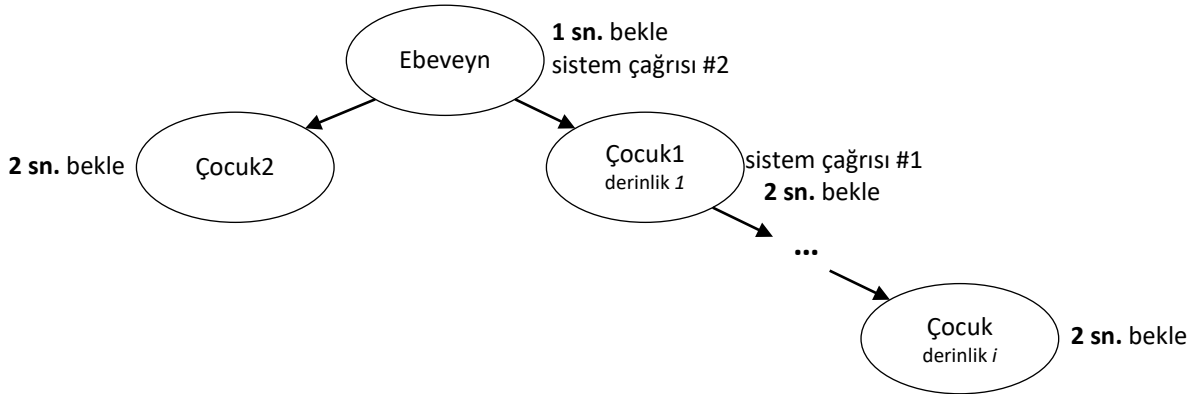


Hafta 7, Uygulama 2 (A&B): Proses Yaratma ve Çalıştırma

Çalışmanızı *dys*'deki yükleme alanına yüklemeyi unutmayınız!

Prosedür: Aşağıdaki proses ağacını oluşturan kodu yazınız. Uygulama demo videosu için [tıklayınız](#).

- Proses ağacı, kullanıcının argüman olarak girdiği sayısal değere göre derinleşmektedir (*Hint: \$./uyg2 3*). Kullanıcının girdiği değerin değişiminden bağımsız ebeveyn prosesin iki adet çocuğu vardır. Ağacın sadece Çocuk1'in olduğu tarafı derinlik bilgisine göre oluşturulur.
- Doğru derinlik kurgusu için derinlik minimum değeri 2 olarak varsayılacaktır. Terminal kodu argüman kontrolü bu doğrultuda gerçekleştirilmelidir. *Hint: int dummyint = atoi(char); //char* to integer in c*
- Her bir proses kendisinin ve ebeveyninin proses ID'sini yazmalıdır.



- Kaynak kod içerisinde **yalnızca bir adet fork()** sistem çağrısı satırı yer almalıdır.
- Her bir proses, şekilde gösterilen ve alt adımlarda istenen işlemler için **maksimum bir tane** sistem fonksiyonu çağırmalıdır. Sistem çağrısı sonuçları üst satırında bilgilendirme cümlesi (*Hint: \$ echo*) olacak şekilde (*Hint: çıktı yönlendirme*) proses temelli **log** dosyasında tutulmalıdır (**logAll.log**).
- Ebeveyn proses ilgili komutu içeren sistem fonksiyonu ile dizine **logs** isiminde bir klasör ve içerisine **log (logAll.log)** dosyasını oluşturarak tarih bilgisini basar (*Hint: \$ date*). Sonrasında diğer proseslerin de oluşturulması için 1 saniye bekler (Bu bekleme amacı semboliktir ve diğer proseslerin belirtilen hiyerarşiye göre düzenli bir şekilde oluşmasını sağlamaktır). Ardından ilgili komutu gerçekleştiren sistem fonksiyonu ile **logs** klasörü altındaki **log** dosyasına o anda işletim sistemi üzerinde koşan **yalnızca** üstteki proses ağacını PID bilgileriyle birlikte basar ve işlemini tamamlar.
Hint: \$ man pstree \$ pstree <options?> <PID> (pstree mini demo)
Hint: char mychar[size]; sprintf(mychar, "%s %d", dummytext, dummyint); //hybrid string+int to char in c*
- Çocuk1 (derinlik1), **log** dosyasının içeriğini ekrana bastırır (önceki çalıştırmaların sonuçları dosya içerisinden görülüyor olacak, mevcut çalıştırma ayrıca birazdan basılacak).
- Ebeveyn haricindeki diğer tüm prosesler derinlik bilgisine göre kendisinden sonraki prosesi yarattıktan sonra 2 saniye bekler (bu bekleme amacı da semboliktir ve proses ağacının düzenli bir şekilde görülmesini sağlamaktır) ve işlemini tamamlar.

Not: Yukarıda belirtilen her bir işlemin sıralamada ayrıca bir önemi bulunabilir. Doğru sonuç için istenenleri dokümandaki sıralama ile gerçekleştiriniz.

Yüklenmesi Gereken Dosyalar: <OgrNo> için öğrenci numaranızın son altı hanesini giriniz!

kaynak kod dosyası <OgrNo>_uyg<#><Şube>.<dil>
*çalıştırma sonuçlarını içeren çıktı dosyası <OgrNo>_uyg<#><Şube>_output.txt
log dosyası <OgrNo>_uyg<#><Şube>_logAll.log

*derlenmiş bir executable dosya değildir, koşu sonuçlarını içerir! Manuel oluşturulabilir.

Puanlama Sistemi:

	QUIZ	PERFORMANS
doğru ağaç	30	15 zamanlama
argüman	20	15 pre-lab tartışma platformu kullanımı
sistem çağrıları	15	15 düzen, açıklama

Uygulamalar performans ve uygulamanın doğru kısımlarına göre değerlendirilir. Yüklenmesi gereken dosyalar "**<OgrNo>_<Ders><Şube><AkademikYıl><GUZ/BHR/YAZ>_<UYG/HW><#>.zip**" (Örnek: **152120081026_IsSisLabC1920BHR_UYG9.zip**) isimlendirme formatında sıkıştırılarak yükleme alanına yüklenir. **Yükleme hatalarına ceza puanı uygulanır. Covid-19 önlemleri ve uzaktan eğitim-öğretim süreci doğrultusunda sisteme yüklenmeyen veya yüklenip laboratuvar sırasında gösterilmeyen çalışmalar geçersiz sayılır.**