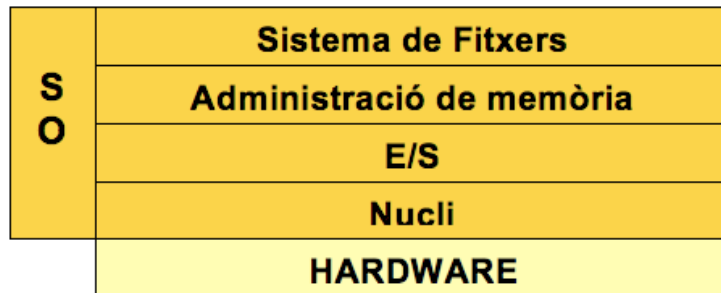


Overlook System



Introducció a les pràctiques de Sistemes Operatius

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



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 de 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 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 *Atreides*, *Atreides++*, *Harkonnen* i *Fremen*. 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 Ferroviari" 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 Ferroviari* era una pràctica de simulació de càrrega de processos i administració de memòria. Si coneixeu el paper de *El Ferroviari* 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.

A partir d'aquí, la interpretació dels noms i els conceptes de la pràctica d'aquest any (relacionant la dificultat, el contingut i l'assignatura) us pertoca a vosaltres...i esperem veure'ls a l'informe de la memòria!

Overlook System

Any que passa, els equips programadors de Sistemes Operatius de La Salle Campus Barcelona assoleixen una credibilitat i reputació incontestables. Així que els encàrrecs de grans corporacions arriben sense parar i cal anar seleccionant les peticions que més s'adeqüen a la tipologia de la matèria. Una corporació que vol romandre en l'anonimat ens ha encarregat poder crear un sistema, que vol anomenar **Overlook System**, que té per objectiu monitoritzar la majoria de serveis meteorològics que l'empresa controla.

Els socis de la corporació anònima, que anomenarem TS, ens demanen que dissenyem tot el **sistema Overlook** segons els requisits que ens especificaran detalladament i que aquest software el puguin integrar en el seu entorn de baix nivell. Cansats de les ineficiències d'entorns d'alt nivell, han optat per la nostra proposta d'utilitzar el llenguatge C com a element d'implementació de tot el sistema.

El sistema Overlook serà una novetat no tant per les funcionalitats a desenvolupar sinó per l'eficiència i eficàcia del seu disseny i posada en producció. El sistema Overlook estarà compost per 3 tipus d'elements funcionals clarament diferenciats: **2 servidors** que centralitzaran la monitorització de dades (**Jack** i **Wendy**) i les estacions meteorològiques (**Danny**) que s'encarreguen de recollir dades arreu del territori.

Una primera aproximació funcional que TS vol que tingui el seu sistema Overlook, tindria l'aspecte que es presenta la Figura 1 tot i que, evidentment, l'arquitectura interna serà molt més complexa.

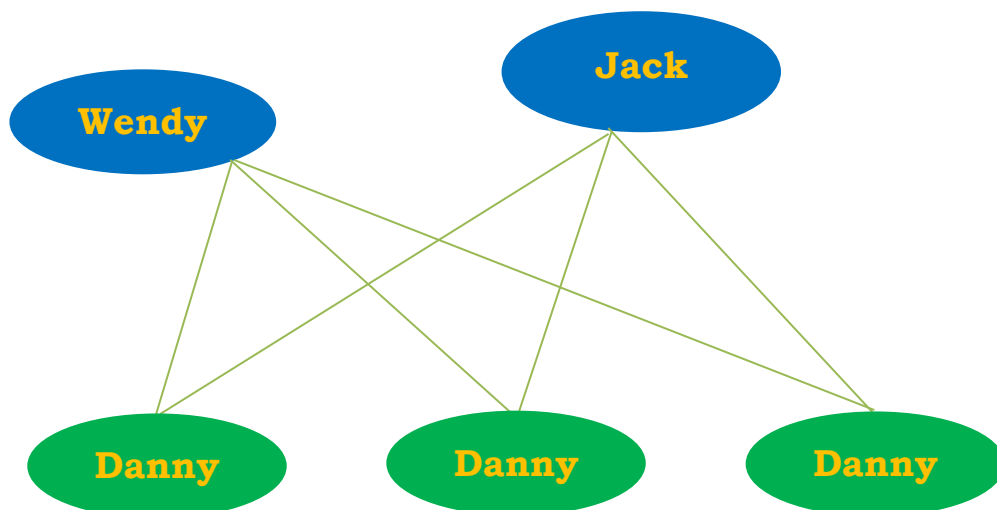


Figura 1. Overlook System: esquema funcional.

Descripció general del funcionament del sistema

El **sistema Overlook** està format pels dos mencionats servidors **Jack i Wendy** i per unitats replicades **Danny** que permeten recollir les dades a les estacions meteorològiques. Cal dissenyar el sistema per a que sigui escalable en quant a estacions.

Els 3 processos funcionen de manera totalment automatitzada, així que es minimitzarà la intervenció de l'usuari i d'aquesta manera es podrà evitar l'error humà. Tot i que la interacció amb els diferents components del sistema Overlook és pràcticament inexistent es recomana que es mostrin per pantalla les interaccions que aquest va realitzant per tal de poder debugar el sistema en la seva implementació. Cadascun dels 3 components, Jack, Wendy i Danny, tindran un fitxer de configuració associat que contindrà una sèrie de paràmetres importants per la seva execució i funcions.

Per poder intuir una primera execució del sistema la Figura 2 ens mostra un exemple:

```
montserrat$ Danny Config.dat

Starting Danny...

$Danny:
Testing...
No files available

$Danny:
Testing...
No files available

$Danny:

^C
Disconnecting Danny...

montserrat$
```

Figura 2. Execució d'un procés Danny.

L'objectiu general del **Sistema Overlook** és processar totes les dades que van recollint les diferents estacions meteorològiques. Segons quines siguin aquestes dades, dades atmosfèriques o imatges, les enviaran a un servidor o a un altre que s'encarregaran de fer estadístiques o emmagatzemar les imatges rebudes.

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 5 fases:

1. A la **fase 1** es crearà el procés Danny. Es processarà el fitxer de configuració i es realitzarà tota la captura de dades que provenen d'arxius de text o d'imatges JPG.
2. A la **fase 2** s'implementarà el servidor Jack i una part de la connectivitat amb els processos Danny.
3. A la **fase 3** s'acabarà el processament estadístic de les dades rebudes per Jack mitjançant la implementació d'un servei que realitza les estadístiques corresponents.
4. A la **fase 4** s'implementarà el servidor Wendy, la seva connectivitat amb els processos Danny i la transmissió dels arxius JPG amb les imatges.
5. La **fase 5** consistirà poder optimitzar i/o ampliar el sistema Overlook d'una manera més eficient. Aquesta fase serà **optativa**.

Fase 1: Starting Danny

En la fase 1 es començarà la programació de Danny. Primer de tot caldrà processar el fitxer de configuració, el nom del qual es rep per paràmetre.

Fitxer de configuració:

Aquest fitxer de text tindrà diversos camps:

- El nom de l'estació meteorològica que executa Danny.
- Nom de la carpeta on es troben els arxius a processar.
- El temps (en segons) per revisar si hi ha fitxers.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on es troba el servidor Jack.
- La IP i port (cadascun en una línia que acaba amb un '\n') d'on es troba el servidor Wendy.

Exemple de fitxer de text de configuració:

```
Igualada
/ArxiusDanny
30
192.168.50.52
8010
192.168.50.51
8011
```


Un cop arrencat l'estació meteorològica (veure Figura 1), Danny cada cert temps s'encarrega de revisar si hi ha arxius al directori indicat. En aquesta primera fase els detecta i els identifica mirant si és un arxiu de text (sempre tindrà .txt) o una imatge (sempre tindrà .jpg). En cas de ser un fitxer de text el processa, mostra per pantalla les dades capturades i elimina el fitxer de la carpeta. El fitxer té el següent format:

- Data (dd/mm/aaaa)
- Hora (hh:mm:ss)
- Temperatura (°C): enter amb 1 decimal
- Humitat (%): enter sense decimals
- Pressió atmosfèrica (hPa): enter amb 1 decimal
- Precipitació (mm): enter amb 1 decimal

Per veure una possible execució d'un procés Danny podeu observar la figura 3.

```
montserrat$ Danny Config.dat

Starting Danny...

$Igualada:
Testing...
No files available
$Igualada:
Testing...
3 files found
Dades.txt
Imatge1.jpg
Imatge2.jpg

Dades.txt
25/09/2020
15:30:58
24.5
73
1014.2
2.5

$Igualada:
^C
Disconnecting Danny...
montserrat$
```

Figura 3. Execució d'un procés Danny a la població d'Igualada.

Consideracions Fase 1:

- Un cop finalitzada l'execució del procés Danny s'ha d'**alliberar** tot tipus de memòria dinàmica si n'hi hagués.
- Podem considerar que el **format del fitxer de configuració és correcte**, així com els fitxers de dades atmosfèriques.
- Es pot considerar que el nom de l'estació meteorològica no ocuparà mai més de 100 caràcters.
- **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 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 Danny pot finalitzar la seva execució utilitzant la comanda adequada o prement per 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.

Fase 2: Connecting Jack

En aquesta segona fase caldrà poder fer un primer pas cap a la interconnectivitat de Danny (de les diverses estacions meteorològiques) amb el servidor Jack.

Com que els diversos processos Danny és evident que poden estar en màquines diferents aquesta connectivitat caldrà dissenyar-la mitjançant **sockets**. Reviseu el **protocol de comunicació** als Annexos.

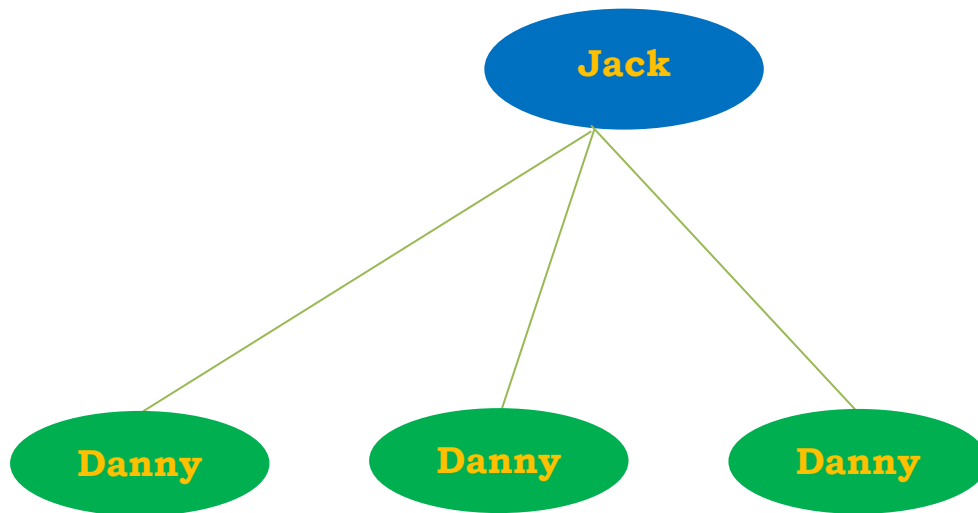


Figura 4. Esquema a dissenyar i implementar.

Com es pot veure en l'exemple a continuació, Danny es connecta al servidor Jack un cop s'ha iniciat i llegit el fitxer. Aquesta connexió serà estable durant tota la seva execució. Jack ha de mantenir totes les connexions amb totes les estacions meteorològiques Danny ja que aquestes poden enviar dades en qualsevol moment.

Quan una estació Danny processa un fitxer de dades atmosfèriques les enviarà al servidor Jack que, en aquesta segona Fase, tan sols les traurà per pantalla, indicant primer de tot de quina estació les ha rebut.

Quan el servidor Jack surt del sistema amb CTRL+C cal desconnectar-se dels altres processos Danny que s'havien connectat. Els processos Danny si perden la connexió també s'acabaran ja que no té sentit processar dades i no poder-les enviar.

Per veure un petit exemple senzill de comportament podeu fixar-vos en les Figures 5 i 6 simulant dos programes Danny i Jack al sistema Overlook.

```
montserrat$ Danny Config.dat

Starting Danny...
Connecting Jack...

$Igalada:
Testing...
No files available
$Igalada:
Testing...
3 files found
Dades.txt
Imatge1.jpg
Imatge2.jpg

Dades.txt
25/09/2020
15:30:58
24.5
73
1014.2
2.5

Sending data...
Data sent
$Igalada:
^C
Disconnecting Jack
Disconnecting Danny
montserrat$
```

Figura 5. Execució de Danny (Igalada).

```
montserrat$ Jack Config.dat

Starting Jack...

$Jack:
Waiting...
New Connection: Igualada

$Jack:
Receiving data...
Igualada
25/09/2020
15:30:58
24.5
73
1014.2
2.5

$Jack:
Waiting...
^C
Disconnecting Jack...
montserrat$
```

Figura 6. Execució de Jack.

Consideracions Fase 2:

- Les de la Fase 1 segueixen essent vàlides.
- Jack també tindrà un fitxer de configuració on es trobarà la seva IP i port.

Exemple de fitxer de text de configuració:

```
192.168.50.52
8010
```

Fase 3: Processing!

En aquesta fase, s'acabarà la implementació de les funcionalitats del servidor Jack afegint tot el processament estadístic de les dades que va rebent. Per això crearem un servei addicional, anomenat Lloyd, que depèn de Jack que realitzarà tots els càlculs necessaris.

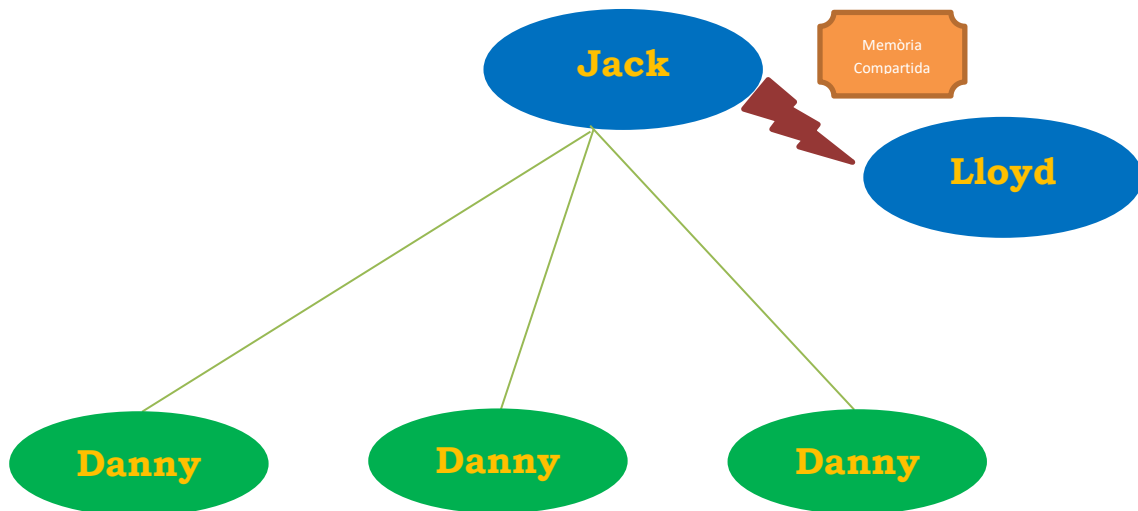


Figura 7. Un servei associat a Jack: Lloyd.

Cada cop que Jack rebi dades d'una estació Danny, aquest avisarà a Lloyd per a que les processi. Lloyd tindrà un array de les estacions on per cadascuna guardarà **la quantitat de lectures rebudes i les mitjanes aritmètiques dels 4 valors atmosfèrics**. Quan se l'avisí d'una nova dada actualitzarà l'estadística d'aquella estació a la llista. Cada 2 minuts reescriurà un **arxiu de text**, anomenat **Hallorann.txt**, que guarda les estadístiques a memòria secundària.

Consideracions Fase 3:

- Les de les fases anteriors segueixen essent vàlides.
- Tot l'emmagatzemament de les dades a Lloyd ha de ser dinàmic. No és permet guardar més informació a l'array d'estacions de la descrita anteriorment.
- Per crear el servei Lloyd des de Jack **no està permès** fer-ho amb un thread.
- Només està permès avisar al servei Lloyd mitjançant **mecanismes de sincronització**.
- Les dades meteorològiques que rep Jack (en el sentit més ampli del terme) de cada estació només poden ser transmeses a Lloyd mitjançant la utilització de **memòria compartida**. **Aquesta memòria compartida només tindrà la capacitat d'emmagatzemar una única lectura** (només podar guardar una única estructura amb 5 camps: el nom de l'estació i els 4 valors atmosfèrics).

Fase 4: Wendy

En aquesta fase es farà la implementació del servidor Wendy. Aquest servidor es dedica a rebre les **imatges** enviades des de les estacions meteorològiques i emmagatzemar-les en una carpeta repositori anomenada Barry. Wendy necessita també un arxiu de configuració que rep per paràmetre.

Exemple de fitxer de text de configuració:

```
192.168.50.51  
8011
```

Cada cop que arrenquem una estació meteorològica Danny aquesta farà dues connexions: al servidor Jack i al servidor Wendy. En aquest servidor Wendy s'hi enviaran les imatges trobades per cada Danny.

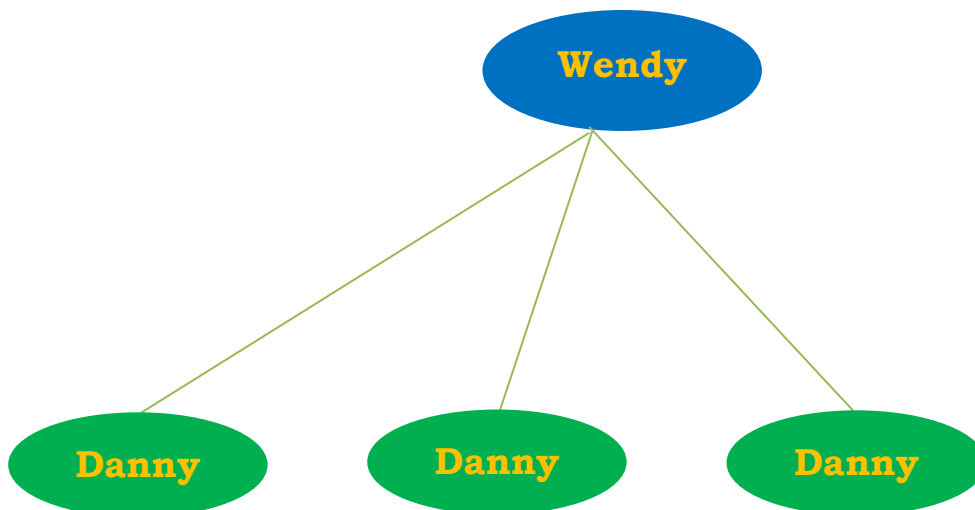


Figura 8. Disseny funcional de Wendy i Danny.

Així doncs, el funcionament del procés Danny quedarà modificat en l'exemple de la següent manera que mostra la Figura 9. Una possible representació que facilita l'execució del servidor Wendy seria la que es mostra en la Figura 10.

Com que els diversos processos Danny i el servidor Wendy és evident que poden estar en màquines diferents aquesta connectivitat caldrà dissenyar-la mitjançant sockets. Reviseu el protocol de comunicació a l'Annex. Cal respectar-lo escrupolosament.

```
montserrat$ Danny Config.dat
```

```
Starting Danny...
```

```
Connecting Jack...
```

```
$Igalada:
```

```
Testing...
```

```
No files available
```

```
$Igalada:
```

```
Testing...
```

```
3 files found
```

```
Dades.txt
```

```
Imatge1.jpg
```

```
Imatge2.jpg
```

```
Dades.txt
```

```
25/09/2020
```

```
15:30:58
```

```
24.5
```

```
73
```

```
1014.2
```

```
2.5
```

```
Sending data...
```

```
Data sent
```

```
Sending Imatge1.dat
```

```
Sending Imatge2.dat
```

```
$Igalada:
```

```
^C
```

```
Disconnecting Jack
```

```
Disconnecting Danny
```

```
montserrat$
```



```
montserrat$ Wendy Config.dat

Starting Wendy...

$Wendy:
Waiting...
New Connection: Igualada

$Wendy:
Receiving data from Igualada
Imatge1.dat
Imatge2.dat

$Wendy:
Waiting...
^C
Disconnecting Wendy...

montserrat$
```

Figura 10. Execució de Wendy en Fase 4.

Consideracions Fase 4:

- Les de les Fases anteriors segueixen essent vàlides.
- Es pot suposar que els fitxers JPG tenen noms diferents sempre.
- El correcte enviament d'imatges 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).
- Si es tanca Wendy de manera inesperada se seguirà el mateix protocol de quan es tancava Jack. Dit d'una altra manera, només té sentit tenir les estacions meteorològiques connectades si estan els dos servidors en funcionament.

Fase 5: Optimizing!

Aquesta darrera fase és **opcional**. Si s'ha realitzat un bon disseny de la pràctica, en el seu procés d'implementació us haureu adonat que la restricció de la Fase 3 de tenir una única estructura en memòria compartida pot arribar a ser un coll d'ampolla pel sistema si hi haguessin moltes actualitzacions de dades atmosfèriques.

De cara a poder optimitzar aquest punt del sistema es demana que feu una extensió, permetent que aquesta estructura sigui un **array d'estructures** limitat a un determinat nombre (que es podrà definir a l'inici de l'execució del programa Jack).

Així doncs, en arribar diferents actualitzacions de dades atmosfèriques de les estacions meteorològiques, els càlculs estadístics es podran fer de manera més ràpida, doncs podrem anar emmagatzemant aquestes dades a tractar en aquest nou array d'estructures. D'aquesta manera el sistema no haurà d'esperar sempre a tenir l'única casella disponible per poder llençar els càlculs estadístics.

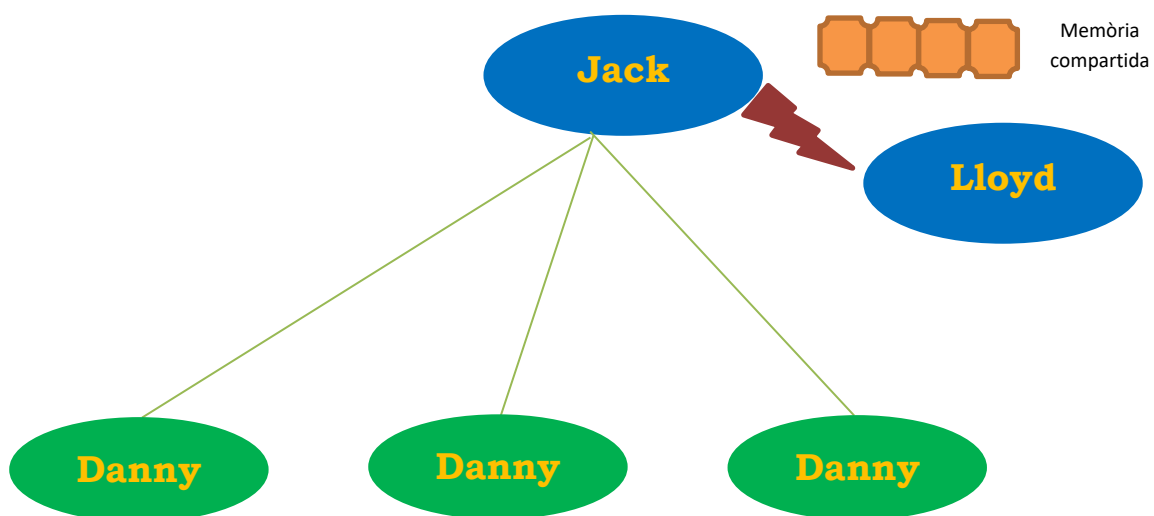


Figura 11. Optimització del procés de càlcul per la Fase 5.

Així doncs, caldrà reprogramar el mecanisme d'estadístiques per tal de poder aprofitar l'emmagatzemament de les dades a calcular per Lloyd per a que sigui més eficient possible. Quan una actualització arriba a Jack aquest podrà guardar-la (sempre que es pugui a la memòria compartida) i notificar-ho a Lloyd. A partir d'aquí no caldrà que s'espera a que aquesta informació sigui ni recollida ni tractada.

Aquesta optimització pot afectar als codis de Jack (en el sentit més ampli de la paraula) i a Lloyd. Caldrà explicar clarament a la memòria els canvis que s'han realitzat i de quina manera s'ha hagut de reprogramar aquesta funcionalitat.

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	01/11/2020
Fase 2	22/11/2020
Fase 3	10/12/2020
Lliurament Final 1 (sobre 10)	10/01/2021
Lliurament Final 2 (sobre 10)	08/02/2021
Lliurament Final 3 (sobre 8)	17/05/2021
Lliurament Final 4 (sobre 6)	18/07/2021

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. **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. D'aquesta manera es simplifica el protocol i totes les trames que s'envien tenen exactament la mateixa mida. Aquesta tindrà 3 camps ocupant sempre **un espai total de 115 bytes**:

Origen: array de 14 caràcters que contindrà el nom de qui envia la trama. Si el nom no arriba a 14 caràcters, la resta de bytes s'han d'omplir amb '\0'. Aquest camp sempre ocupa exactament 14 bytes.

Tipus: indica el tipus de trama que s'està enviant. Depenent de la trama aquesta tindrà diferents valors tal i com s'especifica posteriorment. Aquest camp ocupa sempre només 1 byte.

Data: camp de 100 bytes. Aquest camp serveix per emmagatzemar-hi valors o dades que ha d'enviar la trama. Depenent del cas pot contenir diferents valors com s'especifica en cada funcionalitat. En cas de que la informació no arribi als 100 bytes s'ompliran amb '\0'.

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 estricteament, 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.

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

Procediment de connexió

Un cop connectat el socket, Danny seguirà un procés per verificar la connexió:

1. Danny enviarà una trama sol·licitant connexió:

- a) Origen: "DANNY"
- b) Tipus: 'C'
- c) Dades: <nom de l'estació meteorològica>

2. En cas que la trama rebuda sigui correcta, Jack/Wendy contestarà amb la següent trama:

- a) Origen: "JACK" O "WENDY"
- b) Tipus: 'O'
- c) Dades: "CONNEXIO OK"

3. En cas que la trama rebuda contingui errors, Jack/Wendy contestarà amb la següent trama:

- a) Origen: "JACK" O "WENDY"
- b) Tipus: 'E'
- c) Dades: "ERROR"

Una vegada acabat el procés, s'habilitarà l'enviament d'informació.

Procediment de desconexió

Per desconectar voluntàriament DANNY se seguirà el següent mètode:

1. Danny enviarà una trama sol·licitant de desconexió:

- a. Origen: "DANNY"
- b. Tipus: 'Q'
- c. Dades: <nom de l'estació meteorològica>

I es desconnectarà...

Procediment d'enviament de dades atmosfèriques

Per enviar dades atmosfèriques de DANNY al servidor JACK se seguirà el següent mètode:

1.Danny enviarà una única trama amb **la informació necessària serialitzada** al camp dades:

- a. Origen: "DANNY"
- b. Tipus: 'D'
- c. Dades: <Data#Hora#Temp#Humitat#Pressió#Precipitacio>

on

- Data: 10 caràcters (dd/mm/aaaa)
- Hora: 8 caràcters (hh:mm:ss)
- Temperatura: 5 caràcters màxim (-XX.X)
- Humitat: 3 caràcters màxim (XXX)
- Pressió: 6 caràcters màxim (XXXX.X)
- Precipitació: 4 caràcters màxim (XX.X)

2.En cas que la trama rebuda sigui correcta, Jack contestarà amb la següent trama:

- d) Origen: "JACK"
- e) Tipus: 'B'
- f) Dades: "DADES OK"

3.En cas que la trama rebuda contingui errors, Jack contestarà amb la següent trama:

- d) Origen: "JACK"
- e) Tipus: 'K'
- f) Dades: "DADES KO"

Procediment d'enviament d'imatges

Per enviar imatges de DANNY al servidor WENDY se seguirà el següent mètode:

1.Danny enviarà una primera trama amb **la informació necessària serialitzada** al camp dades:

- a. Origen: "DANNY"
- b. Tipus: 'I'
- c. Dades: <NomDelFitxer#Mida#MD5SUM>

on

- NomDelFitxer: de màxim 30 caràcters
- Mida: en bytes expressat en format ASCII
- MD5SUM: 32 caràcters del MD5SUM

2.Seguidament DANNY anirà enviant a WENDY tantes trames com calgui parts de la imatge :

- a. Origen: "DANNY"
- b. Tipus: 'F'
- c. Dades: <DadesBinàries>

3.En cas que la imatge tramesa sigui correcta (verificant l'MD5SUM) , WENDY contestarà amb la següent trama:

- a. Origen: "WENDY"
- b. Tipus: 'S'
- c. Dades: "IMATGE OK"

4.En cas que la imatge tramesa no sigui correcta (verificant l'MD5SUM) , WENDY contestarà amb la següent trama:

- a. Origen: "WENDY"
- b. Tipus: 'R'
- c. Dades: "IMATGE KO"

Procediment de detecció de trames errònies

Si qualsevol dels processos del sistema Overlook 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:

- a) Origen: "JACK" o "WENDY" o "DANNY"
- b) Tipus: 'Z'
- c) Dades: "ERROR DE TRAMA"