

HULTSFREDSPROJEKTET DATABASPROTOKOLL V.1

Detta protokoll är del av Hultsfredsprojektet i kursen Design av samverkande system. Det bestämmer hur de olika applikationerna och simuleringen kan kommunicera med databasen över TCP, och därmed visar det även vilken data databasen innehåller.

Övergripande så består databasen av artister samt besökande personer. Personer och artister har attribut och kommandon för att få åtkomst till och i vissa fall ändra dessa attribut. Utöver detta finns även kommandon för att upprätta kommunikation över UDP för att få besökarnas simulerade positioner strömmade från databasen.

För personattributen så är de i svart gemensamma för "The Game" och Visualiseringsgrupperna, attribut i **rött** är specifika för "The Game" och i **blått** är specifika för Visualiseringsgrupperna.

PERSON

PERSONKOMMANDON

PERSON_LIST <ATTR>=value[& <ATTR>=value]*

Förklaring: Returnerar samtliga personer det/de angivna attributet/attributen. Operatorerna '<' och '>' kan användas tillsammans med attributen AGE, LENGTH, PULSE, GAMER samt PULSE.

Returnerar: Lista med ett eller flera ID-nr: [id,...], tomma listan [] eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
PERSON_LIST NAME='courtney'  
>>[3421,77,12]  
PERSON_LIST LENGTH=130 & PULSE>180  
>>[32]
```

PERSON_RAND <ATTR>=value[& <ATTR>=value]*

Förklaring: Returnerar en slumpmässig person med det/de angivna attributet/attributen.

Returnerar: Ett ID-nr: [id], tomma listan [] eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
PERSON_RAND NAME='göran'  
>>[2132]  
PERSON_RAND LENGTH=130 & FAVOURITES=[hives]  
>>[23]
```

PERSON_GET <ATTR>[, <ATTR>]* FROM id

Förklaring: Returnerar värdet för det angivna attributet för personen med angivet idnr.

Returnerar: Ett/flera attributvärden: [value,...], tomma listan [] (om personen inte existerar) eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
PERSON_GET NAME, AGE, FRIENDS FROM 891  
>>[BENGT-ERIK, 32, [932,165,3412]]
```

PERSON_SET <ATTR> TO value FOR id

Förklaring: Sätter värdet för det angivna attributet för personen med angivet idnr.

SPECIALFALL: När vänner (FRIENDS) ska läggas till finns två möjliga värden på <ATTR>, FRIENDS_ADD och FRIENDS_NEW. När FRIENDS_ADD används läggs en vän till i en persons lista över vänner, när FRIENDS_NEW används skapas en helt ny lista över vänner och den gamla listan tas bort.

Returnerar: SUCCESS eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
PERSON_SET NAME TO 'nisse' FOR 1234  
>>SUCCESS  
PERSON_SET FRIENDS_ADD TO [567,912,90] FOR 1234  
>>SUCCESS
```

PERSON_GENRES

Förklaring: Returnerar samtliga id med resp. genrepreferensvektorer.

Returnerar: Lista med id och genrepreferense: [[id1,genres1],[id2,genres2],...]

Exempel:

```
PERSON_GENRES  
>> [[12799,1 3 0 5 0 0 0 0 0 0 1 2 4 5 ...],...] (vänta 20-30 sek...)
```

PERSONATTRIBUT (<ATTR>)

ID

Förklaring: Ett unikt ID på max. 5 siffror, i intervallet 1-20000

Typ: int

Exempel: ID=1234

NAME

Förklaring: Namn på personen. Om efternamn önskas kan detta anges.

Typ: string

Exempel: NAME='nisse andersson'

POSITION

Förklaring: Personens aktuella position.

Typ: tuple (x,y) med heltal: [int,int]

Exempel: POSITION=[5,89]

STATUS

Förklaring: Simulerad eller "verklig" aktör?

Typ: int, 0 för simulerad, 1 för "verklig"

Exempel: STATUS=1

AGE

Förklaring: Personens ålder.

Typ: int

Exempel: AGE=23

GENDER

Förklaring: Personens kön.

Typ: int, 0 för man, 1 för kvinna

Exempel: GENDER=1

HAIR

Förklaring: Personens hårfärg.

Typ: string

Exempel: HAIR='BLONDE'

GAMER

Förklaring: Med i "the game"?

Typ: int, 0 för inte med i "the game", 1 för med i "the game" men passiv samt 2 för med i "the game" samt aktiv.

Exempel: GAMER=1

PARTNER

Förklaring: Spelpartner till en person som är med i "the game".

Typ: int, motsvarande id-nummer.

Exempel: PARTNER=12799

POINTS

Förklaring: Poäng i "the game", alltid noll poäng för aktörer med GAMER=0
Typ: int
Exempel: POINTS=129

RFID

Förklaring: ID-nr för den RFID-tag som personen har.
Typ: string
Exempel: RFID='0101a84c8f'

LENGTH

Förklaring: Personens längd i cm.
Typ: heltal
Exempel: LENGTH=123

PULSE

Förklaring: Personens puls.
Typ: int
Exempel: PULSE=160
Kommentar: Simuleringsapplikationen ansvarar för att uppdatera pulsen.

FRIENDS

Förklaring: Personens vänner. OBS! med kommandot PERSON_SET används istället FRIENDS_ADD eller FRIENDS_NEW (se beskrivning för PERSON_SET ovan).
Typ: lista med int (id): [id,...]
Exempel: FRIENDS=[123,678,12,1098]
Kommentar: Kan vara en eller flera id (eller i värsta fall tomma listan).

GENRES

Förklaring: Hur mycket personens musikpreferens stämmer överens med respektive genre lagras i en s.k. genrevektor. Varje position i en sträng motsvarar en musikgenre (sammanlagt 21 stycken), separerade med mellanslag. Varje position kan anta ett värde mellan 0 och 9 beroende på vilka 10 artister personen gillar mest. Genrevektorn byggs upp genom att varje deltagare får välja 10 favoritartister, som även de har genrevektorer. Artisternas genrevektorer adderas helt enkelt ihop för att få en deltagares genrevektor.
Typ: string
Exempel: GENRES='7 3 1 0 0 1 0 2 0 0 3 1 1 1 0 0 2 0 1 5 0'
Kommentar: Varje position i strängen motsvarar en genre, dessa positionsmappningar måste användas konsekvent av alla grupper som använder GENRES.

ARTIST

ARTISTKOMMANDON

ARTIST_LIST <ATTR>=value[& <ATTR>=value] *

Förklaring: Returnerar samtliga artister med det/de angivna attributet/attributen.

Returnerar: Lista med en eller flera artister: [artist,...] , tomma listan [] eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
ARTIST_LIST PERFORM=1
>>[BRODER DANIEL,CEASARS,SKATALITES,MUSTASCH,THE (INTERNATIONAL) NOISE
CONSPIRACY]
```

ARTIST_RAND <ATTR>=value[& <ATTR>=value] *

Förklaring: Returnerar en slumpmässig artist med det/de angivna attributet/attributen , tomma listan [] eller SERVER_ERROR <ERROR_CODE>

Returnerar: En artist: [artist] eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
ARTIST_RAND PERFORM=1 & GENRES='1 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 1 0 0 0 0'
>>[CEASARS]
```

ARTIST_GET <ATTR>[, <ATTR>]* FROM artist

Förklaring: Returnerar värdet för angivet attribut och angiven artist

Returnerar: Ett attributvärde: [value] , tomma listan [] eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
ARTIST_GET GENRE_ROCK FROM 'donnas'
>>[80]
ARTIST_GET INFO, GENRES FROM 'fu manchu'
>>[örongodis för vuxna,0 0 1 0 0 1 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1]
```

ARTIST_SET <ATTR> TO value FOR artist

Förklaring: Sätter värdet för angivet attribut och angiven artist

Returnerar: SUCCESS eller SERVER_ERROR <ERROR_CODE>

Exempel:

```
ARTIST_SET GENRES='0 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 1 0 1 0 1' FOR 'fjärde
världen'
>>SUCCESS
ARTIST_SET INFO='pop från götet' FOR 'håkan hellström'
>>SUCCESS
```

ARTISTATTRIBUT

ARTIST

Förklaring: Artistens unika namn.
Typ: string
Exempel: ARTIST='moneybrother'
ARTIST='kristofer åström'

GENRES

Förklaring: Artistens genrevektor, 21 mellanslagsseparerade positioner med värden 0 eller 1.
Typ: string
Exempel: GENRES='0 0 1 0 0 1 0 0 0 0 0 0 1 0 0 1 0 1 0 1'

INFO

Förklaring: Kort information om artisten
Typ: string
Exempel: INFO='spelar skjortan av dig'

PERFORM

Förklaring: Anger om artisten spelar på Hultsfredsfestivalen 2005 eller inte, 1 om artisten spelar, annars 0.
Typ: heltal
Exempel: PERFORM=1

ÖVRIGA KOMMANDON

CONNECT <PORT>

Förklaring: Upprättar UDP-kommunikation med databasen till specifierad port för att få data om simulerad position och tid strömmat till sig.

Returnerar: Bekräftelse: 'OK' om porten inte var upptagen exempelvis

Exempel:
CONNECT 33333
>>OK

START

Förklaring: Börja få UDP-paket skickade till sig enligt porten ovan.

Returnerar: Bekräftelse: 'OK'

Exempel:
START
>>OK

STOP

Förklaring: Avsluta strömningen.

Returnerar: Bekräftelse: 'OK'

Exempel:
STOP
>>OK

SET_RATE <HELTAL>

Förklaring: Sätter rate på UDP-strömningen, dvs. hur många paket/sek som ska skickas.

Returnerar: Bekräftelse: 'OK'

Exempel:
SET_RATE 3
>>OK

TIME

Förklaring: Får den aktuella simulerade tiden. Den som senast skickats från simuleringen till databasen

Returnerar: Lista [tim:min:sek]

Exempel:
TIME
>> [2:10:17]