a fer proves
 - e. Documentació: temps dedicat a escriure la memòria i a la documentació interna del codi (comentaris, etc.).
6. **Conclusions i propostes de millora.**
7. **(Opcional) Explicació de TOTS els noms dels processos de la pràctica. En quin tema ens hem basat aquest any per a fer l'enunciat? 😊**
8. **Bibliografia utilitzada** (en un format bibliogràfic correcte IEEE, APA, etc.): tant recursos bibliogràfics com links.

Es valorarà la redacció i la correctesa gramatical i ortogràfica. La memòria ha d'estar amb **numeració de pàgines**.

La memòria s'ha d'entregar en **format PDF** juntament amb el lliurament final però es recomana anar-la elaborant durant el transcurs de la pràctica.

Annex II: protocol de comunicació general

Per dur a terme la comunicació entre processos en diferents màquines, s'utilitzarà un protocol específic d'enviament de trames que s'explicarà a continuació. S'ha de tenir en compte que els missatges s'enviaran mitjançant *sockets* utilitzant una comunicació orientada a connexió.

En aquest protocol s'utilitzarà un **únic tipus** de trama. Sent aquestes de dimensió **fixa (256 Bytes)** i sempre formades pels següents 4 camps:

Type: Descriu el tipus de la trama en format hexadecimal 1 caràcter que contindrà un identificador de trama.

Header_Length: Camp de 2 bytes en format numèric que serveix per indicar la llargària del Header.

Header: Aquest camp serveix per a identificar la capçalera de la trama. La llargària d'aquest camp s'especifica en el camp previ HEADER_LENGTH

Data: Aquest camp serveix per emmagatzemar-hi valors o dades que ha d'enviar la trama. La llargària d'aquest camp depèn directament de la llargada total de la trama i de la mida del header, especificada al camp HEADER_LENGTH. Depenent del cas pot contenir diferents valors com s'especifica en cada funcionalitat.

TYPE	HEADER_LENGTH	HEADER	DATA
(1 Byte)	(2 Bytes)	(X Bytes)	(256 – 3 - X Bytes)

És molt important de cara a garantir la bona comunicació amb el procés que la definició de l'estructura de la trama de comunicació sigui estrictament, en ordre i format, la descrita amb anterioritat. Això vol dir que qualsevol canvi de tipus dels camps o ordre dels mateixos dins de l'estructura a dissenyar farà que les trames rebudes o enviades a procés no compleixin el protocol de comunicació i, per tant, no funcioni aquesta comunicació correctament.

Si la mida de les dades a enviar són inferiors a 256, caldrà omplir la trama amb el *padding* corresponent.

És del tot normatiu l'enviament de trames senceres en una única escriptura al socket.

NOVES CONNEXIONS A DISCOVERY - POOLE**Poole → Discovery**

Trama per demanar una connexió al servidor Discovery.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: NEW_POOLE
- DATA: [userName&IP&Port]

Discovery → Poole

Trama OK connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: CON_OK
- DATA: Buit

Trama KO connexió. S'envia en el cas que no s'hagi pogut establir la connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: [CON_KO]
- DATA: Buit

NOVES CONNEXIONS A DISCOVERY - BOWMAN**Bowman → Discovery**

Trama per demanar una connexió al servidor Discovery.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: NEW_BOWMAN
- DATA: [userName]

Discovery → Bowman

Trama OK connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: CON_OK
- DATA: [serverName&IP&Port]

Trama KO connexió. S'envia en el cas que no s'hagi pogut establir la connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: CON_KO
- DATA: Buit

NOVES CONNEXIONS A POOLE - BOWMAN**Bowman → Poole**

Trama per demanar una connexió al servidor Poole.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: NEW_BOWMAN
- DATA: [userName]

Poole → Bowman

Trama OK connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: CON_OK
- DATA: Buit

Trama KO connexió. S'envia en el cas que no s'hagi pogut establir la connexió.

- TYPE: 0x01
- HEADER_LENGTH: Llargària del header.
- HEADER: CON_KO
- DATA: Buit

LIST SONGS**Bowman → Poole**

Trama per a demanar la llista de cançons al servidor Poole.

- TYPE: 0x02
- HEADER_LENGTH: Llargària del header.
- HEADER: LIST_SONGS
- DATA: Buit

Poole → Bowman

Trama que respon a la petició de LIST SONGS d'un usuari.

Cal tenir en compte que la llista de cançons pot no caber-hi en una sola trama.

Podeu lliurement modificar el camp data de la trama per a informar de quantes songs en total hi ha, o modificar el header per a informar a Bowman de quantes trames ha d'esperar rebre.

- TYPE: 0x02
- HEADER_LENGTH: Llargària del header.
- HEADER: SONGS_RESPONSE
- DATA: *[song1&song2&song3&song4&song5...&songN]*

LIST PLAYLISTS**Bowman → Poole**

Trama per a demanar la llista de playlists al servidor Poole.

- TYPE: 0x02
- HEADER_LENGTH: Llargària del header.
- HEADER: LIST_PLAYLISTS
- DATA: Buit

Poole → Bowman

Trama que respon a la petició de LIST PLAYLISTS d'un usuari.

Cal tenir en compte que la llista de playlists i songs pot no caber-hi en una sola trama.

Podeu lliurement modificar el camp data de la trama per a informar de quantes lists i songs en total hi ha, o modificar el header per a informar a Bowman de quantes trames ha d'esperar rebre.

- TYPE: 0x02
- HEADER_LENGTH: Llargària del header.
- HEADER: PLAYLISTS_RESPONSE
- DATA: *[list1&song1&song2&song3#list2&song1&song2...#listN&song1&song2]*

DOWNLOAD SONG/PLAYLIST**Bowman → Poole**

Trama per a demanar una cançó a Poole.

- TYPE: 0x03
- HEADER_LENGTH: Llargària del header.
- HEADER: DOWNLOAD_SONG
- DATA: [song_name]

Bowman → Poole

Trama per a demanar una list a Poole.

- TYPE: 0x03
- HEADER_LENGTH: Llargària del header.
- HEADER: DOWNLOAD_LIST
- DATA: [playlist_name]

SEND FILE**Poole → Bowman**

Trama per a enviar un fitxer de música d'un servidor Poole a un procés Bowman.

Primer, s'enviarà la informació del fitxer. El camp ID és l'identificador de la cançó, ha de ser un número aleatori del 0-999.

- TYPE: 0x04
- HEADER_LENGTH: Llargària del header.
- HEADER: NEW_FILE
- DATA: *[FileName&FileSize&MD5SUM&ID]*

On:

- FileSize: en bytes expressat en format ASCII
- MD5SUM: 32 caràcters del MD5SUM

Posteriorment, s'enviarà el fitxer.

- TYPE: 0x04
- HEADER_LENGTH: Llargària del header.
- HEADER: FILE_DATA
- DATA: *[ID&dades_del_fitxer]*

Cal tenir en compte que caldrà enviar N trames de 256 bytes per a cada fitxer de música.

Bowman → Poole

Comprovació del MD5SUM correcta.

- TYPE: 0x05
- HEADER_LENGTH: Llargària del header.
- HEADER: CHECK_OK
- DATA: Buit

Comprovació de MD5SUM incorrecta.

- TYPE: 0x05
- HEADER_LENGTH: Llargària del header.
- HEADER: CHECK_KO]
- DATA: Buit

LOGOUT**Bowman → Poole**

Trama per notificar al Servidor d'una desconnexió.

- TYPE: 0x06
- HEADER_LENGTH: Llargària del header.
- HEADER: EXIT
- DATA: [userName]

Poole → Bowman

Trama OK desconnexió.

- TYPE: 0x06
- HEADER_LENGTH: Llargària del header.
- HEADER: CONOK
- DATA: Buit

Trama KO desconnexió.

- TYPE: 0x06
- HEADER_LENGTH: Llargària del header.
- HEADER: CONKO
- DATA: Buit

Procediment de detecció de trames errònies

Si qualsevol dels processos del sistema HAL 9000 rep una trama que no es correspon a cap dels formats definits caldria que enviés una trama de retorn per poder detectar l'error amb la següent trama:

- TYPE: 0x07
- HEADER_LENGTH: Llargària del header.
- HEADER: UNKNOWN
- DATA: Buit